The impact of modularization on the understandability of declarative process models: A research model*

Amine Abbad Andaloussi[1], Pnina Soffer[2], Tijs Slaats[3], Andrea Burattin[1] and Barbara Weber[4]

[1] Software and Process Engineering, Technical University of Denmark
2800 Kgs. Lyngby, Denmark – amab@dtu.dk
[2] Department of Information Systems, University of Haifa
3498838, Haifa, Israel
[3] Department of Computer Science, University of Copenhagen
2100 København Ø, Denmark
[4] Institute of Computer Science, University of St. Gallen
9000 St. Gallen, Switzerland

Abstract. Process models provide a blueprint for process execution and an indispensable tool for process management. Bearing in mind their trending use for requirement elicitation, communication and improvement of business processes, the need for understandable process models becomes a must. In this paper, we propose a research model to investigate the impact of modularization on the understandability of declarative process models. We design a controlled experiment supported by eye-tracking, electroencephalography (EEG) and galvanic skin response (GSR) to appraise the understandability of hierarchical process models through measures such as comprehension accuracy, response time, attention, cognitive load and cognitive integration.

Keywords: Modularization, Understandability, Declarative process models, DCR Graphs, Neurophysiological experiment

1 Introduction

Process digitization begins with a set of process specifications, which are represented as process models and then implemented as part of a process-aware information system (PAIS). Process models serve both enactment and management purposes [1]. They provide a blue-print for process execution – but can also be used to elicit, communicate, and improve the quality of business processes. Designing understandable models is crucial for attaining these purposes.

Processes are represented using languages from the imperative-declarative paradigm (for a literature review, see [2]). Imperative languages clearly depict the different executions supported by the process, and this makes them relatively easy to comprehend.

* Work supported by the Innovation Fund Denmark project *EcoKnow* (7050-00034A)

However, their support is limited to rigid specifications, which is suitable for repetitive and structured processes, but not for ones where flexibility is an inherent requirement (e.g., knowledge-intensive processes). This need is satisfied by declarative languages, which allow flexible process specifications [1,3] but are difficult to understand [1].

The use of declarative languages results often in complex models, which are hard to interpret and maintain by humans. Comprehending the human cognitive processes impacting the understandability of process models, paves the way toward the adoption of modeling practices, enhancing the comprehension of declarative models and thus supporting their use for management purposes [4, 5]. In the field of cognitive psychology, existing research has shown the limited capacity of the human working memory [6]. Accordingly, this limited resource must be utilized in a way such that a reader can easily interpret the constraints of the model and extract the required information efficiently. Cognitive load is a common indicator of the use of working memory [7]. Whenever a reader is introduced to a process model, 3 types of load emerge: intrinsic load, extraneous load and germane load [8]. Intrinsic load relates to the inherent complexity of the process, whereas extraneous load raises from the way the process is represented. Germane load, in turn, emerges from the effort invested by the reader to comprehend and reason about the model. While the intrinsic load is changing from one process to another, the extraneous load can be reduced by refining the model representation, hence leaving more capacity for the germane load to emerge and thus an increased ability for the reader to comprehend the process model.

Our research taps into the representation of process models. Considering the intrinsic complexity of processes and the different ways in which entangled constraints could interact in declarative models [9], readers can exceed their working memory capacity and thus limit their understanding of the model. Modularization could reduce the complexity of process models by decomposing them into sub-processes. Modularization has been investigated in computer programming [10], conceptual modeling [11–13] and process modeling [4, 14-16]. With regards to declarative languages, a qualitative study [16] suggests that abstraction and fragmentation are two opposing forces affecting the understandability of modularized process models expressed in the Declare language [17]. Grounded in the theory of cognitive fit [18], we use local and global tasks to perceive the influence of abstraction and fragmentation. As part of our ongoing research, we design a controlled experiment supported by eye-tracking, electroencephalography (EEG) and galvanic skin response (GSR) to investigate end-users' understandability through measures such as comprehension accuracy, response time, attention, cognitive load and cognitive integration. We study modularization in the context of declarative models expressed using the Dynamic Condition Response (DCR graphs) language [19]. We focus on this language in particular because of the availability of industrial-level tools [20] and a wide array of documented real-world applications [21, 23]. Section 2 presents the theoretical background, Section 3 introduces our research method, and Section 4 concludes the paper.

2 Theoretical Background

Modularization and hierarchy. Modularization denotes the degree to which a system can be devised into independent, composable units [24]. Information hiding is a branch of modularization [24]. In computer programming, it denotes the distinction between the interface and the implementation of a system, which in turn allows refining the implementation without invalidating the interface. In that sense, an interface is seen as an abstraction of the implementation, allowing to reason about the system on a more abstract level [24]. The same principle holds with process models. Hereby, an interface is equivalent to a high-level model providing an overview of the process, while implementations are just like sub-processes describing the low-level details of the process. Information hiding is better supported through hierarchy. Dawkins [25] discusses the notion of "hierarchical reductionism". He used hierarchy to organize complex systems into units, such that each unit of the hierarchy abstracts the details of its subsequent units (placed one level down in the hierarchy) and concretize the details of its former units (placed one level up in the hierarchy). In process modeling, hierarchical reductionism takes information hiding beyond a single abstraction level, making it possible to define sub-processes within a sub-process itself.

Hierarchy was introduced to Declare in [16] and expanded upon in [26]. In DCR, different forms of decomposing models have been introduced [27–29]. For our study we focus on a special type of hierarchy referred to as single-instance sub-processes, where a sub-process is a non-atomic activity which contains embedded activities and constraints that need to be completed before the sub-process can execute [30], similar to what was done for Declare [16,26].

Impact of modularization. Modularization has been widely investigated in the literature [10–15]. However, its impact on understandability remains inconclusive and hard to generalize (for a systematic review, see [4]). Hierarchy is claimed to abstract the details in the process model and provide better means to cope with complexity [15,16]. The notion of complexity has been studied in the computer programming literature [31]. Structural complexity is among the different types of complexity identified in [32]. It denotes the complexity associated with the representation of the artifact (e.g., process model, source code). Structural complexity has been empirically investigated in model comprehension [33]. Using different representations, existing research has shown a significant impact of structural complexity on users' cognitive load (e.g., [34]), comprehension accuracy and response time (e.g., [35]).

Petrusel and Mendling [36] show that users do not typically focus on the entire model but rather limit their attention to only relevant parts of the model. In that vein, abstraction could presumably focus readers' attention on relevant sub-processes. Attention has been investigated in model comprehension. More specifically, recent research has shown how different process representations guide readers' attention towards the relevant parts of the model, and how increased attention accounts for comprehension accuracy [37].

Besides, increased modularization can cause fragmentation and thus requires the reader to continually switch attention between the sub-processes of the model, which leads to the split-attention effect [16,38]. This effect happens when readers are required to distribute their attention between different sources of information (e.g., different sub-processes) [8]. The split-attention effect can distract readers' ability to focus on relevant aspects, which in turn requires investing additional mental effort when solving a task [8]. In model comprehension, research suggests that splitting the process control-flow and the underlying business rules could presumably influence the reader comprehension accuracy, response time and cognitive load [39].

Additionally, the split attention effect requires the reader to mentally integrate the information extracted from the model sub-processes to understand the process. In model comprehension, recent research linked cognitive integration to comprehension accuracy [37]. In the same research, it has been shown that the visual associations exhibited by readers when making sense of the different components of a model can be used as an indicator for cognitive integration.

Impact of task type. The influence of task type on the comprehension of visual representations has been shown in several contexts [40–43]. Following the cognitive fit theory [18], a fit between the task type and the information exposed by the visual representation (e.g., model) is associated with better performance. The impact of the task type has been widely investigated in model comprehension studies [40–43]. Vessey and Galletta [40] identified symbolic tasks (i.e., addressing discrete data values) and spatial tasks (i.e., addressing relationships in data) and proclaimed that tabular representations create better fit for symbolic tasks, while graphical representations make a better fit with spatial tasks. Likewise, Ritchi et al. [41] discerned schema-based tasks (i.e., can be solely completed from the model) and non-schema-based tasks (i.e., addressing aspects beyond the explicit information exposed in the model) and investigated the extent to which graphical and textual representations fit for different tasks. Dun and Grabski [43], in turn, integrated the notion of localization introduced by Larkin and Simon [42] and asserted that the more local the information allowing to solve a particular task, the better is the performance, making the distinction between *local* and *global* tasks a pertinent factor defining the understandability of visual representations. Building upon the tasks' classification of Dun and Grabski [43], we consider the proposed distinction as a relevant dimension for model comprehension.

Interaction between modularization and task type. In process modeling, the interaction between both factors has been raised in Zugal's literature review [4]. The author suggested that hierarchical models are more efficient for solving local tasks than global tasks. However, he did not provide a clear theoretical understanding of how a fit between the task type and the model representation could affect understandability. To fill this gap, we turn to the theory of cognitive fit to explain how local and global tasks could indeed influence the comprehension of hierarchical process models.

We postulate that modularization supports local tasks through abstraction: the process is divided into sub-processes, and the task addresses specifications within a single sub-process, which in turn creates a good fit between the visual representation of the process and the task at hand. Conversely, modularization complicates global tasks due to fragmentation: the process is divided into sub-processes while the task requires continuous integration of information from different sub-processes, causing a mismatch between the representation of the process and the task at hand.

To further explore the interaction between modularization and task type, we refer to the discussion of structural integration by Gilmore and Green [44], which underlines that understandable representations are those where information can be easily located and transferred to working memory. Gilmore and Green [44] also showed that mental representations preserve some features of the used notation. Hereby, a hierarchical model would produce a more or less "structured" mental model compared to a flat model. Local tasks, in turn, could benefit from the structure of the mental model, which facilitates the retrieval and the transfer of information to working memory. Conversely, additional load could emerge when solving global tasks, as the reader is required to disregard the acquired structure and rather perceive the interplay between activities lying within different sub-processes. Herein, the imposed structure adds additional burden to the reader. It is nonetheless worthwhile to mention that the mental model could also be affected by other factors such as background and experience, which could, in turn, pre-define the way the user acquires and incorporates new information.

3 Research Method

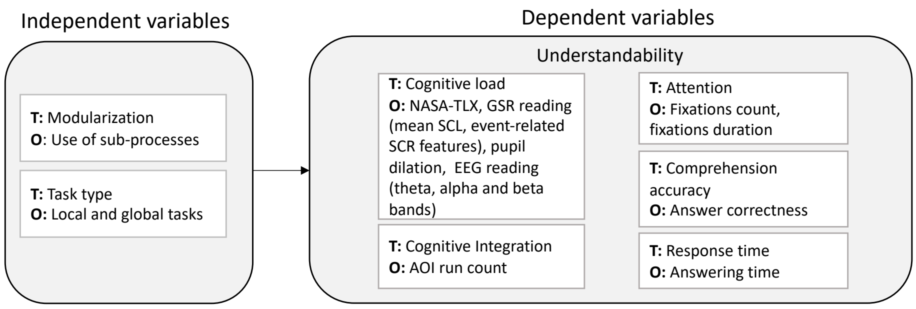

Fig. 1. Envisioned Research Model. T and O refer to the theoretical constructs (T) and their operationalization (O) respectively

Research model. The theoretical background presented in Section 2 suggests that abstraction and fragmentation drive the understandability of modularized process models. Our research aims at providing empirical evidence supporting this proposition. Following a 2x2 factorial design [45], we define *Modularization* (levels: *modularized models versus flat models*) and *Task type* (levels: *local tasks versus global tasks*) as two distinct factors. These factors are expected to impact the user understanding of the model. *Understandability* is a cognitive concept, created in the reader cognition and, thus, not directly tangible [34]. Eventually, it can be estimated only using indirect constructs. Aranda et al. [46] evoke difficulty (i.e., cognitive load), correctness (i.e., comprehension accuracy), and time (i.e., response time) as indicators of understandability. Motivated by the existing literature on model comprehension [36,37,39], we additionally consider cognitive integration and attention. Our research model is summarized in Figure 1.

Following the theoretical foundations set in Section 2 and the requirements for a factorial design, we formulate our first *main effect* hypothesis as follows: H_1**: There is a significant difference in the understandability of modularized and flat process models**, while we formulate our second main hypothesis as follows: H_2**: There is a significant difference in the understandability of local and global tasks**. In addition, we formulate our interaction effect hypothesis as follows H_3**: Modularized models are more understandable for local than for global tasks**.

The theoretical constructs depicted in Figure 1 are operationalized as follows. With regards to our independent variables, modularization is operationalized using modularized models with sub-processes and flat models without sub-processes, while the task type is operationalized using local and global tasks. Local tasks are meant to make use of abstraction without being affected by fragmentation. They address local aspects requiring to perceive the interplay between activities belonging to the same sub-process. In contrast, global tasks are designed to neglect abstraction and rather cause fragmentation. They address global aspects requiring to perceive the interplay between activities belonging to different sub-processes.

The dependent variables covered by our research model are operationalized using *subjective, neurophysiological, behavioral and performance* measures. To measure cognitive load, we use a subjective rating of cognitive load i.e., *NASA-TLX* [47]. In addition, we use a set of physiological measures. Namely, we rely on the *GSR reading* to extract *skin conductance level* (*SCL*, also know as tonic signal) and *skin conductance response* (*SCR*, also known as phasic signal) [48]. As measures, we compute the *mean SCL* (relative to the baseline) and extract *event-related SCR features* including *number of peaks*, *peak amplitude*, and *area under curve* [49]. Similar features are used to measure cognitive load of users when solving tasks with different levels of difficulty [49, 50]. Moreover, we monitor *pupil dilation* through changes in pupil diameter across different tasks, which, in turn, is used to estimate cognitive load [51]. Furthermore, we perform a frequency-based analysis of EEG bands i.e., *theta, alpha and beta* powers [52] to track the changes in cognitive load. Our design also deploys behavioral measures. Based on the notions of fixation (i.e., the timespan where the eye remains still at a specific position of the stimulus [53]) and areas of interest (AOI, i.e., a grouping of fixations covering a specific area of the stimulus [53]), we use the *AOI run count* (i.e., number of entry and exists to AOIs [37]) to evaluate the participants' cognitive integration, and we use *fixations count and duration* to measure attention. Moreover, we rely on performance measures such as *answer correctness* and *answering time* to analyze the participants' comprehension accuracy and response time.

Material. The material meant for the experiment comprises a set of information-equivalent models represented with and without modularization and a set of local and global tasks allowing to test the impact of abstraction and fragmentation respectively.

Based on the guidelines and recommendations in [54], we define a set of requirements addressing the design of *models* and *tasks*. By following these requirements, we aim at reducing the effects of the confounding factors threatening the validity of our study.

With regards to the models, the insights of Zimoch et al. [54] provide a good starting point to define a uniform visual layout applying to all process models. Herein, we carefully set up a layout where activities are oriented from left to right, depending on their likely order of execution. We also avoid crossing arrows as much as possible, ensure proper spacing between the model's elements and name activities consistently. Besides, we address the complexity of the models to ensure that all of them have the same number of activities and deploy similar constraint patterns. Moreover, similar to [55], we use anonymized models (where activities are labeled with random letters) to avoid the influence of the domain. Last but not least, we append a legend describing the DCR semantics to all models, assuring that participants are able to interpret the DCR notation represented in the models.

As for the design of tasks, we use local tasks addressing the interplay between activities within a single sub-process, and global tasks addressing the interplay between activities located in different sub-processes. Within each task, we ask one question. We design questions to reflect the use of process models in practice. Similar to [5, 16, 44], we formulate questions addressing: the presence and absence of constraints in the model, order of activities and validity of process executions. We make dichotomous questions. Nevertheless, to reduce the chances of guessing answers, we ask participants to justify their answers verbally and allow them to skip answering questions which they are unsure about.

To ensure that our models and tasks are representative, there are a couple of measures which we take into consideration. We design our models in DCR graphs that is a known declarative language with academic and industrial tool-support [20] deployed by several private and public institution in Denmark[1]. Additionally, we rely on the recommendations of experts in DCR graphs to provide models covering a large subset of constraint patterns which are frequently used in practice. As for the tasks, we build upon existing literature [5, 16, 44] and use different types of questions reflecting different scenarios where process models are used in practice. Similar to [44], the variety of our questions is not meant to predict differences between different types of questions, but rather to ensure that our findings could be generalized.

We propose different manipulations to cover all the conditions where modularized and flat process models are used to solve local and global tasks. We group the models into sets. Each set S_i ($i \in \mathbb{N}$) is composed of (1) a Process P_i modeled in two variants: one modularized M_{i_m} and one flat M_{i_f} and (2) two tasks: one local T_{i_l} and one global T_{i_g}. Combining models and tasks, each set contains the following: $\{M_{i_m}T_{i_l}$, $M_{i_m}T_{i_g}$, $M_{i_f}T_{i_l}$, $M_{i_f}T_{i_g}\}$. Afterwards, we group the sets into collections. Each Collection is composed of 4 distinct sets, where the respective tasks within each set address presence, absence, order or execution questions. Figure 2 shows an example of a collection.

[1] see https://dcrsolutions.net

$M_{1_m}T_{1_l}$	$M_{1_m}T_{1_g}$	$M_{2_m}T_{2_l}$	$M_{2_m}T_{2_g}$	$M_{3_m}T_{3_l}$	$M_{3_m}T_{3_g}$	$M_{4_m}T_{4_l}$	$M_{4_m}T_{4_g}$
$M_{1_f}T_{1_l}$	$M_{1_f}T_{1_g}$	$M_{2_f}T_{2_l}$	$M_{2_f}T_{2_g}$	$M_{3_f}T_{3_l}$	$M_{3_f}T_{3_g}$	$M_{4_f}T_{4_l}$	$M_{4_f}T_{4_g}$
Set 1: presence questions		Set 2: absence questions		Set 3: order questions		Set 4: executions' questions	

Fig. 2. Example of a collection

Participants. Confounding factors related to the subjects of the study (i.e., participants) represent significant threats to validity if not handled correctly. Following the recommendations in [54], we limit our study to novice participants with no or very limited experience with DCR graphs. Doing so, we ensure that the observed effects are due to our manipulations rather than personal factors associated with participants' expertise. Moreover, we perform a screening of all participants checking their physical ability to participate in neuropsychological experiments. Furthermore, to guarantee that all participants are equally trained, we provide a uniform and comprehensive familiarization to all participants, so they have the necessary background about the investigated theory and experiment procedure. By the end of the familiarization, we provide a short quiz to evaluate the technical aptitude of participants.

Experiment design. We use a within-subject experiment design where each participant is exposed to all conditions. We motivate our design by the idiosyncratic nature of many eye-tracking, EEG and GSR measures [53, 56, 57] (i.e., each participant has her own baseline), which in turn requires a within-subject comparison of the different experiment's conditions. A possible threat to this design might be associated with a learning effect and fatigue during the experiment, which could influence the results of the within-subject comparison. To mitigate these effects, we randomize the experiment's tasks, ensuring that participants receive tasks in different orders.

To ensure a good data quality, we follow existing guidelines on collecting clean eye tracking [53], GSR [56] and EEG [57, 58] data. Before the experiment, a screening form (asking about age range, gender, proficiency in English, vision issues, neurological diseases e.g., epilepsy, attention disorder, handedness and allergies) is sent to check the participant's physical ability to participate in neuropsychological experiments and obtain relative information allowing to determine the most suitable EEG cap size for her. Upon approval, an invitation is sent and information regarding what to avoid (e.g., mascara, eyelash extensions, reflective glasses, artificial hair products, hair pins and clips) are shared. In addition, we ask the participant to wash her hair and dry it completely prior to the experiment day. We prepare and set up electrodes in the EEG recording cap, verify the light conditions and the temperature in the lab before receiving the participant.

We start each experiment session with a familiarization and a quiz. Afterwards, we collect demographic and expertise information from the participant. Next, we seat the participant in front of the eye-tracking station comfortably by (i) adjusting the chair and the table to the participant's preferences, while guaranteeing that the eye tracker can still capture her eyes, (ii) ensuring that feet are flat on the ground, and (iii) adjusting the lumbar support of the chair. Then, we place the EEG cap, and adjust the GSR electrodes on the non-dominant hand. We instruct the participant to breathe normally, avoid chewing and tensing her jaw, keep the hand with the GSR electrodes stable, not to move her head, and limit all other body movements. We calibrate the eye-tracking, GSR and EEG devices and check the quality of the different signals. To keep track of the participant's verbal utterances (i.e., the justifications of her answers), we record the audio for the whole data collection part.

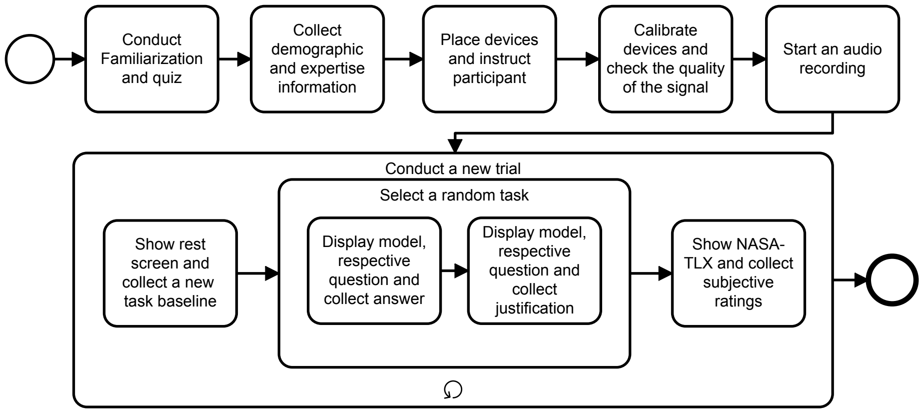

Fig. 3. Data collection procedure

The data collection is composed of a set of trials. During each trial, we show a grey rest screen for 1-2 minutes and collect new baseline measurements. Afterwards, we select a random task. Here, we display the model and respective question, and then collect the answer from the participant. Next, we display the same model and respective question again, but this time, we ask the participant to justify her answer. By doing so, we can differentiate the initial response time from the time used to justify the answer verbally. Finally, we provide the NASA-TLX questionnaire to obtain a subjective rating of cognitive load. Figure 3 depicts a BPMN [59] model summarizing the experiment procedure.

4 Conclusion

This paper describes a research model aimed at investigating the impact of modularization on the understandability of declarative process models. As future work, we are planning to concretize this model and report empirical evidence about the impact of local and global tasks on the understandability of hierarchical process models in DCR graphs.

References

1. Manfred Reichert and Barbara Weber. Enabling Flexibility in Process-Aware Information Systems. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
2. Amine Abbad Andaloussi, Andrea Burattin, Tijs Slaats, Ekkart Kindler, and Barbara Weber. On the declarative paradigm in hybrid business process representations: A conceptual framework and a systematic literature study. Information Systems, 91:101505, 2020.
3. Tijs Slaats. Declarative and hybrid process discovery: Recent advances and open challenges. Journal on Data Semantics, pages 1–18, 2020
4. Stefan Zugal. Applying Cognitive Psychology for Improving the Creation, Understanding and Maintenance of Business Process Models. 2013.
5. Amine Abbad Andaloussi, Andrea Burattin, Tijs Slaats, C.M Anette Petersen, T. Thomas Hildebrandt, and Barbara Weber. Exploring the Understandability of a Hybrid Process Design Artifact Based on DCR Graphs. In Enterprise, Business Process and Information Systems Modeling, BPMDS 2019, pages 1–15, 2019.
6. Alan Baddeley. Working memory: Theories, models, and controversies. Annual review of psychology, 63:1–29, 2012.
7. Fang Chen, Jianlong Zhou, Yang Wang, Kun Yu, Syed Z Arshad, Ahmad Khawaji, and Dan Conway. Robust Multimodal Cognitive Load Measurement, pages 13–32. 2016.
8. John Sweller. Cognitive load theory. In Psychology of learning and motivation, volume 55, pages 37–76. Elsevier, 2011.
9. M. Pesic. Constraint-based workflow management systems: shifting control to users. PhD thesis, Department of Industrial Engineering & Innovation Sciences, 2008.
10. D. L. Parnas. On the criteria to be used in decomposing systems into modules. Commun. ACM, 15(12):1053–1058, December 1972.
11. Peretz Shoval, Revital Danoch, and Mira Balabam. Hierarchical entity-relationship diagrams: the model, method of creation and experimental evaluation. Requirements Engineering, 9(4):217–228, 2004.
12. Jose A Cruz-Lemus, Marcela Genero, and Mario Piattini. Using controlled experiments for validating uml statechart diagrams measures. In Software Process and Product Measurement, pages 129–138. Springer, 2007.
13. J Cruz-Lemus, Marcela Genero, Mario Piattini, and Ambrosio Toval. Investigating the nesting level of composite states in uml statechart diagrams. Proc. QAOOSE, 5:97–108, 2005.
14. Florian Johannsen and Susanne Leist. Wand and weber's decomposition model in the context of business process modeling. Business & Information Systems Engineering, 4(5):271–286, 2012.
15. Hajo A Reijers, Jan Mendling, and Remco M Dijkman. Human and automatic modularizations of process models to enhance their comprehension. Information Systems, 36(5):881–897, 2011.
16. Stefan Zugal, Pnina Soffer, Cornelia Haisjackl, Jakob Pinggera, Manfred Reichert, and Barbara Weber. Investigating expressiveness and understandability of hierarchy in declarative business process models. Software & Systems Modeling, 14(3):1081–1103, 2015.
17. Maja Pesic, Helen Schonenberg, and Wil M.P. van der Aalst. DECLARE: Full Support for Loosely-Structured Processes. In 11th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007), pages 287–287. IEEE, oct 2007.

18. Iris Vessey. Cognitive fit: A theory-based analysis of the graphs versus tables literature. Decision sciences, 22(2):219–240, 1991.
19. Thomas T. Hildebrandt and Raghava Rao Mukkamala. Declarative Event-Based Workflow as Distributed Dynamic Condition Response Graphs. Electronic Proceedings in Theoretical Computer Science, 69:59–73, 2011.
20. Morten Marquard, Muhammad Shahzad, and Tijs Slaats. Web-based modelling and collaborative simulation of declarative processes. In Business Process Management, pages 209–225, Cham, 2015. Springer International Publishing.
21. Søren Debois, Thomas Hildebrandt, Morten Marquard, and Tijs Slaats. Hybrid Process Technologies in the Financial Sector: The Case of BRFkredit, pages 397–412. Springer International Publishing, Cham, 2018.
22. S. Debois and T. Slaats. The analysis of a real life declarative process. In 2015 IEEE Symposium Series on Computational Intelligence, pages 1374–1382, Dec 2015.
23. Tijs Slaats, Raghava Rao Mukkamala, Thomas Hildebrandt, and Morten Marquard. Exformatics declarative case management workflows as dcr graphs. Business Process Mangement, pages 339–354, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.
24. Klaus Ostermann, Paolo G. Giarrusso, Christian Kastner, and Tillmann Rendel. Revisiting information hiding: Reflections on classical and nonclassical modularity. In ECOOP 2011 – Object-Oriented Programming, pages 155– 178, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.
25. Richard dawkins, the blind watchmaker, ww norton and company, new york and london (1986), p. xiii, 332 pp, isbn: 0-393-02216-1, 1986.
26. Tijs Slaats, Dennis M. M. Schunselaar, Fabrizio M. Maggi, and Hajo A. Reijers. The semantics of hybrid process models. In On the Move to Meaningful Internet Systems: OTM 2016 Conferences, pages 531–551, Cham, 2016. Springer International Publishing.
27. Thomas Hildebrandt, Raghava Rao Mukkamala, and Tijs Slaats. Nested dynamic condition response graphs. In Proceedings of Fundamentals of Software Engineering (FSEN), April 2011.
28. Søren Debois, Thomas Hildebrandt, and Tijs Slaats. Hierarchical declarative modelling with refinement and sub-processes. In Business Process Management, pages 18–33, Cham, 2014. Springer International Publishing.
29. Søren Debois, Thomas T. Hildebrandt, and Tijs Slaats. Replication, refinement & reachability: complexity in dynamic condition-response graphs. Acta Informatica, 55(6):489–520, September 2018.
30. J. Costa Seco, S. Debois, T. Hildebrandt, and T. Slaats. Reseda: Declaring live event-driven computations as reactive semi-structured data. In 2018 IEEE 22nd International Enterprise Distributed Object Computing Conference (EDOC), pages 75–84, Oct 2018.
31. SN Cant, D Ross Jeffery, and Brian Henderson-Sellers. A conceptual model of cognitive complexity of elements of the programming process. Information and Software Technology, 37(7):351–362, 1995.
32. Brian Henderson-Sellers. Object-oriented metrics: measures of complexity. Prentice-Hall, Inc., 1995.
33. Constantin Houy, Peter Fettke, and Peter Loos. Understanding understandability of conceptual models–what are we actually talking about? In International Conference on Conceptual Modeling, pages 64–77. Springer, 2012.
34. Kathrin Figl and Ralf Laue. Cognitive complexity in business process modeling.

In International Conference on Advanced Information Systems Engineering, pages 452–466. Springer, 2011.
35. Laura Sanchez-Gonzalez, Francisco Ruiz, Felix Garcia, and Jorge Cardoso. Towards thresholds of control flow complexity measures for bpmn models. In Proceedings of the 2011 ACM symposium on Applied computing, pages 1445–1450, 2011.
36. Razvan Petrusel and Jan Mendling. Eye-tracking the factors of process model comprehension tasks. In Advanced Information Systems Engineering, pages 224–239, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.
37. Palash Bera, Pnina Soffer, and Jeffrey Parsons. Using eye tracking to expose cognitive processes in understanding conceptual models. MIS Quarterly, 43(4):1105–1126, 2019.
38. John Sweller and Paul Chandler. Why some material is difficult to learn. Cognition and instruction, 12(3):185–233, 1994.
39. Wei Wang, Marta Indulska, Shazia Sadiq, and Barbara Weber. Effect of linked rules on business process model understanding. In Business Process Managemen, BPM 2017, pages 200–215. 2017.
40. Iris Vessey and Dennis Galletta. Cognitive fit: An empirical study of information acquisition. Information systems research, 2(1):63–84, 1991.
41. Hamzah Ritchi, Mieke J Jans, Jan Mendling, and Hajo Reijers. The influence of business process representation on performance of different task types. Journal of Information Systems, 2019.
42. Jill H Larkin and Herbert A Simon. Why a diagram is (sometimes) worth ten thousand words. Cognitive science, 11(1):65–100, 1987.
43. Cheryl Dunn and Severin Grabski. An investigation of localization as an element of cognitive fit in accounting model representations. Decision Sciences, 32(1):55–94, 2001.
44. David J. Gilmore and Thomas R. G. Green. Comprehension and recall of miniature programs. International Journal of Man-Machine Studies, 21(1):31–48, 1984.
45. Claes Wohlin, Per Runeson, Martin Host, Magnus C Ohlsson, Bjorn Regnell, and Anders Wesslen. Experimentation in software engineering. Springer Science & Business Media, 2012.
46. Jorge Aranda, Neil Ernst, Jennifer Horkoff, and Steve Easterbrook. A framework for empirical evaluation of model comprehensibility. In International Workshop on Modeling in Software Engineering (MISE'07: ICSE Workshop 2007). IEEE, 2007.
47. Sandra G Hart and Lowell E Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In Advances in psychology, volume 52, pages 139–183. Elsevier, 1988.
48. Jason J Braithwaite, Derrick G Watson, Robert Jones, and Mickey Rowe. A guide for analysing electrodermal activity (eda) & skin conductance responses (scrs) for psychological experiments. Psychophysiology, 49(1):1017–1034, 2013.
49. Thomas Fritz, Andrew Begel, Sebastian C Muller, Serap Yigit-Elliott, and Manuela Zuger. Using psycho-physiological measures to assess task difficulty in software development. In Proceedings of the 36th international conference on software engineering, pages 402–413, 2014.
50. Nargess Nourbakhsh, Fang Chen, Yang Wang, and Rafael A Calvo. Detecting users' cognitive load by galvanic skin response with affective interference. ACM Transactions on Interactive Intelligent Systems (TiiS), 7(3):1–20, 2017.
51. Eckhard H Hess and James M Polt. Pupil size in relation to mental activity during simple problem-solving. Science, 143(3611):1190–1192, 1964.

52. Ernst Niedermeyer and FH Lopes da Silva. Electroencephalography: basic principles, clinical applications, and related fields. Lippincott Williams & Wilkins, 2005.
53. K. Holmqvist, M. Nystrom, R. Andersson, R. Dewhurst, H. Jarodzka, and J. van de Weijer. Eye Tracking: A comprehensive guide to methods and measures. OUP Oxford, 2011.
54. Michael Zimoch, Rudiger Pryss, Johannes Schobel, and Manfred Reichert. Eye tracking experiments on process model comprehension: lessons learned. In Enterprise, Business-Process and Information Systems Modeling, pages 153–168. Springer, 2017.
55. Paul Pichler, Barbara Weber, Stefan Zugal, Jakob Pinggera, Jan Mendling, and Hajo A Reijers. Imperative versus declarative process modeling languages: An empirical investigation. In International Conference on Business Process Management, pages 383–394. Springer, 2011.
56. Imotions. Galvanic skin response: The complete pocket guide, 2017.
57. Imotions. Electroencephalography: The complete pocket guide, 2019.
58. Gernot R Muller-Putz, Rene Riedl, and Selina C Wriessnegger. Electroencephalography (EEG) as a research tool in the information systems discipline: Foundations, measurement, and applications. CAIS, 37:46, 2015.
59. Object Management Group OMG. Business Process Modeling Notation V 2.0. 2006.