Understanding Business Process Models: The Costs and Benefits of Structuredness

Marlon Dumas[1], Marcello La Rosa[2,3], Jan Mendling[4], Raul Mäesalu[1],
Hajo A. Reijers[5], and Nataliia Semenenko[1]

[1] University of Tartu, Estonia
{marlon.dumas, maekas, nataliia}@ut.ee
[2] Queensland University of Technology, Australia
m.larosa@qut.edu.au
[3] NICTA Queensland Research Lab, Australia
[4] Vienna University of Business and Economics, Austria
contact@mendling.com
[5] Eindhoven University of Technology, The Netherlands
h.a.reijers@tue.nl

Abstract. Previous research has put forward a number of properties of business process models that have an impact on their understandability. Two such properties are compactness and (block-)structuredness. What has not been sufficiently appreciated at this point is that these desirable properties may be at odds with one another. This paper presents the results of a two-pronged study aimed at exploring the trade-off between compactness and structuredness of process models. The first prong of the study is a comparative analysis of the complexity of a set of unstructured process models from industrial practice and of their corresponding structured versions. The second prong is an experiment wherein a cohort of students was exposed to semantically equivalent unstructured and structured process models. The key finding is that structuredness is not an absolute desideratum vis-a-vis for process model understandability. Instead, subtle trade-offs between structuredness and other model properties are at play.

1 Introduction

In many contexts where information systems are developed and used, conceptual models are employed to inform stakeholders on the business processes supported by such systems. Sometimes, hundreds or thousands of process models are created and maintained in order to document large information systems. Given that such model collections are consulted, validated and updated by a wide range of stakeholders with various levels of expertise, ensuring the understandability of process models is a key concern in such settings.

In this respect, a central guideline for business process modeling is to use structured building blocks as much as possible [18]. This insight has triggered a stream of research on transforming unstructured process models into structured ones. The approach in [24] provides a formal foundation for determining when and how such a transformation is possible. However, it turns out that in some cases structuredness can only be achieved at the expense of compactness. Specifically, there are situations where nodes must be duplicated in order to transform an unstructured model into a structured one. Importantly, this node duplication is not a limitation of the technique proposed in [24], but an unavoidable constraint that has been studied in the field of compiler theory [22].

While both structuredness and compactness are recognized as basic design principles for process models [18], to the best of our knowledge the tradeoff between these two principles has not been studied so far. In this context, this paper examines whether or not the benefits of structuring an unstructured process model outweigh the costs of duplication from the perspective of understandability. Specifically, the following research question is addressed:

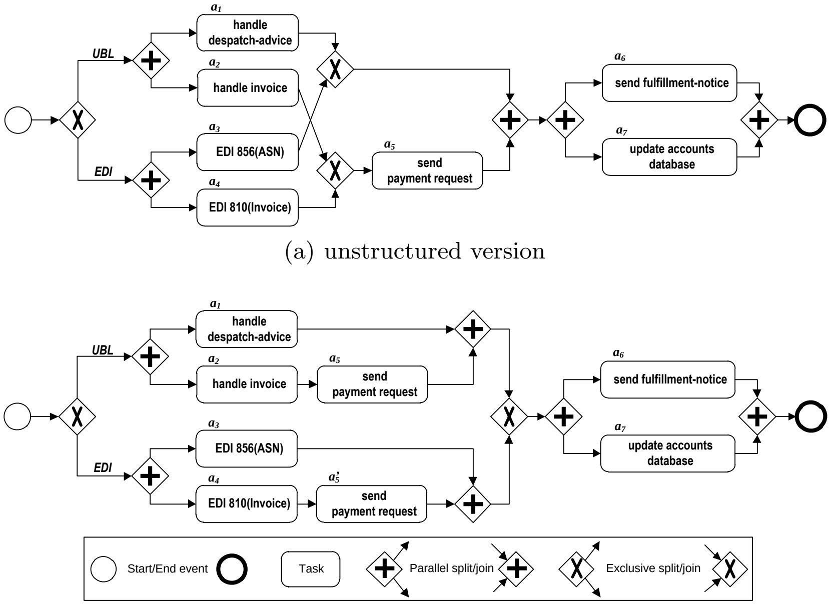
> If an unstructured process model U is transformed into a structured one S, is S more understandable than U?

To tackle this question, we adopt a two-pronged approach. First, by using a collection of industrial process models, we compare the relative complexity of unstructured process models and of their corresponding structured versions. This study is based on a collection of complexity metrics that have been shown to be negatively correlated with understandability [18], that is, higher complexity is associated with lower understandability. The results of this comparative complexity study are inconclusive with respect to the hypothesis. In this sense, they confirm the existence of a tradeoff between structured and unstructured process models. Second, we conducted an experiment with structured and unstructured models as a treatment, and measured the understanding performance of a cohort of process modeling students. The results show that in some cases it is preferable to leave a process model unstructured from the perspective of understandability.

The paper is structured as follows. Section 2 introduces background notions, including that of structuredness and some complexity metrics, and motivates our work against the state of the art. Section 3 presents the comparative complexity study, which is followed by the design of the conducted experiment (Section 4) and its results (Section 5). Section 6 concludes the paper.

2 Background

This section provides the background knowledge for understanding the rest of this paper. It introduces the notions of structuredness, complexity metrics and understandability of process models. It also summarizes previous research on structuring process models and empirical studies on the importance of structuredness for model understanding and correctness.



(a) unstructured version

(b) structured version

Fig. 1. Unstructured process model and its equivalent structured version

2.1 The Notion of Structuredness

Although process models captured in graph-oriented languages such as BPMN or EPCs may have almost any topology, it is often preferable that they adhere to some structural rules. In this respect, a well-known property of process models is that of *block-structuredness*, meaning that for every node with multiple outgoing arcs (a *split*) there is a corresponding node with multiple incoming arcs (a *join*) such that the subgraph between the split and the join forms a single-entry-single-exit (SESE) region. For example, the BPMN process model shown in Fig.1(a) is unstructured because the parallel split gateways do not satisfy the above condition. Fig.1(b) shows a semantically equivalent yet structured model.

Previous research on structured process modeling has demonstrated that not all unstructured process models can be transformed into equivalent structured models [11]. Such models are hereby called *inherently unstructured*. There are two root causes for process models being inherently unstructured. The first one arises from the structure shown in Fig.2; it has been shown to be inherently unstructured in [11]. The second cause is a cycle that has more than one exit point, in which case it can only be structured by introducing boolean variables to capture part of the control-flow [22] – arguably an undesirable solution in the context of process modeling.

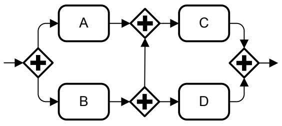
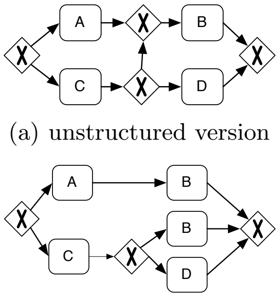


Fig. 2. Inherently unstructured BPMN model

It is also known that transforming an unstructured process model into a structured one may lead to the introduction of duplicate nodes [22]. This duplication arises when there is an XOR-join that is not the boundary of a SESE region of the model. In some cases, to structure these models, the paths that follow the XOR-join need to be duplicated until the exit point of the smallest enclosing SESE region is reached. For example, in the structured model in Fig.1(b), the "send payment-request" task is duplicated. In cases where loops have multiple entries, the paths between the entry of the loop and the XOR-join need to be duplicated [22]. A direct effect of this duplication is that when transforming an unstructured process model into a structured one, the size of the resulting model is typically higher. In a study based on the SAP R/3 reference model collection, converting unstructured process models into structured ones led to an average size increase of 22% and a maximum size increase of 63% [24]. This size increase is attributable to an increase in the number of task/event nodes. The number of gateways increases marginally and, in some cases, even decreases due to consolidation. This is illustrated in Figure 3.



(a) unstructured version

(b) structured version

Fig. 3. Unstructured model with less gateways than structured version.

2.2 Structuring process models

The problem of structuring process models has been extensively studied since the early 2000s. It has its roots in compiler theory [22] and code refactoring [33]. Kiepuszewski et al. [11] made a first attempt at classifying unstructured process models that can be transformed to structured equivalents. They showed that unstructured process models cannot always be replaced with structured models that are behavior-equivalent. Later, Liu and Kumar [15] presented an alternative taxonomy of unstructured process models and sketched a method to transform some types of unstructured models into structured versions. Different transformation strategies were also illustrated in [17] and an alternative classification of (unstructured) process models was proposed in [9]. A method specifically tailored to untangling unstructured cyclic models and transforming them into structured BPEL models is presented in [10, 12]. Finally, Polyvyanyy et al. [24] provide a complete characterization of unstructured acyclic process models that are inherently unstructured, based on the RPST decomposition developed by Vanhatalo et al. [31].

Previous research on structuring process models has been consolidated in a tool called BPStruct[6]. BPStruct tries to structure a model to the maximum possible extent. The resulting model is then said to be *maximally-structured*.

[6] `http://code.google.com/p/bpstruct/`.

2.3 Complexity Metrics and Understandability

In addition to formal work on structuring process models, there is a growing body of research on how the understandability of a process model can be evaluated from an empirical perspective. Different factors of model understanding have been studied based on experiments and survey research, including modeling notation [1], model complexity [26], modeling expertise [23, 26], secondary notation [7, 25], and labeling conventions [20]. Throughout this paper, we will focus on model complexity and take an observer's expertise into account.

The importance of complexity for model understanding has been emphasized by various authors. Inspired by works in software measurement [5, 16], early work focused on the definition of metrics for process models [14, 21]. Further metrics were proposed since then in [2–4, 29, 30]. These measures can be roughly categorized into six groups [19]: measures of size, density, modularity, connector interplay, cyclicity, and concurrency. *Size measures* count different types of elements that appear in a process model, such as the number of nodes or number of arcs. *Density measures* capture the relationship between nodes and arcs. For instance, density is the ratio of arcs divided by the maximum possible number of arcs for a given set of nodes. Another option is to use the average connector degree (ACD), which yields the average number of arcs a node is associated with. *Modularity* covers aspects such as structuredness. The degree of structuredness defines the ratio of nodes in structured components to all components. *Connector interplay* quantifies a potential mismatch of split and join nodes. For instance, the cross connectivity (CC) metric defines how many routing elements can be expected on a path between two nodes [29]. *Cyclicity* refers to the extent to which models include cycles. Finally,*concurrency measures* describe how much concurrency can be encountered in a process model. However, while in software engineering duplication is generally considered harmful for maintenance and comprehension [27], the issue of duplication is completely missing from the complexity metrics of process models up until now.

Empirical research in the area of process model understanding has focused on controlled experiments in order to find out how strong understanding is affected by a varying complexity. Several studies demonstrate the effect of size (see [26]). As a result of a study on predicting error probability for a set of 2,000 process models from practice [19], structuredness appears to be the best determinant to distinguish low-error-probability models from ones with high error probability. Another study confirms the significance of structuredness, albeit that different definitions are used [13]. These and other experiments are summarized in the seven process modeling guidelines [18]. Specifically, one of these guidelines is to model processes as structured as possible, which ranked as the guideline with the highest relative potential for improving process model understandability. Recently, this work has been complemented with methodological guidance for determining statistically significant threshold values for the complexity metrics [28].

Importantly, all of the above studies on the link between complexity metrics and understandability take as input samples consisting of distinct sets of structured and unstructured process models, by which we mean that when the sample includes an unstructured process model, it does not include an equivalent structured version of this model. Thus, the above studies do not directly address the question formulated in Section 1. Also, these studies altogether do not permit one to draw any conclusions on the trade-off between structuredness and duplication. A process model with small size and high structuredness appears to be preferable from a perspective of model comprehension. Yet, it is not clear whether a structured model should be preferred in case it implies duplicating a specific amount of nodes and gateways. While an increase in size is considered as harmful, duplication might even be worse. The negative effect of duplication when structuring a process model has been briefly considered in [8]. Here the authors point out that when structuring leads to unwanted duplications, it might be preferable to leave a process model unstructured. However, they do not validate these intuitions empirically.

3 Complexity Comparison

In this section, we present a comparative complexity study aimed at testing the hypothesis formulated in Section 1. For this study, we relied on the IBM Business Integration Technology (BIT) library: a publicly-available collection of process models gathered from IBM's consultancy practice [6]. The BIT library is divided into three collections, namely A, B3 and C.[7] After removing semantically incorrect models, these collections contain 269, 247 and 17 models respectively.

In order to test the hypothesis, we are interested in comparing the complexity of unstructured models against the complexity of the corresponding structured models (or the maximally-structured models in case of inherently-unstructured models). Accordingly, we extracted all 59 correct and unstructured models from the BIT library (41 in collection A, 15 in B3 and 3 in C). Each of these models was structured using BPStruct. The output was manually checked to ensure that no errors had been introduced by the tool. 11 models with cycles were found to be inherently unstructured (cycles with multiple exit points), in which case we took the maximally-structured model produced by BPStruct as being the "structured version". None of the correct, acyclic models in the library was found to be inherently unstructured.

For each of the 59 resulting pairs of models (one structured, one unstructured), we computed seven metrics: number of arcs, number of gateways, number of tasks, size (number of nodes), Average Connector Degree (ACD), Cross-Connectivity (CC), and Density as introduced in Section 2. The average values of these metrics for the unstructured models and for the corresponding structured models are shown in Table 1.

As expected, the average size of the structured models is higher because of the node duplication phenomenon discussed in Section 2. On average, the increase in size is in the order of 50%, which is consistent with previous findings

[7] There are also two other collections (B1 and B2) containing earlier versions of the models in B3, but for the purposes of this study these collections are redundant.

Metric	Avg. unstructured	Avg. structured	Increase/decrease (%)
# arcs	30.02	44.76	49.12%
# gateways	8.41	12.31	46.37%
# tasks	16.81	25.76	53.23%
Size	25.22	38.07	46.53%
ACD	3.41590	3.29957	-3.04%
CC	0.05118	0.04455	-11.44%
Density	0.16096	0.12376	-22.5%

Table 1. Complexity of unstructured models vs. corresponding structured models

reported in [24]. On the other hand, we observe a notable decrease in CC and density when the models become structured. The decrease in density reflects the intuition that structured process models are less "cluttered" while the decrease in CC supports the intuition that the control-flow relations between pairs of tasks in structured models are cognitively simpler. We also note a marginal decrease in ACD because of structuring, attributable to gateway reshuffling.

The differential in size entails that the structured process models are more difficult and, according to the empirical evidence in [18], we expect these models to be less understandable than the unstructured ones. On the other hand, lower ACD, CC and density suggest that structured models are less complex and, based on the same empirical study, we expect that the structured process models are more understandable than the unstructured ones. Therefore, the comparative complexity study shows that there is a tradeoff between different complexity indices when the unstructured models in the dataset are transformed into structured ones. In other words, the comparative complexity analysis neither supports nor refutes the hypothesis that structuring an unstructured process model leads to a higher understandability.

4 Controlled Experiment

Considering the ambivalent outcomes produced by the comparative complexity study, we decided to approach the research question with a questionnaire-based, controlled experiment. In this section, we describe its design.

Subjects: The experiment was run at Queensland University of Technology in early October 2011. The population consisted of 110 students from two business process modeling courses. The students were in their tenth week of training, such that they had acquired the requisite background to read BPMN process models. Participation in the study was voluntary.

Objects: We took 8 models from the IBM dataset (cf. Section 3) as the objects of our experimental investigation. The models were selected on the basis of their size and whether they had cycles or not. Based on these two factors, four categories of models were created:

1. Acyclic unstructured models whose equivalent structured models are at least 25% larger than the unstructured versions.
2. Cyclic unstructured models whose equivalent structured models are at least 25% larger than the unstructured versions.
3. Acyclic unstructured models whose equivalent structured models are less than 10% larger than the unstructured versions.
4. Cyclic unstructured models whose equivalent structured models are less than 10% larger than the unstructured versions.

For each of these four categories, two models were selected. When selecting models in a given category, we sought models with heterogeneous values for CC, ACD and density. The differential in complexity metrics between the unstructured and structured versions is discussed in Section 5.

Factor and Factor Levels: The main factor manipulated in our experiment is the *structuredness* of each process model, as presented to the respondents. Given the binomial character of this factor, it has *structured* and *unstructured* as its levels for each of the used models.

Response Variables: As response variable we used a composite measure to reflect a respondent's understanding of a process model – in a structured or unstructured form – as determined by the sum of correct answers to a set of six questions asked for each model.

Hypothesis Formulation: The goal of the experiment was to investigate whether the level of structuredness of process model influences a respondent's understanding of a process model. While we would expect a beneficial effect of structuring a process model, we appreciate the potential drawback of using duplicate nodes. Accordingly, we postulate the following hypotheses:

- **Hypothesis**: Structured models yield a greater amount of correct answers than unstructured models.
- **Alternative Hypothesis:** Models without duplicates yield a greater amount of correct answers than structured models.

Instrumentation: The participants conducted the experiment by observing the selected process models on paper, as part of a work-book. On a single page, one process model along with a set of six questions to test the respondent's comprehension of that model were presented. All models were presented in the BPMN notation. The questions used were designed in a similar fashion to the questions of earlier experiments into process model understanding [26]. An example question from the questionnaire is: "If J is executed for a case, can F be executed for the same case?" Additionally, subjects were asked to subjectively rate the complexity of each model on a 5-point Likert scale. Besides these understanding questions, the questionnaire recorded personal characteristics of the participants, including their theoretical knowledge on process modeling as well as the amount of process modeling training they had received in the past.

Experimental Design: The experimental setup was based on literature providing guidelines for designing experiments [32]. Following these guidelines a *randomized balanced single factor* experiment was conducted with *a single measurement*. The experiment is called *randomized* because subjects are randomly assigned to two different groups. We denote the experiment as *balanced* as each factor level is used by each subject, i.e., each student is confronted to both structured and unstructured process models. As only a single factor is manipulated (i.e., the level of structuredness of each process model), the design is called *single factor*. Due to the balanced nature of the experiment, each subject generates data for both factor levels. To avoid learning effects, each subject sees each process model only once: either in its structured or unstructured form. This is achieved by having two versions of the questionnaire: one per group. In the version of the questionnaire for the first group, all odd-numbered models were structured while in the version for the second group all even-numbered models were structured. Thus, each subject evaluates four structured and four unstructured models and none of the models a subject evaluates is equivalent to any other model evaluated by the same subject.

5 Results

In this section, we present the results of the experiment described in the previous section. First, we explain the data cleansing steps we conducted. The resulting sample serves us for testing the research hypothesis.

5.1 Data Cleansing and Demographics

To avoid bias in terms of modeling expertise, we decided to filter out the responses of those respondents who were highly familiar with process modeling and who had experience with process modeling in practice. Furthermore, we filtered out some potentially meaningless answers (e.g. training of more than 20 work days on process modeling in the last year). The resulting analysis sample includes the responses from 55 students who can all be classified as educated novices in process modeling. On average, these participants got 6.07 out of twelve theory questions right (std. deviation of 2.209) and 4.35 of the six understanding questions for each of the eight models (std. deviation 1.701). By having 55 students evaluating eight models each, we received 440 model understanding observations. 27 and 28 participants account for each version of the questionnaire.

5.2 Hypothesis Testing

By splitting the 440 observations into structured versus unstructured models, we did not find an aggregated effect. Therefore, we run a univariate ANOVA for each of the eight models by taking theory and structuredness as independent variables into account. In this way, we can analyze the hypothesis that participants might understand structured process models better than unstructured ones.

Table 2. Results of ANOVA analysis for all eight models

Model	Structure	Mean Correct	(Std.D.)	Estim. Mean	Intercept Sig.	Struct. Eff.Size	Struct. Sig.	Theory Eff.Size	Theory Sig.	R^2	better
1	struct.	4.29	(1.21)	4.213							
	unstruct.	4.26	(1.63)	4.211	.001	.002	.776	.303	.115	.375	struct.
2	struct.	4.85	(1.13)	4.823							
	unstruct.	4.64	(1.31)	4.611	.001	.001	.853	.165	.628	.257	struct.
3	struct.	3.32	(1.28)	3.264							
	unstruct.	3.85	(1.46)	4.263	.001	.122*	.032	.245	.271	.342	unstruct.
4	struct.	5.07	(1.64)	4.955							
	unstruct.	4.46	(1.35)	4.278	.001	.176*	.009	.396*	.019	.495	struct.
5	struct.	4.43	(1.89)	3.866							
	unstruct.	4.70	(1.71)	4.600	.001	.136*	.023	.420*	.011	.494	unstruct.
6	struct.	4.89	(1.79)	4.786							
	unstruct.	4.32	(2.16)	3.694	.001	.168*	.011	.355*	.045	.446	struct.
7	struct.	4.18	(2.28)	3.676							
	unstruct.	4.56	(1.78)	4.140	.001	.066	.121	.454*	.005	.518	unstruct.
8	struct.	4.30	(1.71)	3.711							
	unstruct.	3.57	(1.89)	3.273	.001	.065	.122	.395*	.020	.485	struct.

The results of the ANOVA analysis are summarized in Table 2. The table lists the mean sum of correct answers along with standard deviation and estimated marginal means for both the structured and the unstructured version of each model. From these mean values, it can be seen that the effect of structuring has a different orientation for the models. Models 3, 5, and 7 yield better understanding in their unstructured version while for the other models the structured version is better. The columns following the estimated mean shows the significance of the ANOVA model coefficients. The intercept is significant in all cases. The effect of structure is only significant for models 3 to 6. Notably, the direction for these four models is leaning towards the structured version in two cases, and also two cases towards the unstructured model. The effect size is between 0.122 and 0.176. For models 4 to 8, there is a significant effect of theory ranging between 0.355 to 0.454. The R^2 values range from 0.257 to 0.518. These statistical results demonstrate that a considerable share of variance can be explained by structuredness and theoretical knowledge. They also point to the fact that structuredness and theory alone cannot explain the direction of the structuring effect on understanding. In the following, we will investigate which further characteristics might help to explain these diverging effects.

5.3 Qualitative Analysis

In order to analyze the factors that may determine whether structured models are preferred over unstructured ones, we summarize in Table 3 the characteristics of each model, the aggregate scores for perceived complexity, and the number of correct answers broken down by model. Columns 1-2 indicate the model number and a label indicating whether the model is the structured (s) version or the unstructured (u) one. For example, the first row corresponds to model 1, structured version. Columns 3-9 provide the following properties for each model: number of nodes, number of gateways, whether or not the model contains cycles, and the increase in size, CC, ACD and Density between the structured version of the model and the unstructured one. Column 10 provides the mean score for perceived complexity. This score was obtained by asking the following question for each model: “Please rate the difficulty of the model on a 5-point scale” with 1 mapping to “very simple” and 5 mapping to “very difficult”. Finally, column 12 provides the mean number of correct answers – copied here for the sake of convenience from Table 2.

Table 3. Model characteristics versus dependent variables

Model		#	#	Cycles?	Size	CC	ACD	Density	Perceived	Mean
		nodes	gateways		Increase	Increase	Increase	Increase	complexity	correct
1	s	30	5	no	1.43	2.07	14.55	-17.44	2.22	4.29
	u	21	7						2.84	4.26
2	s	24	5	no	1.50	-7.63	8.00	-21.43	2.66	4.85
	u	16	6						2.66	4.64
3	s	21	8	yes	1.31	-21.60	0.00	-11.11	3.21	3.32
	u	16	6						3.04	3.85
4	s	33	14	yes	1.65	-41.10	-12.50	-44.75	3.52	5.07
	u	20	7						3.75	4.46
5	s	32	10	no	1.23	2.62	-9.33	-26.00	3.60	4.43
	u	26	8						3.13	4.70
6	s	27	8	no	1.13	11.37	5.00	0.00	3.04	4.89
	u	24	9						3.49	3.32
7	s	21	8	yes	1.17	-9.97	-4.55	-18.75	3.51	4.18
	u	18	7						3.18	4.56
8	s	25	7	yes	1.04	30.60	4.76	6.06	2.85	4.30
	u	24	8						3.60	3.57

From this table, we observe the following:

- An increase in perceived complexity is accompanied in all cases by a decrease in the number of correct answers and vice-versa. The only exception is Model 2: the perceived complexity is exactly the same for both versions of the model.
- Increases in the four complexity metrics shown in the table (size, CC, ACD and Density) do not explain why some unstructured models are preferred over the structured counterparts or vice-versa. For example, for Model 3 there is a decrease in CC and its unstructured version seems is preferred, while for Model 1 and Model 8 there is an increase in CC but the structured version is preferred. The same can be said for the density metrics. More striking is the fact that for the models with the highest increase in size (Models 1, 2 and 4), the structured version was preferred. This latter observation suggests that duplication does not play an important role.
- In all three cases where the unstructured version is preferred, the number of gateways is *higher* in the structured version than in the unstructured one. Similarly, 4 of the 5 instances where the structured model is preferred, the number of gateways is *higher* in the unstructured versions. The sole exception is Model 4.

With reference to the last two observations, Fig. 4 shows the unstructured and structured versions of Model 4. This model has the highest increase in size after

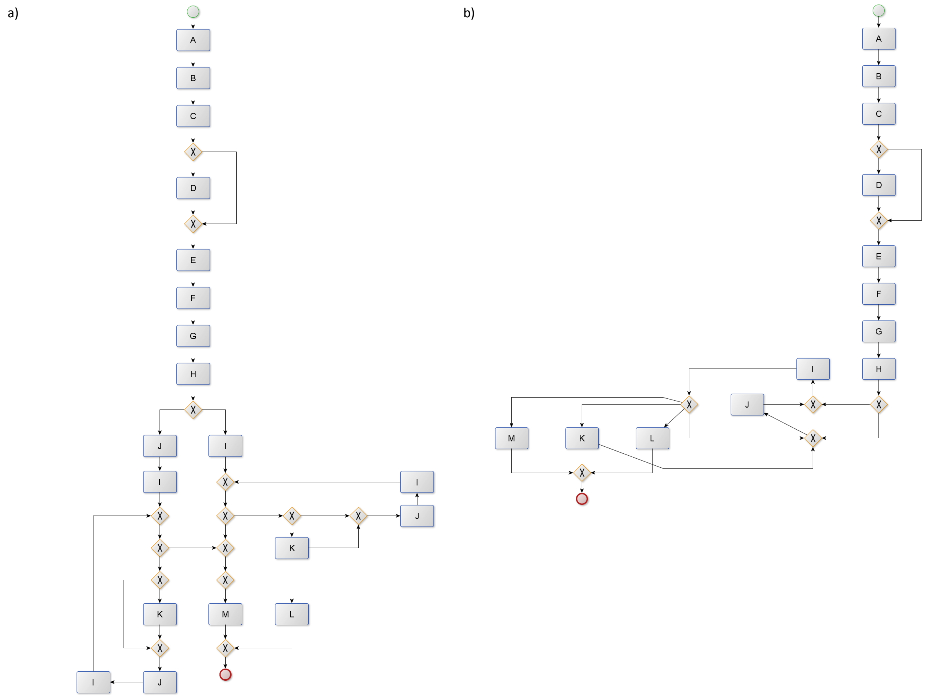


Fig. 4. Structured (a) and unstructured (b) versions of Model 4 used in the experiment.

restructuring. It also has a higher number of gateways in the structured version. Still, this version is perceived to be less complex and the number of correct answers is higher in the structured version. Interestingly, this is the only model where all three non-size-related complexity metrics are noticeably lower for the structured model vs. the unstructured one, and it is the model with the strongest decrease in CC and in density. Thus, whilst these three complexity metrics cannot predict in the general case whether subjects would prefer the structured or the unstructured version of a model, it appears that the predictions made by these metrics are accurate in cases where the differentials in complexity are clear-cut. This observation suggests that there might be some "thresholds" beyond which a decrease in CC or density when structuring an unstructured model do imply an increase in understandability. Previous research has identified thresholds beyond which complexity metrics (including density) are significant [28]. Unfortunately, this previous study does not consider the case where an unstructured process model is pitched against its corresponding structured version.

Although the number of models in the experiment is too small to draw a statistically significant conclusion, it appears that structuring leads to more easily understandable models if it does not increase the number of gateways. This is not a general rule as, indeed, instances can be found where increases in understandability can be achieved by means of structuring despite increases in the number of gateways.

5.4 Threats to validity

The controlled experiment was conducted mainly with novice process modelers. Answers given by subjects with higher level of expertise were filtered out during analysis since dealing with multiple levels of expertise would require a substantially larger group or, better still, multiple groups with different backgrounds. Thus, a possible extension of this work is to conduct similar experiments with cohorts of different expertise levels.

The number of models (8) is insufficient to make definite conclusions regarding the factors that affect whether or not the structured model is preferred over the unstructured one. This is why the discussion in Section 5.3 is qualitative in nature. A separate study would be needed to zoom into this question.

The fact that all models in the study came from the same source, i.e. the IBM BIT library, can also be seen as a limitation. While this library contains process models from industrial practice and encompasses three collections of models (from three different origins), it is possible that taking process models from other sources might impact our insights. Also, in this library, the labels in the process model were anonymized into labels of the form "s00000982" or "s00001088" prior to the public release of the library [6]. For our controlled experiment, we replaced these anonymized labels with single symbols ("A", "B", "C", ...) in order to more easily refer to them in the questionnaire. It is possible that the lack of node labels referring to a real-world application domain might have an effect on the results of the study. The latter limitation is shared with several other studies on complexity and understandability [13, 26, 28].

6 Conclusion

This study has exposed the subtlety of tradeoffs that are involved with trying to improve the understandability of process models. Specifically, we have shown that while structuring an unstructured process model may be in line with existing guidelines for ensuring process model understandability, its intention to make a model more easily readable may be thwarted by its side-effects.

The comparative complexity study based on a set of industrial models provided no conclusive insights into the relative tradeoff between structuredness and a variety of complexity measures. The controlled experiment involving a cohort of process modeling novices confirmed the equivocal nature of the research issue at hand by showing that in certain cases a unstructured version of a process model is more easily interpreted than its corresponding structured equivalent, while in other cases the opposite holds.

The contribution of this paper is that it provides a first insight into an important factor that mediates the noted effect, i.e. the number of gateways in the structured version of an unstructured process model. The results suggest that if structuring a process model leads to a smaller number of gateways, the structured model would be preferred over presenting the unstructured version of that model. While we are the first to report on this specific relation and accompanying suggestion, it is congruent with earlier work that has pointed at specific properties of gateways in a process model as a source of mental strain [26].

We believe that this work contributes to the further development and finetuning of guidelines that will help to optimize the use of business process models. Considering the enormous interest from research and praxis into such guidelines, it makes sense to push further towards a more complete understanding of the conditions under which process transformations are beneficial. In future research, we aim to analyze the observed mediating effect of gateways as identified through this work. To this end, we need to conduct experiments with a bigger set of process models. A specific challenge will be to find a research design that circumvents fatigue effects with participants. Showing eight models can still be considered acceptable; this is not so for a set of 30 or 40 models. Furthermore, we aim to investigate the application of classification techniques to analyze the tradeoffs between structuredness and variations in other complexity metrics.

Acknowledgments This work is partly funded by the Estonian Science Foundation and by ERDF via the Estonian Centre of Excellence in Computer Science (Dumas), and by the ARC Linkage project LP110100252 (La Rosa).

References

1. R. Agarwal, P. De, and AP Sinha. Comprehending object and process models: An empirical study. *IEEE T. on Software Engineering*, 25(4):541–556, 1999.
2. E. Rolón Aguilar, F. García, F. Ruiz, and M. Piattini. An exploratory experiment to validate measures for business process models. In *Proc. of RCIS*, pages 271–280, 2007.
3. G. Canfora, F. García, M. Piattini, F. Ruiz, and C.A. Visaggio. A family of experiments to validate metrics for software process models. *Journal of Systems and Software*, 77(2):113–129, 2005.
4. J. Cardoso. Evaluating workflows and web process complexity. In Layna Fischer, editor, *Workflow Handbook 2005*, pages 284–290.
5. S.R. Chidamber and C.F. Kemerer. A metrics suite for object oriented design. *IEEE T. on Software Engineering*, 20(6):476–493, 1994.
6. D. Fahland, C. Favre, B. Jobstmann, J. Koehler, N. Lohmann, H. Völzer, and K. Wolf. Instantaneous soundness checking of industrial business process models. In *Proc. of BPM*, volume 5701 of *LNCS*, pages 278–293.
7. T.R.G. Green and M. Petre. Usability Analysis of Visual Programming Environments: A 'Cognitive Dimensions' Framework. *J. Vis. Lang. Comput.*, 7(2):131–174, 1996.
8. V. Gruhn and R. Laue. Good and bad excuses for unstructured business process models. In *Proc. of EuroPLoP*, pages 279–292, 2008.
9. R. Hauser, M. Friess, J.M. Kuster, and J. Vanhatalo. An Incremental Approach to the Analysis and Transformation of Workflows Using Region Trees. *IEEE T. on Systems, Man, and Cybernetics, Part C*, 38(3):347–359, 2008.

10. R. Hauser and J. Koehler. Compiling Process Graphs into Executable Code. In *GPCE*, volume 3286 of *LNCS*, pages 317–336, 2004.
11. B. Kiepuszewski, A.H.M ter Hofstede, and C. Bussler. On Structured Workflow Modelling. In *Proc. of CAiSE*, volume 1789 of *LNCS*, pages 431–445.
12. J. Koehler and R. Hauser. Untangling Unstructured Cyclic Flows – A Solution Based on Continuations. In *OTM*, volume 3290 of *LNCS*, pages 121–138, 2004.
13. R. Laue and J. Mendling. Structuredness and its significance for correctness of process models. *Inf. Syst. E-Business Management*, 8(3):287–307, 2010.
14. G.S. Lee and J.-M. Yoon. An empirical study on the complexity metrics of petri nets. *Microelectronics and Reliability*, 32(3):323–329, 1992.
15. R. Liu and A. Kumar. An Analysis and Taxonomy of Unstructured Workflows. In *Proc. of BPM*, volume 3649 of *LNCS*, pages 268–284, 2005.
16. T.J. McCabe. A complexity measure. *IEEE T. on Software Engineering*, 2(4):308–320, 1976.
17. J. Mendling, K.B. Lassen, and U. Zdun. On the transformation of control flow between block-oriented and graph-oriented process modelling languages. *IJBPIM*, 3(2):96–108, 2008.
18. J. Mendling, H.A. Reijers, and W.M.P. van der Aalst. Seven Process Modeling Guidelines (7PMG). *Information and Software Technology*, 52(2):127–136, 2010.
19. J. Mendling. *Metrics for Process Models*, volume 6 of *LNBIP*. Springer, 2008.
20. J. Mendling, H.A. Reijers, and J. Recker. Activity labeling in process modeling: Empirical insights and recommendations. *Inf. Syst.*, 35(4):467–482, 2010.
21. M.E. Nissen. Redesigning reengineering through measurement-driven inference. *MIS Quarterly*, 22(4):509–534, 1998.
22. G. Oulsnam. Unravelling unstructured programs. *Comput. J.*, 25(3):379–387, 1982.
23. Marian Petre. Why looking isn't always seeing: Readership skills and graphical programming. *Commun. ACM*, 38(6):33–44, 1995.
24. A. Polyvyanyy, L. García-Bañuelos, and M. Dumas. Structuring Acyclic Process Models. *Information Systems*, 2012 (to appear).
25. H.A. Reijers, T. Freytag, J. Mendling, and A. Eckleder. Syntax highlighting in business process models. *Decision Support Systems*, 51(3):339–349, 2011.
26. H.A. Reijers and J. Mendling. A study into the factors that influence the understandability of business process models. *IEEE T. on Systems, Man, and Cybernetics, Part A*, 41(3):449–462, 2011.
27. M. Rieger, S. Ducasse, and M. Lanza. Insights into system-wide code duplication. In *Proc. of WCRE*, pages 100–109, 2004.
28. L. Sánchez-González, F. García, J. Mendling, and F. Ruiz. Quality assessment of business process models based on thresholds. In *OTM*, pages 78–95, 2010.
29. I. Vanderfeesten, H.A. Reijers, J. Mendling, W.M.P. van der Aalst, and J. Cardoso. On a Quest for Good Process Models: The Cross-Connectivity Metric. In *Proc. of CAiSE*, volume 5074 of *LNCS*, pages 480–494, 2008.
30. J. Vanhatalo, H. Völzer, and F. Leymann. Faster and more focused control-flow analysis for business process models through sese decomposition. In *Proc. of ICSOC*, pages 43–55, 2007.
31. J. Vanhatalo, J. Volzer, and J. Kohler. The Refined Process Structure Tree. *DKE*, 68(9):793–818, 2009.
32. C. Wohlin. *Experimentation in software engineering: an introduction*, volume 6. Springer, 2000.
33. F. Zhang and E.H. D'Hollander. Using Hammock Graphs to Structure Programs. *IEEE T. on Software Engineering*, 30(4):231–245, 2004.