Do Declarative Process Models Help to Reduce Cognitive Biases Related to Business Rules?

Kathrin Figl[1](✉), Claudio Di Ciccio[2], and Hajo A. Reijers[3]

[1] University of Innsbruck, Innsbruck, Austria
kathrin.figl@uibk.ac.at
[2] Sapienza University of Rome, Rome, Italy
diciccio@di.uniroma1.it
[3] Utrecht University, Utrecht, The Netherlands
h.a.reijers@uu.nl

Abstract. Declarative process modeling languages, such as DECLARE, represent processes by means of temporal rules, namely constraints. Those languages typically come endowed with a graphical notation to draw such models diagrammatically. In this paper, we explore the effects of diagrammatic representation on humans' deductive reasoning involved in the analysis and compliance checking of declarative process models. In an experiment, we compared textual descriptions of business rules against textual descriptions that were supplemented with declarative models. Results based on a sample of 75 subjects indicate that the declarative process models did not improve but rather lowered reasoning performance. Thus, for novice users, using the graphical notation of DECLARE may not help readers properly understand business rules: they may confuse them in comparison to textual descriptions. A likely explanation of the negative effect of graphical declarative models on human reasoning is that readers interpret edges wrongly. This has implications for the practical use of business rules on the one hand and the design of declarative process modeling languages on the other.

Keywords: Business rule representation · Declarative process modeling · Reasoning · Diagrams · Cognition

1 Introduction

Visual diagrammatic process models are widely used to analyze, design, and improve business processes. As process orientation and process awareness are guiding paradigms for organizational innovation, standardization, and information systems design, they attract high attention in research and practice. Against this background, human comprehensibility of diagrammatic process models is a highly relevant issue. Many researchers have recently turned to empirically investigate factors that influence their comprehensibility with the final goal to optimize the fit between process model design and human cognitive capabilities [1].

In the last decade, a paradigm to represent processes has gained momentum, which is an alternative to the classical, state-transition based one: the declarative specification

G. Dobbie et al. (Eds.): ER 2020, LNCS 12400, pp. 119–133, 2020.
https://doi.org/10.1007/978-3-030-62522-1_9

of processes. Declarative process models dictate the business rules that the process must comply with in the form of constraints. In comparison with procedural approaches, which consist of "closed" representations (i.e., only the represented process runs are allowed), declarative approaches yield "open" representations (i.e., every run is allowed, unless it violates a constraint). Declarative languages such as DECLARE provide a repertoire of constraint templates that come bundled with a graphical representation [2, 3]. The understandability of declarative process models, however, is still a matter of debate [4].

The main aim of this paper is to investigate the effect that declarative process models have on the way humans reason. To that extent, we report in this paper on our controlled experiment in which we compare textual descriptions of business rules against textual descriptions supplemented with DECLARE models. Interestingly, our results indicate that the DECLARE models do not improve but rather decrease the reasoning performance of participants. In fact, the use of supplementary DECLARE models did not help novice users understand business rules better, but seemed to have confused them in comparison to textual descriptions alone. The evidence from this study points at the difficulty of novice users to master the meaning of such models and use these effectively. This motivates further research into more intuitive notations for declarative constraints.

2 Related Work

Considerable research work has been conducted on the comprehension of process models [5], However, relatively limited research has taken place to investigate the effect of process models on *deductive reasoning*. In such an approach, the "mental process of making inferences that are logical" [6, p. 8] based on process models is compared against a baseline (such as, for instance, a textual narrative of the process). One of the few examples of this type of research is the work of Boritz et al. [7], who report the superiority of a narrative process version over a process model for deductive reasoning tasks while identifying and assessing control risks of the process. Our work can be placed in this tradition.

Relevant from the viewpoint of investigating declarative process models more generally is the work by Haisjackl and Zugal [8]. They compared graphical and textual declarative process models and reported lower comprehension performance for the textual representation. However, the textual notation they used was not a natural language narrative version of the declarative process model, but a domain-specific textual language. More recently, López et al. [9] introduced a tool that allows users to highlight in a process textual description the passages that describe constraints. The tool automatically generates the corresponding visual elements of the DCR Graphs declarative notation. To improve the comprehensibility of declarative process models, De Smedt et al. [10] introduced an approach to uncover "hidden" behavioral restrictions, which are not explicitly visible when reading a declarative model yet are entailed by the interplay of the rules. Other research efforts have been made to investigate the effect of using gateway symbols in process models on human reasoning [11]. Further studies have identified specific difficulties of reasoning on the basis of process models, e.g., high interactivity of model elements and the presence of control-flow patterns as loops heighten the cognitive difficulty and error-proneness of reasoning tasks [12].

The present study differs from prior experiments in that we compare textual narratives with textual narratives that are supplemented with declarative process models. More specifically, we address the research question whether DECLARE models, when used in addition to if-then clauses in written language, help humans understand conditional if-then arguments in business rules. The idea, here, is that a graphical, declarative process model may serve as a cognitive short-hand for processing the textual clauses. From a cognitive research viewpoint, it is already known that typical logical fallacies may occur in conditional reasoning with text [6]. However, it is not known whether the same fallacies do occur when using DECLARE models or whether these visual representations actually help readers to understand business rules avoiding cognitive fallacies. Answering this question is relevant for a variety of tasks in which humans use process models, e.g., for checking compliance of process execution traces with a process model.

3 Theoretical Background

Business Rules and Deductive Reasoning. Business rules are generally defined as "statements that aim to influence or guide behavior and information in the organization." [13, p. 52] Documentation of business rules is relevant to make them transparent and to avoid rule conflicts. In practice, when using natural language to document business rules, conditional if-then statements (if *cause*, then *effect*) are made to describe causal relationships [14]. Formal logics define whether a conditional inference based on given premises is true. A deduction is valid "if its conclusion must be true given that its premises are true" [15, p. 372]. Table 1 gives an example of the four standard conditional inferences based on a business rule [16].

Table 1. Examples of valid and invalid conditional inferences.

	Affirmative	Negative
Valid	If a rental car is returned late, then a penalty is charged The rental car is returned late Therefore, a penalty was charged "**Modus ponens**"	If a rental car is returned late, then a penalty is charged A penalty is not charged Therefore, the rental car was not returned late **"Modus tollens"**
Invalid	If a rental car is returned late, then a penalty is charged A penalty is charged Therefore, the rental car was returned late **"Affirmation of the consequent"**	If a rental car is returned late, then a penalty is charged The rental car is not returned late Therefore, a penalty was not charged "**Denial of the antecedent**"

However, "natural" human reasoning may not always be sound. Humans are prone to typical misinterpretations of if-then statements. For instance, they may interpret the business rule "If the product is deliverable in less than two days, then the product is ordered from the supplier" in a probabilistic way (it is usually this way, but not always). They may also re-interpret the rule biconditionally as "If the product is ordered from the supplier that means the product has been deliverable in less than two days." From a logical standpoint, these are both logical fallacies. If we take an example from everyday life with the two premises "If it's raining then the streets are wet," then the commutation of conditionals "If the streets are wet then it's raining" would also be logically incorrect. People are still likely to make this logical error, because in reality it might be a good rule of thumb since wet streets dry fast after rain. Depending on content-effects, people are likely to misinterpret conditional relationships [17]. A biconditional misunderstanding leads to the logical fallacy known as "*affirmation of the consequent*."

Representation of Business Rules. Business rules may be found as tacit knowledge in the heads of employees (unwritten), may be part of guidelines (varying from informal to formal), part of enterprise models, are implemented in an information system, or are part of a rules engine (highly formalized) [18]. As most organizations use text and graphical languages to document their processes, business rules are likely to be found in practice in textual descriptions as well as in diagrams.

The Business Rules Manifesto [19] advises to separate business models from processes and states: "Rules should be expressed declaratively in natural-language sentences for the business audience." However, natural language may be ambiguous and not precise enough to document business rules. Traditional means to document business rules are structured English (a subset of natural language), decision tables, and decision trees [20]. Process modeling languages focus on representing the control flow of activities, including decisions. It is possible to implicitly model business rules in a process model embedded in the control flow logic. In a procedural approach, a business rule on, for example, a credit limit may be included in various processes (e.g., new order, change order, change costumer credit limit), while in the business rule approach it might be defined in a rulebase of a rule management system, which all processes can then use [18]. As process models are typically procedural, they do not offer the same modeling convenience for documenting business rules as declarative rule specification languages do. Therefore, research efforts have been undertaken to integrate and combine process models and rule modeling languages [21]. In recent years, researchers have proposed approaches to visually model business rules as an extension to existing process modeling languages [22].

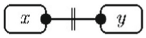
In this paper, we focus on the expression of business rules in declarative process models. A primary goal of this paper is to assess whether declarative process models help humans to better understand business rules in comparison to natural language if-then statements. In particular, we like to understand how a declarative process model as a visual aid influences human cognitive processes.

In general, the cognitive fit between an information representation and the specific task is necessary—a representation may never be optimal in all cases but should relate to the task at hand [23]. While researchers have argued that "once the logical form of the problem has been extracted from a diagram, the same chain of deductions based on the same rules of inference [in mind] should unfold" [15, p. 372]. There are also a variety of theoretical perspectives that suggest superiority of diagrams over narrative text for human reasoning. For instance, externalizations in the form of diagrams can reduce working memory overload due to computational off-loading [24]. Knowledge put down in a diagram as "external storage" need not be maintained in the working memory. Also, both the visual and verbal working memory subsystem can be used when working with process models [25]. There are also hints that the visual structure of process models is closer to human mental models than text [26]. Based on the high importance of mental models for reasoning [27], this may also lead to easier reasoning with process models than with text. We expect similar effects for representing rules in the context of business process models.

Constraints for Declarative Process Models. Declarative process models define the behavior of a process by means of constraints, i.e., temporal rules that specify the conditions under which activities may, must, or cannot be executed. A well-known declarative process modeling language is DECLARE [2]. This language provides a repertoire of (constraint) *templates*, i.e., parametrized rules. The major benefit of using templates is that analysts do not have to be aware of the underlying logic-based formalization to understand the models. They work with the graphical or textual representation of templates, while the underlying formulae remain hidden. The repertoire of templates of DECLARE is based upon the seminal work of Dwyer, Avrunin, and Corbett [28] on the most recurring property specification patterns for the verification of finite-state systems in software engineering. Typical examples of DECLARE templates are *Participation* (x) and *Precedence* (x,y). The former applies the *Participation* template on the parametric activity x (the *target*) and states that x must occur in every run of the process. The latter applies the *Precedence* relation template on activities y (*activation*) and x (*target*), imposing that if y occurs, then x must have occurred earlier in the same run. Intuitively, activations determine the circumstances triggering the constraint (the *if* part of an if-then statement); targets are the consequential conditions being imposed upon the occurrence of the activations (the *then* part of an if-then statement).

A declarative process model is a set of constraints that must all be satisfied during the process run. The graph built from the network of graphical elements denoting the DECLARE constraints is called a DECLARE model [3]. Table 2 illustrates the list of DECLARE templates that will be considered in the context of this paper. For the sake of clarity, we adopt the abbreviated names *Participation*(x) and *AtMostOne*(x), introduced in [29], to indicate the *Existence*$(1, x)$ and *Absence*$(2, x)$ templates, respectively. In particular, we consider in this paper at least a DECLARE template for each of the categories illustrated in [29]: *Participation* and *AtMostOne* (predicating on the number of activity occurrences); *Init* and *Last* (on the position of activity occurrences); *Response* (imposing a temporal order between activation and target); *ChainResponse* (forcing an immediate occurrence of the target after the activation); *Precedence* (reverting the temporal order imposed by *Precedence*); *AlternatePrecedence* (which enforces *Precedence* by avoiding the recurrence of the activation); *ChainPrecedence* (forcing an immediate occurrence of the target before the activation); *Succession* (assigning both activities the role of activation and target, as it stems from the conjunction of *Response* and *Precedence*); *NotCoExistence* (dictating the mutual absence of activities, rather than their co-occurrence). Notice that templates like *Succession*(x, y) and *NotCoExistence*(x, y) are biconditional; therefore, both parameters x and y play the role of activation and target.

Table 2. Some DECLARE templates.

Template	Act.	Tar.	Description	Graphical notation
AtMostOne(x)		x	Activity x occurs at most once	
Participation(x)		x	Activity x occurs at least once	
Init(x)		x	Activity x always occurs first	
Last(x)		x	Activity x always occurs last	
Responded Existence(x,y)	x	y	If x occurs, then y must occur, too	
Response(x,y)	x	y	If x occurs, then y must occur afterwards	
ChainResponse(x,y)	x	y	If x occurs, then y must occur immediately afterwards	
Precedence(x,y)	y	x	If y occurs, then x must have occurred beforehand	
Alternate Precedence(x,y)	y	x	If y occurs, then x must have occurred beforehand, and no other y can have recurred in between	
ChainPrecedence(x,y)	y	x	If y occurs, then x must have occurred immediately beforehand	
Succession(x,y)	x, y	x, y	If x occurs, then y must occur afterwards; if y occurs, then x must have occurred beforehand	
NotCoExistence(x,y)	x, y	x, y	If x occurs, then y cannot occur; if y occurs, then x cannot occur	

4 Research Model and Hypotheses

Having laid out the relevant theoretical foundation to examine textual and diagrammatic representations of business rules, we present our research model in Fig. 1. We expect the *process rule representation* to influence *deductive reasoning performance* of a person (as measured in terms of the percentage of correctly solved reasoning tasks, the time taken to solve those tasks, and the occurrence of reasoning fallacies).

In detail, we advance the following hypotheses concerning business rules in an if-then form. We hypothesize that reasoning based on text and supplementary graphical process models, i.e., a mixed model, may deviate from reasoning based on natural text alone. Based on the fact that rules are made *explicit* within the declarative process model that is part of the mixed model, one can assume that people are less likely to misinterpret the underlying logic than in the setting of having to rely purely on text. Therefore, we first propose that: Declarative process models in combination with textual representations support higher reasoning performance compared to the use of textual representations on their own (**H1**).

We do think that the positive effects of using a declarative process model as part of a mixed model should be qualified. It is conceivable that opposing effects may also occur

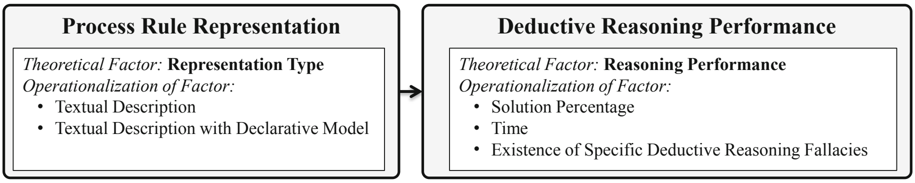


Fig. 1. Research model.

for the use of certain model structures. According to Britton and Jones [30], uni-directed arrow symbols (as the directed edges used in declarative process models) are mostly interpreted as "if… and only if", while bidirectional arrow symbols are interpreted as "if… then". Therefore, it may happen that directed edges – which occur in Declare models – are perceived as "semantically perverse", i.e., "a novice reader would be likely to infer a different (or even opposite) meaning from its appearance" [31, p. 764]. Such a misunderstanding would increase the probability of a biconditional misinterpretation of rules in declarative process models. Thus, we propose that: Rules in declarative process models with directed edges that are combined with a textual representation are more likely to be misinterpreted as biconditional than rules as textual representation alone (**H2**).

5 Research Method

To test the hypotheses, we conducted a fully randomized, controlled laboratory experiment. The research design included one between-group factor business rule representation with two levels: textual vs. text with a diagram. In particular, we compare the results yielded by textual representations with those achieved with *mixed* text-diagram representation. Our rationale is that comparing pure text with declarative models alone (without additional descriptions) would bias our results with factors that are out of scope. Specifically, these factors are (1) the efficacy with which users were trained with the graphical notation of Declare, and (2) the personal inclination of users towards textual or visual means. Instead, we aim at investigating whether the graphical notation of rules adds value over the use of pure text. Therefore, we resort to declarative process models that are enriched with textual descriptions. This representation style is well-known; its benefits are described, for example, by Recker et al. [32].

Materials and Procedures. We employed business rules from different scenarios in two different types of reasoning tasks: card-based Wason selection tasks and model-based comprehension tasks. Experimental materials are available under the following link: http://kathrinfigl.com/declare-questionnaire/. The first type of reasoning tasks for participants were Wason selection tasks [33, 34]. The Wason selection task is a famous puzzle often used in deductive reasoning research. The participants are confronted with a business rule in the form "if P then Q." They get cards (with "P"/"not-P" on one side and "Q"/"not Q" on the other side) and have to select all cards that need to be turned over to test the business rule and to find out whether the business rule is verified or not. Figure 2 shows the Wason selection task for the *RespondedExistence* template.

In company X the following rule has been established:
Rule: "**If K occurs in the process instance, then J occurs as well.**"

Experimental Group 1	Experimental Group 2
[no visual model]	

Imagine you are responsible for process compliance in this company and your task is to find out whether the rule has been violated.

The cards below represent four process instances. The process instance is already finished. On one side of each card is written whether or not the K did occur, on the other side whether or not J did occur.

Your task: Please select all the cards that you have to turn over (i.e., all of which you need to know the information on the back) in order to find out whether the rule was violated.

Not Selected:	Selected Cards:
K did occur *[Modus ponens (valid); card should be selected]*	**chosen**
J did occur *[Affirming the consequent (invalid); card should not be selected]*	**chosen**
K did not occur *[Denying the antecedent (invalid); card should not be selected]*	**chosen**
J did not occur *[Modus tollens (valid); card should be selected]*	**chosen**

Fig. 2. Wason selection task for the compliance check of DECLARE constraints.

In the Wason selection tasks, we could focus on an isolated representation of single constraints. We chose those five constraints having defined activation (if) and target (then), regardless of the temporal perspective (*RespondedExistence*), or containing ordering criteria (*Response* and *Precedence*), also including immediate sequencing (*ChainResponse*, *ChainPrecedence*).

In addition to the five Wason selection tasks, we used two declarative process models with comprehension tasks: one dealt with the issue of handling orders (Fig. 3), the other with handling invoices (Fig. 4). For the "order handling" and "invoice handling" process models we asked 9 and 14 questions respectively, requiring the participants to classify process runs as "correct" or "incorrect" (or select "I don't know"). Although we used an online survey tool, we also provided the two models on paper to ensure readability. Figure 3 and Fig. 4 show the mixed text-diagram representation. In the textual experimental group, we left out the DECLARE models; we only gave the remaining textual parts to the participants. In the following, we present some examples of process runs for the "order handling" process model, which we will henceforth represent as finite sequences delimited by angular brackets. For instance, ⟨"Receive order", "Locate ordered goods", "Dispatch ordered good", "Mark order as completed"⟩ is a correct process run. By contrast, ⟨"Receive order", "Dispatch ordered good", "Mark order as 'out of stock'", "Mark order as completed"⟩ is incorrect because *NotCoExistence*("Dispatch ordered good", "Mark order as 'out of stock'") is violated. Also, ⟨"Locate ordered goods", "Dispatch ordered good", "Mark order as completed"⟩ is incorrect too, because *Init*("Receive order") is violated.

Participants. In this study, 74 information systems students from the Vienna University of Business and Economics participated voluntarily in the context of course units. We

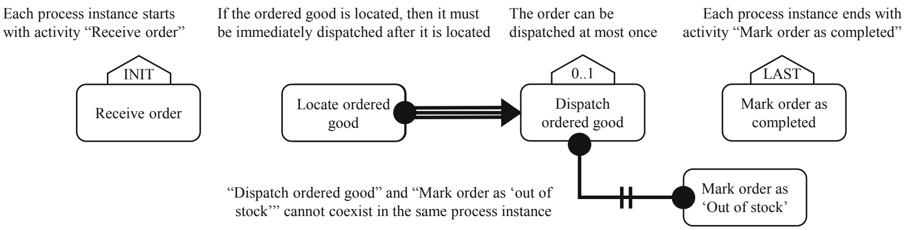


Fig. 3. DECLARE model of an order handling process mixed with textual descriptions.

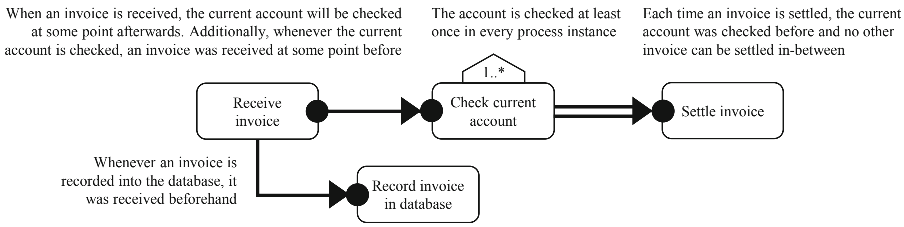


Fig. 4. DECLARE model of an invoice handling process mixed with textual descriptions.

chose to involve information systems students as they serve as an adequate proxy for novice corporate users of business process models. Therefore, they are good target users of declarative process modeling notations. We recruited students from a Bachelor course on enterprise modeling and a Masters course in Business Process Management. This helped to ensure that the students that participated would already have some experience on the use of conceptual, graphical models, in particular event-driven process chains or BPMN models.

6 Results

To compare the experimental groups, we performed analyses of variance. Table 3 illustrates the results. We can see that no differences between experimental groups could be found concerning solution percentages in the Wason selection tasks. Yet there was a significant effect of the presence of DECLARE models (in addition to the textual description) on the solution percentage in the model comprehension tasks. In contrast to the expectation behind H1, participants could answer more model comprehension tasks correctly in the text-only setting (71%) than with an additional DECLARE model (64%). Time did not differ significantly between the groups in both task types.

We also tested for differences between the text-only and mixed text-diagram condition of the solution percentage of the four standard conditional inferences (modus ponens, modus tollens, affirmation of the consequent and denial of the antecedent). No significant differences could be detected either. In general, 'modus ponens' and 'denial of the antecedent' were easiest (correctly identified by 74% and 63% of participants); 'affirmation of the consequent' (55% of participants did solve this task correctly; thus, 45% committed this fallacy) and modus tollens were most difficult (53% solution percentage).

Table 3. Influence of DECLARE models on deductive reasoning.

	Text only (n = 38)		Mixed text + diagram (n = 37)		Stat. test
	M/count	SD/%	M/count	SD/%	
Wason selection tasks					
Solution percentage	61%	0.17	61%	0.16	*n.s.*
Time [sec]	59.73	25.1	77.01	76.64	*n.s.*
Model comprehension tasks					
Solution percentage	71%	0.17	64%	0.16	F = 4.03, p = 0.05
Time [sec]	212.38	86.06	212.10	79.26	
Items indicating biconditional misunderstanding					
Solution percentage	51%	0.38	43%	0.31	*n.s.*

The DECLARE models that were part of the mixed representations did not help to prevent any of the logical errors. However, since absence of evidence is not evidence of absence, an additional graphical representation could have qualitatively altered the reasoning of participants. To avoid a Type II error (i.e. failing to reject an erroneous null hypothesis), we calculated a post hoc power analysis with the G*Power tool [35]. The power (1-β) of a statistical test "is the complement of β, which denotes the Type II or beta error probability of falsely retaining an incorrect H0" [35, p. 176]. In the case of a one-sided t-test for two independent means, the samples sizes (group 1 = 38, group 2 = 37), an error probability of $\alpha = 0.05$ and medium effect size d = 0.5, the power (1-β) = 0.69. Since conventionally a power of 1-β = 0.8 should be reached [36], a higher sample size might be needed to detect a medium effect that might be relevant to practice.

To test hypothesis **H2**, we identified four items in which biconditional misunderstanding could occur due to process model parts with directed edges. Table 3 illustrates the mean solution percentage for these three items alone (but does not report time, because times were not recorded item-wise). **H2** had to be rejected since the mean solution percentage was not significantly different. Still, we want to discuss results for one item of the "order handling" process model. It is interesting to note that the process run ⟨"Receive order," "Dispatch ordered good," "Mark order as completed"⟩ was identified for the "order handling" process model as correct by only 32% of participants. Thus, 67% of participants answered this question wrongly. A likely explanation for this misunderstanding in our opinion is the biconditional misinterpretation of the business rule "If the ordered good is located, then it must be immediately dispatched after it is located." This item was significantly (F = 4.29, p = 0.03) answered correctly more often by participants in the text-only group (42%) than in the group having text and a DECLARE model at hand (19%). Thus, the DECLARE model was in this case even more misleading than the textual if-then statement.

A more detailed analysis of the items in which the two groups differed suggests that DECLARE models were probably read as if they were procedural process models, especially if directed edges were used. Notice that all participants already had experience with procedural process models. Table 4 reports some examples of reasoning tasks based on the "invoice handling" process model (Fig. 4). Visual elements used in procedural process models to depict control flow – directed edges/arrows – can easily be associated with a causal meaning [37] and look similar to directed edges used in DECLARE models. This could explain why participants were confused in the text + DECLARE condition and performed worse than those who only received the textual description.

Table 4. Influence of DECLARE models on selected reasoning tasks.

Process runs for reasoning tasks	Verif.	Text only (n = 38)		Text + DECLARE model (n = 37)		Stat. test
		Mean	SD	Mean	SD	
⟨"Receive invoice", "Record invoice in database", "Check current account"⟩	valid	82%	0.39	51%	0.51	$t = 2.88$, $p = 0.005$
⟨"Receive invoice", "Record invoice in database"⟩	invalid	71%	0.46	49%	0.51	$t = 2.00$, $p = 0.05$
⟨"Receive invoice", "Check current account", "Record invoice in database"⟩	valid	74%	0.45	38%	0.49	$t = 3.30$, $p = 0.001$

7 Discussion and Limitations

Our study set out to empirically evaluate the effect of a declarative process model on human reasoning, focusing in particular on common reasoning fallacies regarding if-then constructs. Our results, while preliminary, suggest that DECLARE models do not help readers to better understand given textual business rules. Rather, they lead to more reasoning mistakes.

Boekelder et al. [38] have compared similar representations of if-then statements in their experiment on operating control panels. Contrary to our results, they found that participants took more time for reading and solving the tasks when using lists (comparable to our textual condition) than when using flowcharts (comparable to a procedural process model), but no significant performance differences were found. A possible explanation for the contrasting results might be that they compared if-then-else statements instead of if-then statements and procedural instead of declarative visual representations. This conjecture finds support in the experimental results described by Haisjackl et al. [39], which indicate that the graphical notation elements that are similar in both procedural and declarative process modeling languages, though different semantics, cause considerable confusion.

As this was the first study addressing the effect of declarative process models on deductive reasoning, we deemed internal validity more important than external validity [40]. We used artificially created snippets of declarative process models and relatively small and straightforward process models to isolate the factor of interest. External validity in the sense of generalizing the findings to more complex process scenarios will thus be limited. Additionally, our choice of a student sample limits generalizability as, e.g., results are not generalizable to users who are already experts in using the DECLARE graphical notation. The main reason to use a student sample was to avoid an experimental bias of prior experience with process modeling. Although students had prior experience with event-driven process chains or BPMN models, we do not think that it would have been advantageous to take participants without any experience with process models to avoid misinterpretations of directed edges in DECLARE models. After all, the semantic association of directed edges with sequence/causality and their interpretation as “if…and only if” is naturally and culturally shaped [31]. Similarly, participants have been familiar with ER modelling and, therefore, might have recognized the use of cardinalities in the symbols for *AtMostOne* and *Participation* more easily than participants without any prior modeling experience.

8 Implications for Research and Practice

The presented work contributes to the advancement of modeling language evaluation methods. It demonstrated how data collection methods from the cognitive science field of deductive reasoning research as the Wason selection task could be used to assess model comprehension empirically.

The research design and the preliminary results presented in this paper serve as a contribution to further open the black box of human understanding of process models. It adds to the growing body of empirical work on process model comprehension. An implication of the results for practitioners includes exercising caution when tasks involve reasoning on the basis of business rules, and formal correctness of human inferences is important. There is a variety of real, practical situations in which human reasoning based on business rules is relevant and cannot or should not be automated. For instance, employees may need to analyze or check conditions for decision points, and they may give instructions on how to enforce specific business rules. Similarly, business process analysts may assess and evaluate differences between rules in existing process models and their application in real-world process instances. However, human actors might use rules of thumb and, as the low solution percentages demonstrated, logical errors do occur. The evidence from this study further emphasizes the importance of developing understandable visual modeling approaches to business rules, to support enterprise modeling practice.

The results gave a hint that readers of a process model tend to misinterpret declarative process models as procedural models, and are less likely to look at the embedded business rules in isolation. Such results support the idea to further separate business decision and process logic to avoid human reasoning fallacies.

9 Conclusion

The present study was designed to determine the effect of declarative process models on human reasoning. By taking a look at various potential deductive reasoning fallacies, this work denotes an essential extension to the literature on process model comprehensibility. Overall, our preliminary findings suggest that declarative process models do not qualitatively alter human reasoning and visual process models do not outperform written language in supporting humans to understand conditional if-then arguments. Rather, they may even confuse readers. As business rules can help organizations to achieve their goals, e.g., by reducing costs or improving communication, their proper understanding by all human actors involved is crucial.

References

1. Houy, C., Fettke, P., Loos, P.: On the theoretical foundations of research into the understandability of business process models. In: European Conference on Information Systems, Tel Aviv (2014)
2. van der Aalst, W.M.P., Pesic, M.: DecSerFlow: towards a truly declarative service flow language. In: Bravetti, M., Núñez, M., Zavattaro, G. (eds.) WS-FM 2006. LNCS, vol. 4184, pp. 1–23. Springer, Heidelberg (2006). https://doi.org/10.1007/11841197_1
3. Bose, R.P.J.C., Maggi, F.M., van der Aalst, W.M.P.: enhancing declare maps based on event correlations. In: Daniel, F., Wang, J., Weber, B. (eds.) BPM 2013. LNCS, vol. 8094, pp. 97–112. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40176-3_9
4. Fahland, D., et al.: Declarative versus imperative process modeling languages: the issue of understandability. In: International Workshop BPMDS and International Conference EMMSAD, pp. 353–366 (2009)
5. Figl, K.: Comprehension of procedural visual business process models. Bus. Inf. Syst. Eng. **59**, 41–67 (2017)
6. Johnson-Laird, P.N.: Deductive reasoning. Wiley Interdisciplinary Rev. Cogn. Sci. **1**, 8–17 (2010)
7. Boritz, J.E., Borthick, A.F., Presslee, A.: The effect of business process representation type on assessment of business and control risks: diagrams versus narratives. Issues Account. Educ. **27**, 895–915 (2012)
8. Haisjackl, C., Zugal, S.: Investigating differences between graphical and textual declarative process models. In: Iliadis, L., Papazoglou, M., Pohl, K. (eds.) CAiSE 2014. LNBIP, vol. 178, pp. 194–206. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07869-4_17
9. López, H.A., Debois, S., Hildebrandt, T.T., Marquard, M.: The process highlighter: from texts to declarative processes and back. BPM (Dissertation/Demos/Industry) **2196**, 66–70 (2018)
10. De Smedt, J., De Weerdt, J., Serral, E., Vanthienen, J.: Discovering hidden dependencies in constraint-based declarative process models for improving understandability. Inf. Syst. **74**, 40–52 (2018)
11. Recker, J.: Empirical investigation of the usefulness of gateway constructs in process models. European J. Inf. Syst. **22**, 673–689 (2013)
12. Figl, K., Laue, R.: Influence factors for local comprehensibility of process models. Int. J. Hum Comput Stud. **82**, 96–110 (2015)
13. Steinke, G., Nickolette, C.: Business rules as the basis of an organization's information systems. Ind. Manage. Data Syst. **103**, 52–63 (2003)
14. Ross, R.G.: What's wrong with if-then syntax for expressing business rules ~ one size doesn't fit all. Bus. Rules J. **8** (2007)

15. Bauer, M.I., Johnson-Laird, P.N.: How diagrams can improve reasoning. Psychol. Sci. **4**, 372–378 (1993)
16. Byrne, R.M.J., Johnson-Laird, P.N.: If and the problems of conditional reasoning. Trends Cogn. Sci. **13**, 282–287 (2009)
17. Cummins, D., Lubart, T., Alksnis, O., Rist, R.: Conditional reasoning and causation. Memory Cogn. **19**, 274–282 (1991)
18. Object Management Group: How Business Rules Relate to Business processes from a business person's point of view. Business Rules Symposium, Minneapolis (2010)
19. Ross, R.G.: The Business Rules Manifesto. Business Rules Group (2003)
20. Vessey, I.: Cognitive fit: a theory-based analysis of the graphs versus tables literature. Decision Sci. **22**, 219–240 (1991)
21. Zur Muehlen, M., Indulska, M.: Modeling languages for business processes and business rules: a representational analysis. Inf. Syst. **35**, 379–390 (2010)
22. Knuplesch, D., Reichert, M., Ly, L.T., Kumar, A., Rinderle-Ma, S.: Visual modeling of business process compliance rules with the support of multiple perspectives. In: Ng, W., Storey, V.C., Trujillo, J.C. (eds.) ER 2013. LNCS, vol. 8217, pp. 106–120. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-41924-9_10
23. Vessey, I., Galletta, D.: Cognitive fit: An empirical study of information acquisition. Information Systems Research **2**, 63–84 (1991)
24. Scaife, M., Rogers, Y.: External cognition: How do graphical representations work? Int. J. Hum.-Comput. Stud. **45**, 185–213 (1996)
25. Baddeley, A.D.: Working memory. Science **255**, 556–559 (1992)
26. Glenberg, A.M., Langston, W.E.: Comprehension of illustrated text: Pictures help to build mental models. J. Memory Lang. **31**, 129–151 (1992)
27. Johnson-Laird, P.N.: Mental models and human reasoning. Proc. Nat. Acad. Sci. **107**, 18243–18250 (2010)
28. Dwyer, M.B., Avrunin, G.S., Corbett, J.C.: Patterns in property specifications for finite-state verification. In: International Conference on Software Engineering, pp. 411–420. IEEE (1999)
29. Di Ciccio, C., Maggi, F.M., Montali, M., Mendling, J.: Resolving inconsistencies and redundancies in declarative process models. Inf. Syst. **64**, 425–446 (2017)
30. Britton, C., Jones, S.: The untrained eye: How languages for software specification support understanding in untrained users. Hum.-Comput. Interact. **14**, 191–244 (1999)
31. Moody, D.L.: The "physics" of notations: towards a scientific basis for constructing visual notations in software engineering. IEEE Trans. Software Eng. **35**, 756–779 (2009)
32. Recker, J., Green, P.: How do individuals interpret multiple conceptual models? a theory of combined ontological completeness and overlap. J. Assoc. Inf. Syst. **20**, 1 (2019)
33. Liberman, N., Klar, Y.: Hypothesis testing in wason's selection task: social exchange cheating detection or task understanding. Cognition **58**, 127–156 (1996)
34. Wason, P.C.: Reasoning. In: Foss, B. (ed.) New Horizons in Psychology. Penguin, London (1966)
35. Faul, F., Erdfelder, E., Lang, A.-G., Buchner, A.: G* power 3: a flexible statistical power analysis program for the social, behavioral, and biomedical sciences. Behav. Res. Methods **39**, 175–191 (2007)
36. Fox, N., Mathers, N.: Empowering research: statistical power in general practice research. Family Pract. **14**, 324–329 (1997)
37. Tversky, B., Zacks, J., Lee, P., Heiser, J.: Lines, blobs, crosses and arrows: diagrammatic communication with schematic figures. In: Anderson, M., Cheng, _.P., Haarslev, V. (eds.) Diagrams 2000. LNCS (LNAI), vol. 1889, pp. 221–230. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-44590-0_21

38. Boekelder, A., Steehouder, M.: Selecting and switching: some advantages of diagrams over tables and lists for presenting instructions. IEEE Trans. Professional Commun. **41**, 229–241 (1998)
39. Haisjackl, C., et al.: Understanding declare models: Strategies, pitfalls, empirical results. Software Syst. Model. **15**, 325–352 (2016)
40. Straub, D.W., Boudreau, M.-C., Gefen, D.: Validation guidelines for is positivist research. Commun. Assoc. Inf. Syst. **13**, 380–427 (2004)