Managing Structural and Textual Quality of Business Process Models

Jan Mendling

HAL Id: hal-01474692
https://hal.inria.fr/hal-01474692

Submitted on 23 Feb 2017

Managing Structural and Textual Quality of Business Process Models

Jan Mendling

Wirtschaftsuniversität Wien, Augasse 2-6, A-1090 Vienna, Austria
`jan.mendling@wu.ac.at`

Abstract. Business process models are increasingly used for capturing business operations of companies. Such models play an important role in the requirements elicitation phase of to-be-created information systems and in as-is analysis of business efficiency. Many process modeling initiatives have grown that big in size involving dozens of modelers with varying expertise creating and maintaining hundreds, sometimes thousands of models. One of the roadblock towards a more effective usage of these process models is the often insufficient provision of quality assurance. The aim of this paper is to give an overview on how empirical research informs structural and textual quality assurance of process models. We present selected findings and show how they can be utilized as a foundation for novel automatic analysis techniques.

1 Introduction

Nowadays, many companies document their business processes in terms of conceptual models. These models provide the basis for activities associated with the business process management lifecycle such as process analysis, process redesign, workflow implementation and process evaluation. Many process modeling initiative have resulted in hundreds or thousands of process models created by process modelers of diverging expertise. One of the major roadblocks towards a more effective usage of these process models is the often insufficient provision of quality assurance. This observation establishes the background for the definition of automatic analysis techniques, which are able to support quality assurance.

In recent years, research into quality assurance of process models and corresponding analysis techniques has offered various new insights. The objective of this paper is to summarize some of the essential contributions in this area. To this end, we aim to integrate both technical contributions and empirical findings. The paper is structured accordingly. In Section 2 we describe the background of quality research distinguishing structural and textual quality. Section 3 discusses how quality factors can be analyzed in terms of their capability to predict aspects of quality. Section 4 discusses different techniques for automatically refactoring process models with the aim to improve their quality. Finally, Section 5 summarizes the discussion and concludes the paper.

2 Background

Research on conceptual modeling often distinguishes syntax, semantics and pragmatics of process models with a reference to semiotic theory [1, 2]. The idea behind this distinction is that a message, here codified as a conceptual model, first has to be understood in terms of its syntax by a model reader before the semantics can be interpreted. Comprehension on the semantic level then provide the foundation for taking appropriate action in a pragmatic way. This semiotic ladder has one major implication for process modeling as a specific area of conceptual modeling and one major research directive. The implication of a semiotic perspective on process modeling is that the comprehension of a process model by a model reader has to be regarded as the central foundation for discussing its quality. As appropriate pragmatics, which comes down to actions taken by a model reader, define the successful progression on the semiotic ladder, research has to establish a thorough understanding how quality on each step of this ladder can be achieved. Indeed, it has been shown empirically that none of the three steps of the semiotic ladder can be neglected, and that they appear to be of equal importance for conceptual modeling [3]. As much of research on process modeling has advanced syntactic analysis of process models, but rather neglected semantic and pragmatic aspects, it is an important directive for future research to complement syntactic analyses with insights on semantics and pragmatics. In the following, we try to give a balanced account of research on process model quality on a syntactic and semantic level while leaving out pragmatics. Our focus in this context is on structural and textual characteristics of a process model.
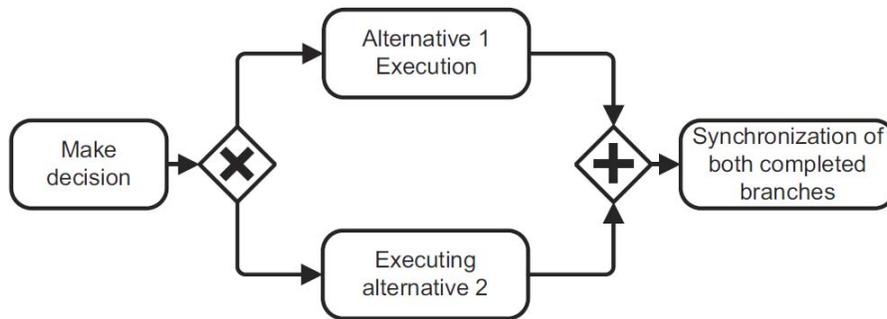
Fig. 1. Example of a process model with structural and textual issues

Figure 1 shows the example of a simple process model in BPMN notation. Process models like this one define the temporal and logical constraints on the control flow between different activities of the process. Here, there are four activities: *Make decision*, *Alternative 1 Execution*, *Executing alternative 2*, and *Synchronization of both completed branches*. The textual labels of these activities describe on the level of domain semantics what this process is supposed to do.

The activity nodes together with the gateways and arcs define the syntax or the formal structure of the process model. In this model, there are two types of gateways used. The first one, an XOR-split, defines a decision point to progress either with the upper or the lower branch, but never with both. There is also a corresponding XOR-join in BPMN, it is not used in the example. Towards the right-hand side of the model, there is an AND-join. This element is used to synchronize concurrent branches. There is no corresponding AND-split in the model. The arcs define the flow relation between activity nodes and gateways.

The quality of a process model like the one in the example can be discussed from the perspective of syntax and of semantics. The quality of the syntax of the model relates to the question whether its formal structure can be readily understood by a model reader. In this context, prior research has focused on the question whether the size and the complexity might be overwhelming. Furthermore, there are formal correctness criteria that can be automatically checked. For the example, we can see that it apparently includes a deadlock: the single branch activated by the XOR-split eventually activates the AND-join, which will then wait forever for the not activated alternative branch to complete. The quality of the semantics of the model relates to the question whether its textual content can be readily understood by a model reader. Here, we observe that the activity labels follow different grammatical structure. *Make decision* starts with a verb and continues with a business object. This is usually considered to be the norm structure of an activity label [4, 5, 6, 7, 8]. The other three labels use a gerund or a noun to express the work content of the activity. Altogether, we can summarize that the example model has both issues with its syntax and with its semantics.

In practice a considerable percentage of process models has quality issues, with often 5% to 30% of the models having problems with soundness [9]. The reason for at least some of these issues is the growth of many process modeling initiatives. This development causes problems at the stage of model creation and model maintenance. An increasing number of employees is becoming involved with modeling. Many of these casual modelers lack modeling experience and adequate training such that newly created models are not always of good quality [10, 11]. Furthermore, the fact that many companies maintain several thousand models calls for automatic quality assurance, which is mostly missing in present tools [10, 11]. A promising direction for increasing process model quality is automatic guideline checking and refactoring. The next section discusses the corresponding foundations.

3 Factors of Process Model Understanding

Various factors for process model understanding have been identified. Characteristics of the *modeling notation* have be investigated in several experiments [12, 13, 14]. Two different factors have to be discussed in this context. First, ontological problems of the notation, e.g. when there are two options to represent the same meaning, might lead to misinterpretations of singular models [15]. Survey research has found support for this argument [16]. Second, properties

of the symbol set of a notation might cause problems, e.g. with remembering or distinguishing them [17]. Empirical support for this hypothesis is reported in [18]. The *secondary notation* plays an important role as well. The concept of secondary notation covers all representational aspects of a model that do not relate to its formal structure. This might relate to the usage of color as a means of highlighting [19]. Corresponding support was found in an experiment in [20]. The visual layout of the model graph is also well-known for its importance to facilitate good understanding [21, 22]. In this section, we focus on structural properties of the process model and properties of its text labels.

3.1 Structural Factors of Process Model Understanding

Structural properties of a process model are typically operationalized by the help of different metrics. Many of them are inspired by metrics from software engineering like lines of code, cyclomatic number, or object metrics [23, 24, 25]. Early contributions in the field of process modeling focus on the definition of metrics [26, 27, 28]. More recent work puts a strong emphasis on the validation of metrics. In these works, different sets of metrics are used as input variables for conducting experiments to test their statistical connection with dependent variables that relate to quality. For instance, the control-flow complexity (CFC) [29] is validated with respect to its correlation with perceived complexity of models [30]. Metrics including size, complexity and coupling are validated for their correlation with understanding and maintainability [31, 32]. Further metrics aim to quantify cognitive complexity and modularity [33, 34, 35, 36]. Various metrics have been validated as predictors of error probability [37], which is assumed to be a symptom of bad understanding by the modeler during the process of model creation. A summary of metrics is presented in [38], an overview of experiments can be found in [39, 40]. In summary, it can be stated that increase in size, decrease in complexity and decrease in structuredness leads is related to greater issues with quality.

One of the major objectives of research into the factors of process model understanding is to establish a set of sound and precise guidelines for process modeling. Guidelines such as the Guidelines of Process Modeling [41] have been available for a while, but they had hardly been tied to experimental findings. The Seven Process Modeling Guidelines (7PMG) might be regarded as a first attempt towards building guidelines based on empirical insight [6]. The challenge in this context is to adapt statistical methods in such a way that metrics can be related to threshold values. In its most basic form, this problem can be formulated as a classification problem: if we consider a particular metric like number of nodes, in how far is it capable of distinguishing e.g. good and bad models.

A specific stream of research in this area investigates in how far different process model metrics are capable of separating models with and without errors. The work reported in [42] uses logistic regression and error probability as a dependent variable. Logistic regression is a statistical model for estimating the probability of binary choices (error or no error in this case) [43]. The logistic regression estimates the odds of error or no error based on the logit function.

This model can be adapted by using structural metrics such as size or complexity of a process model as input variables and observations in terms of whether these models are sound or not. The relationship between input and dependent variables follows an S-shaped curve of the logit curve converging to 0 for $-\infty$ and to 1 for ∞. The value 0.5 is used as a cut-off for predicting error or no error. Based on the coefficient of the input variables in the logit function, one can predict whether an error would be in the model or not.

The quality of such a function to classify process models correctly as having an error or not can be judged based on four different sets: the set of true positive (TP) classifications, the set of false positives (FP), the set of true negatives (TN) and the set of false negatives (FN). A perfect classification based on the logit function would imply that there are only true positives and true negatives. An optimal threshold of separation can then be determined using Receiver Operating Characteristic (ROC) curves [44]. These curves visualize the ability of a specific process metric to discriminate between error and no error models. Each point on the ROC curve defines a pair of *sensitivity* and $1 - specificity$ values of a metric. The best threshold can then be found based on sensitivity and specificity values with: $sensitivity =$ true positives(TP) rate $= \frac{TP}{P}$, $specificity = 1 -$ false positives(FP) rate $= \frac{1-FP}{P}$. Using this approach, several guidelines of the 7PMG could be refined in [42]. Table 1 provides an overview of the results showing, among others, that process models with more than 30 nodes should be decomposed.

Table 1. Ten Process Modeling Rules

Rule	Associated measure	Explanation
G1	Nodes	Do not use more than 31.
G2	Conn. Degree	No more than 3 inputs or outputs per connector.
G3	Start and End	Use no more than 2 start and end events.
G4.a	Structuredness	Model as structured as possible.
G4.b	Mismatch	Use design patterns to avoid mismatch.
G5.a	OR-connectors	Avoid OR-joins and OR-splits.
G5.b	Heterogeneity	Minimize the heterogeneity of connector types.
G5.c	Token Split	Minimize the level of concurrency.
G6	Text	Use verb-object activity labels.
G7	Nodes	Decompose a model with more than 31 nodes.

Although there have been considerable advancements in this area, there are several challenges that persist. Thresholds have been identified based on error probability as a dependent variable, which can be easily expressed in a binary way. An important antecedent of quality is understanding. However, thresholds of understanding are much more difficult to establish as it is mostly measured using score values summed up for a set of comprehension tasks. In this case, good and bad models cannot be exactly discriminated. Furthermore, understanding can be associated with different types of comprehension questions ranging from simple

Table 2. Activity Labeling Styles

Labeling Style	Structure	Example
Verb-Object	Action(Infinitive) + Object	Submit Letter
Action-Noun (np)	Object + Action(Noun)	Letter Submission
Action-Noun (of)	Action(Noun) + 'of' + Object	Submission of Letter
Action-Noun (gerund)	Action(Gerund) + Object	Submitting Letter
Action-Noun (irregular)	*undefined*	Submission: Letter
Descriptive DES	Role + Action(3P) + Object	Student submits Letter
No Action	*undefined*	Letter

recall of a model, understanding its semantics to pragmatic problem solving tasks. Up until now, it has not yet been studied in how far the same or different metrics influence each of these comprehension tasks.

3.2 Labeling Style as a Factor of Process Model Understanding

Empirical research has found that process models from practice do not always follow naming conventions such as the verb-object style for activities. There are three general classes of activity labeling styles [4] (see Figure 2). First, the verb-object style defines an activity label as a verb followed by the corresponding business object (*Make decision*). Second, there are different ways of defining activities as action-noun labels. For such a label, the action is not formulated as a verb, but rather as a gerund (*Executing*) or a substantivated verb (*Execution* from *to execute*). There is also a third category of activity labels that miss referring to an action. An example is the label *information system*, which fails to mention an action, neither as a verb or noun.

With these categories defined, it has to be noted that labeling style is a factor with characteristics quite different to structural metrics. While metrics can be measured on a metric scale, labeling styles can only be distinguished in a nominal way. This means that in the simplest case the input variable can be defined in a binary way, distinguishing usage of verb-object style versus not usage of another style. In terms of defining quality preferences, this makes the task easier: while metrics require a threshold to distinguish good and bad, labeling styles can be directly compared to be better or worse. An experiment reported in [4] takes activity labels of different labeling styles as treatments in order to investigate their potential ambiguity and their usefulness in facilitating domain understanding. ANOVA tests demonstrate that verb-object labels are perceived to be significantly better in this regard, followed by action-noun labels. Labels of the rest category were judged to be most ambiguous.

While the usage of labeling style is covered well in the literature, there are still various challenges in dealing with terminology. From a quality perspective, terms should have a clear-cut meaning. This implies that synonyms (several

words with the same meaning) and homonyms (several meanings of the same word) should be avoided in process modeling. This problem is acknowledged in various papers [45, 46, 47]; however, a proper solution for automatic quality assurance is missing.

4 Automatic Refactoring

The empirical results reported in the previous section provide a basis for the development of automatic refactoring techniques. The general idea of refactoring was formulated for software and relates to "changing a software system in such a way that it does not alter the external behavior of the code, yet improves its internal structure" [48]. For process models, often the notion of trace equivalence [49] or one of the notions of bisimulation [50] is considered when refactoring models. In the following, we summarize work on refactoring the structure of a process model and its activity labels. Frameworks for categorizing refactorings have been proposed in [49, 19, 51].

4.1 Refactoring the Structure of Process Models

Insight into factors of process model comprehension provides a solid basis for optimizing its structure. The challenge in this context is to define a transformation from an unstructured model towards a structured model. It is well known that a structured model can always be constructed for process models without concurrency, but that some concurrent behaviour is inherently unstructured [52]. The research reported in [50] presents a approach based on the identification of ordering relations which leads to a maximally structured model under fully concurrent bisimulation.

Here, two cases have to be distinguished. There are process models for which making them structured comes at the price of increasing its size. Such a case is shown in Figure 2. This increase stems from the duplication of activities in unstructured paths. There are also cases where a process model can be structured without having to duplicate activities. In practice, making a model structured without duplication appears to be rather rare. An investigation with more than 500 models from practice has shown that structuring leads to an increase in size of about 50% on average [53]. It is also important to note that duplication might be more harmful than a usual increase in size. The user experiment reported in [53] points to a potential confusion by model readers who are asked about behavioural constraints that involve activities that are shown multiple times in the model.

The problem of duplicating activities is a key challenge in this area. It is an open research question how the beneficial effects of structuring can be best balanced with the harmful introduction of duplicate activities. Further experiments are needed for identifying a precise specification of the trade-off between structuredness and duplication. In this context, also the size of the model has to be taken into consideration.
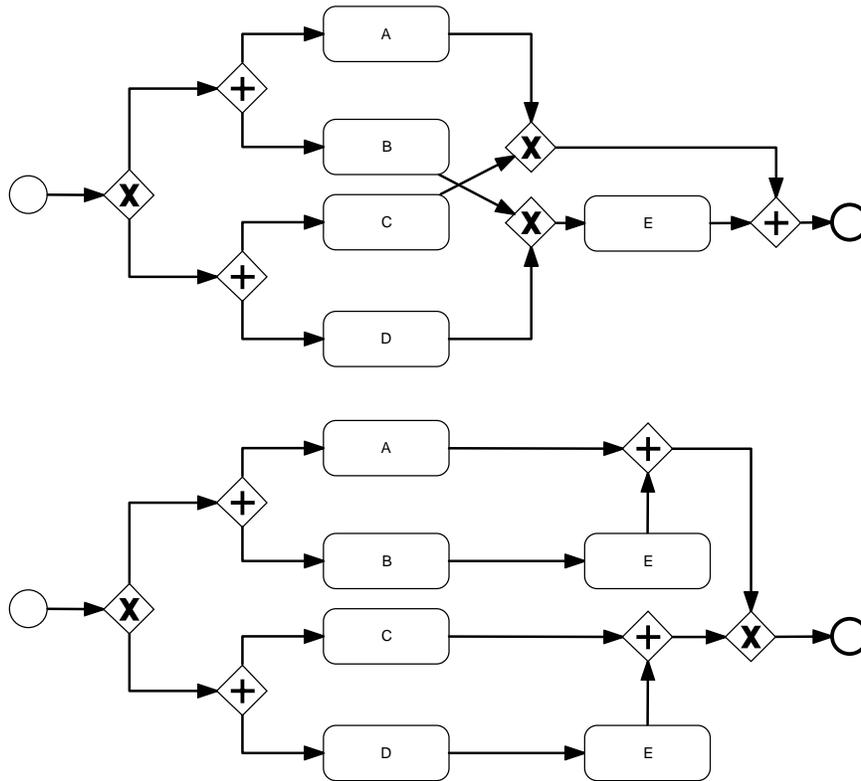
Fig. 2. Example of an unstructured and corresponding structured process model.

4.2 Refactoring Text Labels of Process Models

Experiments and best practices from industry suggest a preference for the verb-object labeling style. The challenge in this context is to achieve an accurate parsing of the different labeling styles such that they can be transformed to the verb-object style. An accurate parsing is difficult in English for two reasons. First, many nouns in English are built from a verb using a zero-derivation mechanism. This means that the noun is morphologically equivalent to the verb. For a word like *plan* we do not directly know whether it refers to a verb or a noun (*the plan* versus *to plan*). Second, the activity labels of a process model usually do not cover a complete grammatically correct sentence structure. Therefore, it has been found difficult to use standard natural language processing tools like the Stanford parser. The approach reported in [54] uses different contextual information to map a label that, for instance, starts with the word *plan* to its correct labeling style. Once the labeling style is known, tools like WordNet can be used to find a verb that matches an action that was formulated as a noun (see Figure 3). It has been shown that this approach works accurately for three

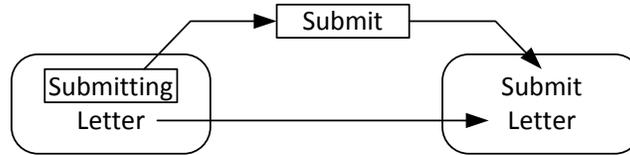different modeling collections from practice including altogether more than 10,000 activity labels [54].

Fig. 3. Example of a label refactored from Action-Noun to Verb-Object style.

It is a topic of ongoing research how these refactoring techniques can be defined in such a way that they do not depend upon the rich set of natural language processing tools available for English. An alternative could be to directly work with annotated corpora. Also, an related to the terminology problem identified above, it is up until now not clear how problems of synonyms and homonyms can be automatically reworked.

5 Conclusion

In this paper we have discussed the management of structural and textual quality of business process models. The growth of many process modeling initiatives towards involving dozens of modelers with varying expertise creating and maintaining thousands of models raises the question of how quality assurance can be defined and implemented in an automatical way. Insights into the factors of process model understanding provide the foundation for building such automatic techniques. On the structural side of process model quality, size and structuredness have been found to be major factors. Guidelines like 7PMG have been formulated based on empirical findings, pointing to the need for rework when certain thresholds are surpassed.

A topic of ongoing research is how refactoring techniques can be defined that balance different structural properties such as size and structuredness while minimizing the duplication of activities. On the side of activity labels, the usage of the verb-object style is recommended. Automatic techniques in this context have to provide an accurate parsing of the labels with a potential reformulation of actions that might be stated as nouns. In this area it is a topic of ongoing research to what extent such automatic techniques for style recognition can be defined without relying on tools like WordNet such that they can be adapted for languages different to English.

References

1. Lindland, O., Sindre, G., Sølvberg, A.: Understanding quality in conceptual modeling. IEEE Software **11**(2) (1994) 42–49
2. Krogstie, J., Sindre, G., Jørgensen, H.: Process Models Representing Knowledge for Action: a Revised Quality Framework. European Journal of Information Systems **15**(1) (2006) 91–102
3. Moody, D., Sindre, G., Brasethvik, T., Sølvberg, A.: Evaluating the quality of process models: Empirical testing of a quality framework. In Spaccapietra, S., March, S.T., Kambayashi, Y., eds.: Conceptual Modeling - ER 2002, 21st International Conference on Conceptual Modeling, Tampere, Finland, October 7-11, 2002, Proceedings. Volume 2503 of Lecture Notes in Computer Science., Springer (2002) 380–396
4. Mendling, J., Reijers, H.A., Recker, J.: Activity Labeling in Process Modeling: Empirical Insights and Recommendations. Information Systems **35**(4) (2010) 467–482
5. Silver, B.: BPMN Method and Style, with BPMN Implementer's Guide. 2nd edn. Cody-Cassidy Press (January 2011)
6. Mendling, J., Reijers, H.A., van der Aalst, W.M.P.: Seven Process Modeling Guidelines (7PMG). Information and Software Technology **52**(2) (2010) 127–136
7. Allweyer, T.: BPMN 2.0 - Business Process Model and Notation. 2nd edn. Books on De-mand GMBH, Norderstedt (2009)
8. Leopold, H., Smirnov, S., Mendling, J.: On the refactoring of activity labels in business process models. Information Systems **37**(5) (2012) 443 – 459
9. Mendling, J.: Empirical Studies in Process Model Verification. LNCS Transactions on Petri Nets and Other Models of Concurrency II, Special Issue on Concurrency in Process-Aware Information Systems **2** (2009) 208–224
10. Rosemann, M.: Potential Pitfalls of Process Modeling: Part A. Business Process Management Journal **12**(2) (2006) 249–254
11. Rosemann, M.: Potential pitfalls of process modeling: part b. Business Process Management Journal **12**(3) (2006) 377–384
12. Sarshar, K., Loos, P.: Comparing the control-flow of epc and petri net from the end-user perspective. In Aalst, W., Benatallah, B., Casati, F., Curbera, F., eds.: Business Process Management, 3rd International Conference, BPM 2005, Nancy, France, September 5-8, 2005, Proceedings. LNCS 3649 (2005) 434–439
13. Hahn, J., Kim, J.: Why are some diagrams easier to work with? effects of diagrammatic representation on the cognitive integration process of systems analysis and design. ACMTCHI: ACM Transactions on Computer-Human Interaction **6** (1999)
14. Agarwal, R., De, P., Sinha, A.: Comprehending object and process models: An empirical study. IEEE Transactions on Software Engineering **25**(4) (1999) 541–556
15. Weber, R.: Ontological Foundations of Information Systems. Coopers & Lybrand and the Accounting Association of Australia and New Zealand, Melbourne, Australia (1997)
16. Recker, J., Rosemann, M., Green, P.F., Indulska, M.: Do ontological deficiencies in modeling grammars matter? MIS Quarterly **35**(1) (2011) 57–79
17. Moody, D.L.: The é;physicsé; of notations: Toward a scientific basis for constructing visual notations in software engineering. IEEE Trans. Software Eng. **35**(6) (2009) 756–779
18. Figl, K., Mendling, J., Strembeck, M.: The influence of notational deficiencies on process model comprehension. Journal of the Association for Information Systems (2012) in press.

19. Rosa, M.L., ter Hofstede, A.H.M., Wohed, P., Reijers, H.A., Mendling, J., van der Aalst, W.M.P.: Managing process model complexity via concrete syntax modifications. IEEE Trans. Industrial Informatics **7**(2) (2011) 255–265
20. Reijers, H.A., Freytag, T., Mendling, J., Eckleder, A.: Syntax highlighting in business process models. Decision Support Systems **51**(3) (2011) 339–349
21. Moher, T., Mak, D., Blumenthal, B., Leventhal, L.: Comparing the Comprehensibility of Textual and Graphical Programs: The Case of Petri Nets. In Cook, C., Scholtz, J., Spohrer, J., eds.: Empirical Studies of Programmers: Fifth Workshop: Papers Presented at the Fifth Workshop on Empirical Studies of Programmers, December 3-5, 1993, Ablex Pub (1993) 137–161
22. Purchase, H.: Which aesthetic has the greatest effect on human understanding? In: Graph Drawing, Springer (1997) 248–261
23. McCabe, T.: A complexity measure. IEEE Transactions on Software Engineering **2**(4) (1976) 308–320
24. Chidamber, S., Kemerer, C.: A metrics suite for object oriented design. IEEE Transaction on Software Engineering **20**(6) (1994) 476–493
25. Fenton, N., Pfleeger, S.: Software Metrics. A Rigorous and Practical Approach. PWS, Boston (1997)
26. Lee, G., Yoon, J.M.: An empirical study on the complexity metrics of petri nets. Microelectronics and Reliability **32**(3) (1992) 323–329
27. Nissen, M.: Redesigning reengineering through measurement-driven inference. MIS Quarterly **22**(4) (1998) 509–534
28. Morasca, S.: Measuring attributes of concurrent software specifications in petri nets. In: METRICS '99: Proceedings of the 6th International Symposium on Software Metrics, Washington, DC, USA, IEEE Computer Society (1999) 100–110
29. Cardoso, J.: Evaluating Workflows and Web Process Complexity. In: Workflow Handbook 2005. Future Strategies, Inc., Lighthouse Point, FL, USA (2005) 284–290
30. Cardoso, J.: Process control-flow complexity metric: An empirical validation. In: Proceedings of IEEE International Conference on Services Computing (IEEE SCC 06), Chicago, USA, September 18-22, IEEE Computer Society (2006) 167–173
31. Canfora, G., García, F., Piattini, M., Ruiz, F., Visaggio, C.: A family of experiments to validate metrics for software process models. Journal of Systems and Software **77**(2) (2005) 113–129
32. Aguilar, E.R., García, F., Ruiz, F., Piattini, M.: An exploratory experiment to validate measures for business process models. In: First International Conference on Research Challenges in Information Science (RCIS). (2007)
33. Vanderfeesten, I., Reijers, H., Mendling, J., Aalst, W., Cardoso, J.: On a Quest for Good Process Models: The Cross-Connectivity Metric. Lecture Notes in Computer Science **5074** (2008) 480–494
34. Vanhatalo, J., Völzer, H., Leymann, F.: Faster and more focused control-flow analysis for business process models through sese decomposition. In Krämer, B., Lin, K.J., Narasimhan, P., eds.: Service-Oriented Computing - ICSOC 2007, Fifth International Conference, Vienna, Austria, September 17-20, 2007, Proceedings. Volume 4749 of Lecture Notes in Computer Science., Springer (2007) 43–55
35. Aalst, W., Lassen, K.: Translating unstructured workflow processes to readable BPEL: Theory and implementation. Information and Software Technology **50**(3) (2008) 131–159
36. Reijers, H.A., Mendling, J., Dijkman, R.M.: Human and automatic modularizations of process models to enhance their comprehension. Inf. Syst. **36**(5) (2011) 881–897

37. Mendling, J., Verbeek, H., Dongen, B., Aalst, W., Neumann, G.: Detection and Prediction of Errors in EPCs of the SAP Reference Model. Data & Knowledge Engineering **64**(1) (January 2008) 312–329
38. Mendling, J.: Metrics for Process Models: Empirical Foundations of Verification, Error Prediction, and Guidelines for Correctness. Volume 6 of LNBIP. Springer (2008)
39. Reijers, H.A., Mendling, J.: A Study Into the Factors That Influence the Understandability of Business Process Models. IEEE Transactions on Systems, Man, and Cybernetics, Part A **41**(3) (2011) 449–462
40. Mendling, J., Strembeck, M., Recker, J.: Factors of process model comprehension - findings from a series of experiments. Decision Support Systems **53**(1) (2012) 195–206
41. Becker, J., Rosemann, M., Uthmann, C.: Guidelines of Business Process Modeling. In van der Aalst, W.M.P., Desel, J., Oberweis, A., eds.: Business Process Management. Models, Techniques, and Empirical Studies. Springer, Berlin (2000) 30–49
42. Mendling, J., Sánchez-González, L., García, F., Rosa, M.L.: Thresholds for error probability measures of business process models. Journal of Systems and Software **85**(5) (2012) 1188–1197
43. Hosmer, D., Lemeshow, S.: Applied Logistic Regression. 2nd edn. John Wiley & Sons (2000)
44. Zweig, M., Campbell, G.: Receiver-operating characteristic (roc) plots: a fundamental evaluation tool in clinical medicine. Clinical chemistry **39**(4) (1993) 561–577
45. Dean, D., Lee, J., Orwig, R., Vogel, D.: Technological support for group process modeling. Journal of Management Information Systems (1994) 43–63
46. Rosemann, M., Muehlen, M.: Evaluation of workflow management systems-a meta model approach. Australian Journal of Information Systems **6** (1998) 103–116
47. Rolland, C.: L'e-lyee: coupling l'ecritoire and lyeeall. Information & Software Technology **44**(3) (2002) 185–194
48. Fowler, M., Beck, K., Brant, J., Opdyke, W., Roberts, D.: Refactoring: improving the design of existing code. Addison-Wesley Professional (1999)
49. Weber, B., Reichert, M., Mendling, J., Reijers, H.A.: Refactoring large process model repositories. Computers in Industry **62**(5) (2011) 467–486
50. Polyvyanyy, A., García-Bañuelos, L., Dumas, M.: Structuring acyclic process models. Inf. Syst. **37**(6) (2012) 518–538
51. Rosa, M.L., Wohed, P., Mendling, J., ter Hofstede, A.H.M., Reijers, H.A., van der Aalst, W.M.P.: Managing process model complexity via abstract syntax modifications. IEEE Trans. Industrial Informatics **7**(4) (2011) 614–629
52. Kiepuszewski, B., Hofstede, A., Bussler, C.: On structured workflow modelling. In Wangler, B., Bergman, L., eds.: Advanced Information Systems Engineering, 12th International Conference CAiSE 2000, Stockholm, Sweden, June 5-9, 2000, Proceedings. Volume 1789 of Lecture Notes in Computer Science., Springer (2000) 431–445
53. Dumas, M., Rosa, M.L., Mendling, J., Mäesalu, R., Reijers, H.A., Semenenko, N.: Understanding business process models: The costs and benefits of structuredness. In Ralyté, J., Franch, X., Brinkkemper, S., Wrycza, S., eds.: CAiSE. Volume 7328 of Lecture Notes in Computer Science., Springer (2012) 31–46
54. Leopold, H., Smirnov, S., Mendling, J.: On the refactoring of activity labels in business process models. Inf. Syst. **37**(5) (2012) 443–459