A Qualitative Comparison of Approaches Supporting Business Process Variability*

Victoria Torres[1], Stefan Zugal[2], Barbara Weber[2], Manfred Reichert[3], Clara Ayora[1], and Vicente Pelechano[1]

[1] Universitat Politècnica de València, Spain
{vtorres,cayora,pele}@pros.upv.es
[2] University of Innsbruck, Austria
{stefan.zugal,barbara.weber}@uibk.ac.at
[3] University of Ulm, Germany
manfred.reichert@uni-ulm.de

Abstract. The increasing adoption of Process-Aware Information Systems, together with the *reuse* of process knowledge, has led to the emergence of process model repositories with large process families, i.e., collections of related process model variants. For managing such related model collections two types of approaches exist. While *behavioral* approaches take supersets of variants and derive a process variant by hiding and blocking process elements, *structural* approaches take a base process model as input and derive a process variant by applying a set of change operations to it. However, at the current stage no framework for assessing these approaches exists and it is not yet clear which approach should be better used and under which circumstances. Therefore, to give first insights about this issue, this work compares both approaches in terms of understandability of the produced process model artifacts, which is fundamental for the management of process families and the *reuse* of their contained process fragments. In addition, the comparison can serve as theoretical basis for conducting experiments as well as for fostering the development of tools managing business process variability.

1 Introduction

The increasing adoption of Process-Aware Information Systems (PAISs) by industry has led to large collections of process models in a variety of application domains. Despite the particularities found in specific domains, many of these models share common parts of their definition (e.g., activities). We denote such related process variant models as *process family* in the following.

To properly handle large process families (i.e., avoid redundancies, foster reusability, and reduce modeling efforts) several proposals have been developed in recent years (e.g., [1], [2], [3]), which can be classified as either behavioral or structural approaches. *Behavioral* approaches represent all members of the

* This work has been developed with the support of MICINN under the project EVERYWARE TIN2010-18011.

family within the same model artifact, capturing both, commonalities and particularities of all process variants. In turn, *structural* approaches use different artifacts to represent the family e.g., by using a base model to which structural changes such as inserting, deleting, or moving activities may be applied to derive process variants. To foster reusability of process model families, understandability of the created artifacts is essential. So far, however, no experimental insights regarding *quality* aspects (e.g., *understandability*) are available and it is not clear under which circumstances the use of one approach is more appropriate than the other. Since the results of assessing a process model's understandability significantly depend on the specific understandability tasks [4, 5], we have structured the comparison of both approaches along a specific comprehension task, i.e., the extraction of a process variant from a configurable model, elaborating on the process followed by a model reader to accomplish such task. In addition, we use cognitive psychology as a tool for explaining the differences between the two approaches. This comparison will provides us the theoretical basis for conducting experiments as well as for fostering the development of tools for managing variability in business processes.

Sect. 2 presents a process family from the film industry. Sect. 3 provides basic notions and introduces the behavioral and structural approaches. Sect. 4 describes concepts from cognitive psychology that will be used in Sect. 5 to assess their understandability. Sect. 6 then presents an overview of related work. Finally, Sect. 7 concludes the paper and gives an outlook.

2 Example of a Process Family

As running example, we consider a modified version of a process family from the film industry for editing a screen project, which varies depending on the shooting media and delivery media used [6]. First, footage is received, either in *tape*, *film*, or *tape and film* and *prepared for edition* depending on the shooting medium. Then *offline edition* is performed. Next, the cut stage is performed through *online edit* (if the *shooting medium* is *tape*), through *negmatching* (in the case of *film*), or through both cuts (when *shooting media* is *tape and film*). After this point, the finishing on a delivery medium phase starts. For this purpose, the delivery media (e.g., *tape, film, tape and film, or new medium*) must be chosen. Now, depending on the delivery medium chosen different variants exist. For example, when shooting media is *film* or *tape and film* and cutting has been performed through negmatching, a *finish on film* has to be performed to maintain the quality of the delivery medium. On the contrary, if the cutting is performed through *online editing* and *film* or *tape and film* is the delivery medium, *record digital film master* needs to be mandatorily performed to transfer the editing results to *film*. Similarly, *telecine transfer* will be performed only if negmatching is performed previously and the expected delivery format is *tape* or *new medium*. Finally, if neither *tape* or *film* have been chosen activity *finish on new medium* must be performed.

3 Approaches for Modeling BP Variability

To properly represent variability in a BP model, it is important to define (1) what parts of the BP model may vary according to a specific context, (2) what alternatives exist in each of those parts, and (3) which conditions make these alternatives being selected. The first issue refers to the precise identification of the parts being subject to variation, which are commonly known as *variation points*. The second issue refers to the different alternatives that exist for all those variation points. In addition, some models may require the definition of *relationships* (e.g., inclusion, exclusion) between alternative process fragment from different variation points. The third issue refers to the *context* of these variations, which is usually represented by a set of variables gathered in a *context model* in which the BP model is used. This subsection presents two different approaches targeted at the representation of such process families, i.e., behavioural and structural.

Behavioural Approach. The behavioral approach represents a process family in a single artifact, known as *configurable process model* capturing both the commonalities and particularities of the *process variants* reflecting all possible behavior. In the following we take C-EPC [2] as the representative proposal since it constitutes the most well-known and mostly cited proposal. C-EPC extends EPC with configurable elements (i.e., *configurable nodes* and *configuration requirements*) to explicitly model variability. Fig. 1 illustrates the configurable process model representing the postproduction process. On the one hand, *configurable nodes* (i.e., connectors and functions) are represented graphically with thick solid borders and define variations points in the model where different alternatives may exist. Specifically, *configurable functions* (e.g., activity *telecine transfer* in Fig. 1) can be configured as ON (i.e., function is kept in the model), OFF (i.e., function is removed from the model), or OPT (i.e., conditional branching is included in the model deferring the decision to run-time. *Configurable connectors*, in turn, can be configured to an equally or more restrictive connector. For example, a configurable OR can configured as a regular OR (not applying any restrictions), or can restrict its behaviour by configuring it as an XOR (i.e., selecting one of the outgoing/incoming alternatives), AND (i.e., selecting all of the outgoing/incoming alternatives), or SEQ_n (i.e., reducing the alternatives for the configuration of the connector to just one of its outgoing/incoming sequences). On the other hand, *configuration requirements* are graphically represented as tags attached to *configurable nodes* and formalize, by means of logical predicates, domain constraints related to the attached *nodes*. The configuration of the node will the be made based on the evaluation of the attached *configuration requirements* (e.g., *req. 5* requires that activity *edit footage online* is chosen when shooting medium is *tape*). However, *configurable nodes* not always have requirements attached to them (since they are only needed when there is a constraint regarding their configuration). In this case the configurable node is transformed into a regular one, maintaining the behaviour of the original connector and deferring the configuration decision to run-time.

Structural Approach. This approach proposes a gradual construction of the process family by modifying the structure of a specific process variant (called

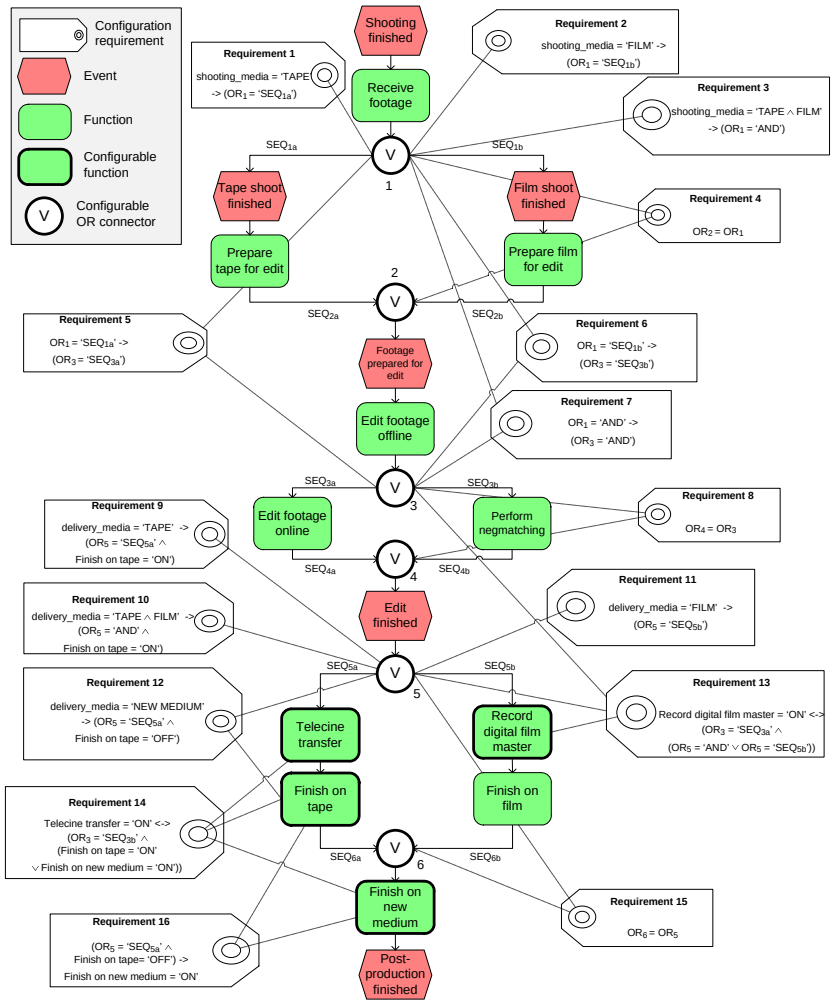

Fig. 1. C-EPC model for the screen postproduction process

base model) at specific points (i.e., *variation points*) through *change operations*. Following this approach, we find proposals such as Provop [1] or Rule representation and processing [3]. In the following we use Provop as representative for the structural approach, since it can be considered the most widely used proposal for this approach. Fig. 2 illustrates the process family representing the screen postproduction process using Provop. Provop allows creating process variants by adjusting the *base model* (cf. top part of Fig. 2) by the application of a set of high-level change operations between a couple of *adjustment points*. Furthermore, Provop allows for more complex configuration adjustments by grouping multiple *change operations* into so-called *change options* (e.g., option 1 combines 2 delete and 2 insert operations). *Change options* are associated with a

context rule, which is used at configuration time to decide, whether a certain option is applicable for the given context (e.g., regarding option 1 the context rule states that this option should only be applied if *shooting media* is *tape*). In addition, Provop allows for an explicit representation of different dimensions of the context (i.e., *context variables* and allowed *values*) by means of the *context model*. Finally, to prevent the derivation of semantically invalid variants, Provop provides the *constraint model* which allows defining *inclusion*, *exclusion*, *order of application*, *hierarchy*, and *cardinality* relationships between *change options* (e.g., the application of *option 1* excludes the application of *option 2*).

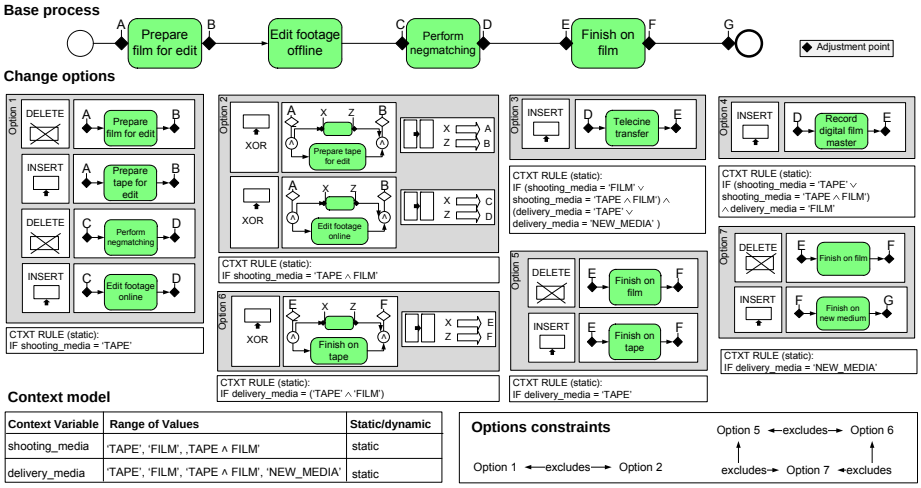

Fig. 2. Provop model for the screen postproduction process

4 Concepts from Cognitive Psychology

In order to discuss differences between C-EPC and Provop, we will make use the concepts of *external memory*, *abstraction*, and *split-attention effect* from cognitive psychology and transfer them to the domain of BP variability.

Basically, three different problem-solving "programs" or "processes" are known from cognitive psychology: search, recognition, and inference [7]. Search and recognition allow identifying information of rather low complexity, i.e., locating an object or recognizing patterns. Most models, however, go well beyond complexity that can be handled by search and recognition. For instance, a Boolean expression certainly cannot be interpreted just by looking at it and without deliberate thought. Here, the human brain as *"truly generic problem solver"* [8] comes into play. Thereby, cognitive psychology differentiates between working memory that contains the information is currently processed, as well as long-term memory in which information can be stored for a long period of time [9]. Most severe, and thus of high interest and relevance, are the limitations of the working memory. As reported in [10], working memory cannot hold more than 7±2 items at the same time. In this context, the concept of mental effort, i.e., the amount of working memory used, is of interest, as it can be used

to assess understanding. As discussed in [11], higher mental effort is in general associated with lower understanding, or more generally, errors are more likely to occur when the working memory's limits are exceeded [12]. Instead of only measuring accuracy or time for assessing the understandability of a notation, measuring mental effort allows for a more fine-grained analysis. In particular, when differences with respect to understandability are small, accuracy may not change, even though the mental effort changes significantly (cf. [13]). Subsequently, we will discuss three factors that influence the required mental effort: *external memory*, *abstraction*, and the *split-attention effect*.

External Memory. First, we would like to introduce a mechanism known for reducing mental effort, i.e., the amount of working memory slots in use. An external memory is referred to any information storage outside the human cognitive system, e.g., pencil and paper or a blackboard [12, 8, 14, 7]. Information taken from the working memory and stored in an external memory is then referred to as cognitive trace. In the context of a diagram, a cognitive trace would be, for instance, to mark, update, and highlight information [14]. Likewise, in the context of process variants, the model itself may serve as external memory. For example, when deriving a process variant in C-EPC for a particular context, the model reader may cross out model elements that have been removed (not requiring her anymore to store them in the working memory). Rather, this information is transferred to the C-EPC model, freeing up working memory capacity.

Abstraction. Basically, the idea of abstraction is to hide information by aggregation. As irrelevant information can be hidden from the reader, it becomes easier to focus on relevant information, i.e., abstraction supports the human mind's attention management [7], leading to decreased mental effort. Unlike in C-EPC, where the process family is represented in a single model, Provop separates the base model from change options and change options are abstracted via variation points. This, in turn, simplifies both the base model and the change options, presumably making both easier to interpret, as attention is not distracted by an abundance of modeling elements.

Split-Attention Effect. Even though abstraction has been attributed to reduce mental effort [11, 15, 16], it typically co-occurs with the split-attention effect, which is known to increase mental effort [17]. In general, the split-attention effect occurs whenever information from different sources needs to be integrated. As the human mind can only focus on a single aspect at the same time [18], attention needs to be constantly switched between the information sources, leading to increased mental effort. In addition, the task of integrating information is known to further increase mental effort [17]. To illustrate the split-attention effect, consider the change options in the Provop approach. Therein, the model reader has to switch attention between the base model and the change options. When extracting a process variant for a specific context, the model reader has to integrate information from change options, i.e., which model elements to change, with the base model, further increasing mental effort.

5 Qualitative Comparison

So far we have discussed C-EPC and Provop as representatives of approaches for modeling BP variability. In the following, we will use concepts from cognitive psychology to systematically assess differences between these two proposals with respect to understandability. Understandability not only depends on the notation, but also on the type of task to be performed [4, 5]. Due to space restrictions we focus in this paper on a specific understandability task, namely the extraction of a process variant from a configurable process model given a certain context. To illustrate the process in both proposals, we assume that a model reader wants to derive the process variant that relates to the production of a low-budget project which implies the use of *tape* as medium for both shooting and delivery tasks. For a description of additional tasks we refer the reader to http://www.pros.upv.es/technicalreports/PROS-TR-2012-03.pdf

In our qualitative comparison, we assume a setting where the model reader has the models available in paper-based form. We are aware that, even though our discussion focuses on cognitive aspects only, tool support is indispensable for working with configurable models. However, to be able to develop effective tool support, it is essential to know what makes configurable process models hard to understand. Without a profound discussion, as provided here, tool development is rather driven by speculation than by systematic consideration.

In the following, we provide a discussion, first for C-EPC and afterwards for Provop, structured along the following points: First, we will describe the steps a model reader has to perform in order to perform the understandability task. This descriptions have been derived in an iterative manner by observing a set of model readers conducting the task. Second, we will perform an analysis to determine the cognitive complexity of the task.

5.1 Extracting a Process Variant Using C-EPC

To obtain the process variant that relates to the low-budget project in C-EPC (i.e., when shooting and delivery medium is tape) the model reader starts with the configuration of *configurable connector 1*. For this purpose, the model reader evaluates all requirements attached to it, i.e., reqs. 1-7. According to the given context (i.e., shooting media is *tape*), *configurable connector 1* is configured as SEQ1_a as stated in req. 1. Next, the configuration of *configurable connector 2* has to be performed. In this case, req. 4 determines that the same configuration performed to *configurable connector 1* should be applied to *configurable connector 2*, i.e., SEQ2_a is chosen. Then, the configuration of *configurable connector 3* has to be done. For such purpose, reqs. 5–8 and 13 are evaluated. After evaluating req. 5 the model reader discovers that SEQ3_a should be chosen. Unlike reqs. 1–3, reqs. 4–8 and 13–16 are entirely expressed in terms of the structure of the model, not providing any information regarding the process variant being configured. This means that the model reader, in order to understand why this configuration is performed, has to either remember the decisions previously taken or go back in the model and revisit the requirements that determined the

configuration of related nodes. Similarly to the configuration performed for *configurable connector 2*, req. 8 determines that *configurable connector 4* should be configured equally to connector 3, i.e., as SEQ4$_a$. Regarding *configurable connector 5*, six requirements are attached to it, i.e., reqs. 9-13, and 15. In this case, the model reader discovers by evaluating req. 9 that it should be configured as SEQ5$_a$ and that function *finish on tape* should be switched ON. The fact that these requirements include context variables in it help the model reader to better understand which configuration should be taken. Next, the model reader has to decide about the configuration of function *telecine transfer*. In this case req. 14 states that function *telecine transfer* should be configured as OFF (since connector 3 was configured as SEQ3$_a$). Then, *configurable connector 6* is configured as SEQ6$_a$ according to req. 15. Finally, function *finish on new medium* is switched OFF since function *finish on tape* has been switched ON (req. 16).

Considering the **cognitive complexity** the use of C-EPC entails we can identify three basic operations: locating elements, evaluating Boolean expressions, and adapting the model accordingly. As argued in [19], the more distinct properties a visual element has, e.g., shape and color, the easier it is to identify. In C-EPC, requirements are represented by white tags, configurable connectors are represented by white circles with a thick border, whereas configurable functions are represented by green rounded rectangles with a thick border. Hence, the reader can draw on two different properties (i.e., color and shape) for identifying distinct modeling constructs presumably requiring a low mental effort. For identifying whether there are requirements attached to a configurable node, the model reader can rely on pattern recognition [7] to efficiently perform this operation (requirements are connected via dotted lines). The first real challenge occurs when the model reader has to evaluate associated Boolean expressions. As they can be arbitrarily complex and have to be interpreted in the model reader's mind, presumably a high mental effort can be expected. In C-EPC, some requirements are expressed in terms of the structure of the alternatives and not by the semantics of the process variants being described (e.g., reqs. 4-8, 13–16). In addition, the configuration of the configurable node being evaluated can depend on the configuration of previous and/or succeeding related configurable nodes. This requires a bigger cognitive effort by the model reader, since the model reader has to remember decisions taken for already configured nodes and might have to anticipate the configuration of succeeding nodes. For example, when evaluating req. 14 for configuring function *telecine transfer*, the model reader has to go back to the related configurable nodes, i.e., to configurable connector 3, and consider the configuration of functions *finish on tape* and *finish on new medium* to understand the semantics of the configuration. Having evaluated the requirements respective model adaptations have to be performed (e.g., removing model elements). Eventhough these operations are rather simple the model reader has to keep track of all changes made during the configuration. Due to the limited nature of working memory (7±2 slots), it seems essential that the model reader can offload parts of the working memory to an external memory, e.g., by annotating a print-out of the model.

5.2 Extracting a Process Variant Using Provop

To extract a process variant in Provop the model reader first examines all change options including their associated context rules to determine which options are applicable for the given context. Based on this, the model reader then selects *options 1* and *5*, since only these two satisfy the given context and applies them to the based model. For this the model reader checks the constraints between the selected options and concludes that both *option 1* and *option 5* can be applied. The application of *option 1* involves deleting activity *prepare film for edit* between variation points A and B, and inserting activity *prepare tape for edit* instead. In addition, activity *perform negmatching* is replaced by activity *edit footage online*. Moreover, the application of *option 5* implies the replacement of activity *finish on film* by *finish on tape* between adjustment points E and F.

Considering the **cognitive complexity** of Provop we can identify two main operations. First, selecting appropriate change options and second, applying these change options to the model. For the identification of relevant change options the model reader inspects all change options and evaluates whether they are applicable for the current context, i.e., the model reader evaluates the Boolean expression associated with the change option. Similar to C-EPC, it can be expected that the interpretation of such Boolean expressions presumably imposes a high mental effort. However, unlike C-EPC, in Provop Boolean expressions are always expressed in terms of context variables. These variables provide semantics to the change options, helping the model reader to understand the intent of the options. After having identified relevant change options, the model reader has to apply them to the base model. Before applying a change option, it has to be checked whether the change option is conflicting with previously applied change options. This task, however, can easily be accomplished using Provop's *option constraints*, i.e., a set of relationships (e.g., inclusion, exclusion) between *change options* targeted to ensure their proper use based on the semantics of the domain. As these option constraints are visually depicted, the model reader's recognition capabilities will help to efficiently identify conflicting options, hence presumably imposing a low mental effort. Regarding the actual manipulation of the model, the effort for integrating the change options into the base model is determined by the *change distance* [20] between the base model and the variant to be derived. In other words, the more modeling elements are added to / removed from the base model, the more complex the integration task will be. In addition, the type of change operations contained in the change options influences complexity. For example, when deleting an activity, respective parts can be removed from the model by hiding them with a finger on the print-out of the model, or by crossing them out using a pen. In this way, the model reader does no longer have to keep this information in his working memory. Rather, the model serves as external memory, freeing up mental resources. When inserting activities, in turn, this possibility is not available (except this is done through a supporting modeling tool) and has to be done in the reader's mind. Therefore, complexity grows with the number of changes applied to the base model. Therefore, an optimized design of the base model that requires minimum changes to

derive different variants presumably requires less effort by the model reader. Altogether it can be said that most mental effort will presumably be required for evaluating Boolean expressions as well as conducting model changes.

5.3 Discussion

After studying both proposals three major differences can be observed. First, in a C-EPC model, modeling elements are mainly removed from the configurable model (with exception of functions when these are configured as optional, which involve the inclusion of some branching condition in the model). By contrast, in Provop model elements can either be added, deleted, or moved during the configuration process. As argued above, the cognitive complexity depends on the type of change operations to be performed (i.e., deletion operations presumably involve less cognitive effort than insertions or movements).

Second, requirements and configurable nodes are integrated in C-EPC, whereas change options and the base model are separated in Provop. Similar to [15, 16], we argue that for small models, C-EPC presumably is easier to understand, as all the information is integrated and hence in contrast to Provop no split-attention effect can be expected. However, when model size increases, models may quickly become too complex resulting in an overload for the model reader, especially when there are many relationships between alternative modeling elements. Here, it can be assumed that the abstraction mechanisms provided in Provop (i.e., represented by change operations defined separately from the base model) contributes to retain understandability even for large models.

Third, even though Boolean expressions need to be evaluated in both approaches, the way they are used by the two proposals differs. In C-EPC, one of the biggest challenges faced by the model reader relates to the fact that alternatives are usually expressed at the structural level, neglecting the semantics associated to the different alternatives. This fact involves that, in some cases, the model reader has to evaluate the Boolean expression at hand, but additionally has to keep track of previously made decisions. In contrast, in Provop, Boolean expressions are always expressed in terms of context variables, which contribute to better understand the semantics of the associated change operations. In addition, the concept of options (i.e., grouping of related change operations) as provided by Provop and the explicit specification of constraints between them in the constraint model presumably reduces the mental effort required by the reader for understanding them. Hence, from this point of view, Provop models presumably impose a lower mental effort on average.

6 Related Work

Several proposals have been developed to deal with business process variability (e.g., [1, 2, 3]). These works take a design-oriented perspective and provide technical solutions for managing variability; the understandability of the artifacts

created using such approaches is not in the focus. Recently qualitative evaluations of the C-EPC proposal in form of case studies have been conducted [21, 22]. However, to the best of our knowledge no work has addressed understandability of process model families explicitly. Closely related is, however, existing empirical research on the understandability of process models. Most approaches thereby employ the concept of metrics computed on structural aspects of the process model to assess understandability, e.g., [23, 24, 25]. While metrics seem to be a promising approach to assess model complexity and understandability, in [4, 5] it is shown that understandability of a process model significantly depends on the type of question asked. Consequently, a metric will only be able to roughly estimate understandability. In [26, 15, 16], concepts from cognitive psychology are used as a tool to discuss understandability of process models for specific comprehension tasks. In this paper we build upon this work, and extend it to discuss understandability of configurable process models.

7 Summary and Outlook

The main goal of this paper is to compare the structural and behavioral approaches for modeling process model families in terms of understandability. Instead of looking at understandability from a broad perspective, the discussion is centered around the extraction task of a process variant from the modeling artifacts produced by C-EPC and Provop. The different approaches are discussed in terms of different concepts from cognitive psychology based on which the mental effort required to understand the modeling artifacts produced by the two approaches can be estimated. In turn, this allows us to estimate the understandability of the two approaches for specific comprehension task. The task addressed in this paper constitutes just a first attempt regarding the investigation of understandability of these two approaches. Formal metrics and experiments involving a large number of subjects and different configurable process models are planned as future work to empirically test the discussion. Based on the comprehension tasks presented in this paper, we will conduct a series of experiments to empirically assess the understandability of both approaches and to investigate the factors that impact understandability of process model families improving the modeling of such families and facilitating their reuse.

References

1. Hallerbach, A., Bauer, T., Reichert, M.: Capturing variability in business process models: the Provop approach, J. Soft. Maintenance. 22(6–7):519–546 (2010)
2. Rosemann, M., van der Aalst, W.M.P.: A configurable reference modeling language. Inf. Systems 32(1):1–23 (2007)
3. Kumar, A., Wen, Y.: Design and management of exible process variants using templates and rules. Int. J. Comput. Ind. 63(2), pp. 112–130 (2012).
4. Figl, K., Laue, R.: Cognitive Complexity in Business Process Modeling. In Proc. CAiSE'11, 452–466.

5. Melcher, J., Detlef, S.: Towards Validating Prediction Systems for Process Understandability: Measuring Process Understandability. In Proc. SYNASC'08, 564–571.
6. La Rosa, M., Lux, J., Seidel, S., Dumas, M., ter Hofstede, A.H.M.: Questionnaire-driven Configuration of Reference Process Models. In Proc. CAiSE'07, 424-438.
7. Larkin, J.H., Simon, H.A.: Why a Diagram is (Sometimes) Worth Ten Thousand Words. Cognitive Science, 11(1):65–100 (1987).
8. Tracz, W. J.: Computer programming and the human thought process. Software: Practice and Experience, 9(2):127–137 (1979).
9. Paas, F., Tuovinen, J.E., Tabbers, H., Van Gerven, P.W.M.: Cognitive Load Measurement as a Means to Advance Cognitive Load Theory. Educational Psychologist, 38(1):63–71 (2003).
10. Miller, G.: The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. The Psychological Review, 63(2):81–97 (1956).
11. Moody, D. L.: Cognitive Load Effects on End User Understanding of Conceptual Models: An Experimental Analysis. In Proc. ADBIS'04, 129–143.
12. Sweller, J.: Cognitive load during problem solving: Effects on learning. Cognitive Science, 12(2):257–285 (1988).
13. Zugal, S., Pinggera, J., Weber, B.: The Impact of Testcases on the Maintainability of Declarative Process Models. In Proc. BPMDS'11, 163–177.
14. Scaife, M., Rogers, Y.: External cognition: how do graphical representations work? Int. J. Human-Computer Studies, 45(2):185–213 (1996).
15. Zugal, S., Pinggera, J., Mendling, J., Reijers, H.A., Weber, B.: Assessing the Impact of Hierarchy on Model Understandability-A Cognitive Perspective. In Proc. EESSMod'11, 123–133.
16. Zugal, S., Soffer, P., Pinggera, J., Weber, B.: Expressiveness and Understandability Considerations of Hierarchy in Declarative Business Process Models. In Proc. BPMDS'12, 167–181.
17. Sweller, J., Chandler, P.: Why Some Material Is Difficult to Learn. Cognition and Instruction, 12(3):185–233 (1994).
18. Feldmann Barrett, L., Tugade, M. M., Engle, R. W.: Individual Differences in Working Memory Capacity and Dual-Process Theories of the Mind, Psychol Bull. 130(4):553-573 (2004).
19. Moody, D.L.: The "Physics" of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. IEEE Trans. Soft. Eng. 35(6):756–779 (2009).
20. Li, C., Reichert, M., Wombacher, A.: On Measuring Process Model Similarity Based on High-Level Change Operations. In Proc. ER'08, 248–264.
21. Lönn, C.M., Uppström, E., Wohed, P., Juell-Skielse, G.: Configurable Process Models for the Swedish Public Sector. In Proc. CAiSE'12, 190–205.
22. Gottschalk, F., Wagemakers, T., Jansen-Vullers, M., van der Aalst, W., La Rosa, M., van Eck, P., Gordijn, J., Wieringa, R.: Configurable Process Models: Experiences from a Municipality Case Study. In Proc. CAiSE'09, 486–500.
23. Mendling, J., Reijers, H.S., Cardoso. J.:What Makes Process Models Understandable? In Proc. BPM'07, 48–63.
24. Vanderfeesten, I., Reijers, H.A., van der Aalst, W.M.P.: Evaluating workflow process designs using cohesion and coupling metrics. Int. J. Comput. Ind. 59(5):420–437 (2008).
25. Reijers, H.A., Mendling, J.: A Study into the Factors that Influence the Understandability of Business Process Models. SMCA 41(3):449–462 (2011).
26. Zugal, S., Pinggera, J., Weber, B.: Assessing Process Models with Cognitive Psychology. In Proc. EMISA'11, 177–182.