Applying guidelines is useful to maintain a model understandable, however the procedure of manually verifying their application is expensive in terms of time and error prone. To mitigate this issue, we developed a freely available tool, available also as a service, named BEBoP that automatically verifies 34 of the 50 guidelines. This is the first open source tool that allows checking a large set of modeling guidelines. The guidelines and the tool were developed in the context of the Learn PAd EU project, and we extensively used them on the related cases studies. In the paper, we present an excerpt of a model from the Learn PAd project, mainly with the intention to show the real application of the guidelines in practice. An extended validation using a public repository has been also done to validate if the proposed guidelines, and the corresponding tool, can have some relevance into practice, or if instead modelers already generally define understandable models in reality.

Interesting aspects worthy to be investigated concern the definition of ranking strategies for guidelines, taking into account different application domains. Also the identification of possible negative correlations among guidelines could provide interesting results. We leave such research lines as future work. Other relevant line of research we would like to follow refers to the identification of strategies for the refactoring of models, so that they will abide by the guidelines. This is line with what it has been done by Sánchez-González et al [2015] in relation to the guidelines defined by [Mendling et al, 2010b]. Clearly this is a quite complex topic in particular when interactions and correlations among guidelines are considered. Finally, an interesting topic that we plan to investigate refers to the definition of automatic helping strategies for continuous model improvement.

Within short terms, we intend to extend the application of the guidelines to other scenarios both from the public sector and from the business domain. Further interaction with experts are also planned to allow the possible identification of additional guidelines. On the technical side we intend to extend the current implementation to return more detailed analysis, for instance the degree of satisfaction of some guideline. It is also interesting to include functionalities permitting the derivation of personalized views depending on the application domain. At the same time, we aim at integrating the tool with the BPMN Modeler for the Eclipse IDE[15], to release BEBoP in a general purpose open environment, and to extend BEBoP with the novel NLP-based indicators recently introduced by Laue et al [2016], to improve the quality of model labels.

References

van der Aalst WM (2000) Workflow Verification: Finding Control-Flow Errors Using Petri-Net-Based Techniques. In: Business Process Management, LNCS, vol 1806, Springer, pp 161–183

Allweyer T (2009) BPMN 2.0 - Business Process Model and Notation: Einführung in den Standard für die Geschäftsprozessmodellierung. Books on Demand

Barjis J, Verbraeck A, Gruhn V, Laue R (2010) A heuristic method for detecting problems in business process models. Business Process Management Journal 16(5):806–821

Becker J, Rosemann M, von Uthmann C (2000) Guidelines of Business Process Modeling. In: Business Process Management, LNCS, vol 1806, Springer, pp 30–49

Bernstein V, Soffer P (2015) How Does It Look? Exploring Meaningful Layout Features of Process Models. In: International Conference on Advanced Information Systems Engineering, Springer, LNBIP, vol 215, pp 81–86

Bosshart E, Märki M, Rigert B, Spöcker N, Tanner C (2014) eCH-0158 BPMN-Modellierungskonventionen für die öffent-liche Verwaltung

Cardoso J (2007) Control-flow complexity measurement of processes and weyuker's properties. International Journal of Mathematical, Computational, Physical, Electrical and Computer Engineering 1(8):213–218

[15] `http://www.eclipse.org/bpmn2-modeler/`

Cardoso J, Mendling J, Neumann G, Reijers HA (2006) A discourse on complexity of process models. In: Business process management workshops, Springer, LNCS, vol 4103, pp 117–128

Claes J, Vanderfeesten I, Reijers HA, Pinggera J, Weidlich M, Zugal S, Fahland D, Weber B, Mendling J, Poels G (2012) Tying process model quality to the modeling process: the impact of structuring, movement, and speed. In: International Conference on Business Process Management, Springer, LNCS, vol 7481, pp 33–48

Cognini R, Corradini F, Gnesi S, Polini A, Re B (2016) Business process flexibility - a systematic literature review with a software systems perspective. Information Systems Frontiers - In Press

Comber T, Maltby JR (1996) Investigating layout complexity. Tech. rep., Southern Cross University

Corradini F, Ferrari A, Fornari F, Gnesi S, Polini A, Re B, Spagnolo G (2015) Quality assessment strategy: Applying business process understandability guidelines for learning. Tech. Rep. 4.1, ISTI-CNR, University of Camerino, Italy, URL http://puma.isti.cnr.it/linkdoc.php?idauth=1&idcol=1&icode=2015-TR-034&authority=cnr.isti&collection=cnr.isti&langver=it

Davis R (2001) Business process modelling with ARIS: a practical guide. Springer Science & Business Media

Dumas M, La Rosa M, Mendling J, Mäesalu R, Reijers HA, Semenenko N (2012) Understanding business process models: the costs and benefits of structuredness. In: Advanced Information Systems Engineering, Springer, LNCS, vol 7328, pp 31–46

Dumas M, La Rosa M, Mendling J, Reijers HA (2013) Fundamentals of business process management. Springer

Eichelberger H, Schmid K (2009) Guidelines on the aesthetic quality of UML class diagrams. Information and Software Technology 51(12):1686–1698

Figl K (2017) Comprehension of procedural visual business process models - A literature review. Business & Information Systems Engineering 59(1):41–67

Figl K, Laue R (2015) Influence factors for local comprehensibility of process models. International Journal of Human-Computer Studies 82:96–110

Figl K, Strembeck M (2015) Findings from an experiment on flow direction of business process models. In: 6th International Workshop on Enterprise Modelling and Information Systems Architectures, LNI, vol 248, pp 59–73

Figl K, Derntl M, Rodríguez MC, Botturi L (2010) Cognitive effectiveness of visual instructional design languages. Journal of Visual Languages & Computing 21(6):359–373

Figl K, Mendling J, Strembeck M (2013a) The influence of notational deficiencies on process model comprehension. Journal of the Association for Information Systems 14(6):312 – 338

Figl K, Recker J, Mendling J (2013b) A study on the effects of routing symbol design on process model comprehension. Decision Support Systems 54(2):1104–1118

Gassen JB, Mendling J, Thom LH, de Oliveira JPM (2015) Towards Guiding Process Modelers Depending upon Their Expertise Levels. In: International Conference on Advanced Information Systems Engineering, Springer, LNBIP, vol 215, pp 69–80

Genon N, Heymans P, Amyot D (2011) Analysing the Cognitive Effectiveness of the BPMN 2.0 Visual Notation. In: Software Language Engineering, LNCS, vol 6563, Springer, pp 377–396

Gruhn V, Laue R (2006) Complexity metrics for business process models. In: 9th international conference on business information systems, Lecture Notes in Informatics, vol 85, pp 1–12

Gruhn V, Laue R (2009) Reducing the cognitive complexity of business process models. In: 8th IEEE International Conference on Cognitive Informatics, IEEE, pp 339–345

Gschwind T, Pinggera J, Zugal S, Reijers HA, Weber B (2014) A linear time layout algorithm for business process models. Journal of Visual Languages & Computing 25(2):117–132

Haisjackl C, Pinggera J, Soffer P, Zugal S, Lim SY, Weber B (2015) Identifying Quality Issues in BPMN Models: an Exploratory Study. In: International Conference on Enterprise, Business-Process and Information Systems Modeling, Springer, LNBIP, vol 214, pp 217–230

Heggset M, Krogstie J, Wesenberg H (2014) Ensuring quality of large scale industrial process collections: Experiences from a case study. In: IFIP Working Conference on The Practice of Enterprise Modeling, Springer, LNBIP, vol 197, pp 11–25

Houy C, Fettke P, Loos P (2014) On the theoretical foundations of research into the understandability of business process models. In: 22st European Conference on Information Systems, ECIS 2014, Tel Aviv, Israel, June 9-11, 2014

Indulska M, Green P, Recker J, Rosemann M (2009) Business Process Modeling: Perceived Benefits. In: Conceptual Modeling, LNCS, vol 5829, Springer, pp 458–471

Jeston J, Nelis J (2014) Business process management. Routledge

Johannsen F, Leist S, Braunnagel D (2014) Testing the impact of wand and weber's decomposition model on process model understandability. In: 35th International Conference on Information Systems

Kalpic B, Bernus P (2002) Business process modelling in industry—the powerful tool in enterprise management. Computers in industry 47(3):299–318

Khlif W, Makni L, Zaaboub N, Ben-Abdallah H (2009) Quality metrics for business process modeling. In: 9th international conference on Applied computer science, World Scientific and Engineering Academy and Society, pp 195–200

Kitchenham B (2007) Guidelines for performing Systematic Literature Reviews in Software Engineering. Tech. rep., Keele University

Kluza K, Nalepa GJ (2012) Proposal of square metrics for measuring business process model complexity. In: Federated Conference on Computer Science and Information Systems, IEEE, pp 919–922

Kluza K, Baran M, Bobek S, Nalepa GJ (2013) Overview of Recommendation Techniques in Business Process Modeling*. Knowledge Engineering and Software Engineering

Kluza K, Nalepa GJ, Lisiecki J (2014) Square Complexity Metrics for Business Process Models. In: Advances in Business ICT, AISC, vol 257, Springer, pp 89–107

Koehler J, Vanhatalo J (2007) Process anti-patterns: How to avoid the common traps of business process modeling. IBM WebSphere Developer Technical Journal 10(2)

Koschmider A, Ullrich M, Heine A, Oberweis A (2015) Revising the vocabulary of business process element labels. In: 27th International Conference Advanced Information Systems Engineering, Springer, LNCS, vol 9097, pp 69–83

Krogstie J (2012) Quality of Business Process Models. In: The Practice of Enterprise Modeling, no. 134 in LNBIP, Springer, pp 76–90

Krogstie J (2015) SEQUAL as a Framework for Understanding and Assessing Quality of Models and Modeling Languages. Encyclopedia of Information Science and Technology, Third Edition

Krogstie J, Sindre G, Jørgensen Ha (2006) Process models representing knowledge for action: a revised quality framework. European Journal of Information Systems 15(1):91–102

Kummer T, Recker J, Mendling J (2016) Enhancing understandability of process models through cultural-dependent color adjustments. Decision Support Systems 87:1–12

Kunze M, Berger P, Weske M (2012) BPM Academic Initiative - Fostering Empirical Research. In: BPM (Demos), Tallinn, Estonia, pp 1 – 5

La Rosa M, Ter Hofstede AH, Wohed P, Reijers HA, Mendling J, van der Aalst WM (2011) Managing process model complexity via concrete syntax modifications. IEEE Transactions on Industrial Informatics 7(2):255–265

Lassen KB, van der Aalst WM (2009) Complexity metrics for Workflow nets. Information and Software Technology 51(3):610–626

Latva-Koivisto AM (2001) Finding a complexity measure for business process models. Tech. rep., Helsinki University of Technology

Laue R, Awad A (2011) Visual suggestions for improvements in business process diagrams. Journal of Visual Languages & Computing 22(5):385–399

Laue R, Koop W, Gruhn V (2016) Indicators for Open Issues in Business Process Models. In: International Working Conference on Requirements Engineering: Foundation for Software Quality, Springer, LNCS, vol 9619, pp 102–116

Leopold H, Smirnov S, Mendling J (2010) Refactoring of process model activity labels. In: 15th International Conference on Applications of Natural Language to Information Systems, Springer, LNCS, vol 6177, pp 268–276

Leopold H, Smirnov S, Mendling J (2012) On the refactoring of activity labels in business process models. Information Systems 37(5):443–459

Leopold H, Eid-Sabbagh RH, Mendling J, Azevedo LG, Baião FA (2013) Detection of naming convention violations in process models for different languages. Decision Support Systems 56:310–325

Leopold H, Mendling J, Gunther O (2015) What we can learn from Quality Issues of BPMN Models from Industry. IEEE Software 33(4)

Leopold H, Mendling J, Günther O (2016) Learning from Quality Issues of BPMN Models from Industry. In: 7th International Workshop on Enterprise Modeling and Information Systems Architectures, Vienna, Austria, pp 36–39

Leopold H, Pittke F, Mendling J (2017) Ensuring the canonicity of process models. Data & Knowledge Engineering - In Press

Melcher J, Seese D (2008) Towards validating prediction systems for process understandability: Measuring process understandability. In: 10th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, pp 564–571

Melcher J, Mendling J, Reijers HA, Seese D (2010) On measuring the understandability of process models. In: Business Process Management Workshops, Springer, LNBIP, vol 43, pp 465–476

Mendling J (2008) Metrics for Business Process Models, LNBIP, vol 6. Springer

Mendling J, Recker JC (2008) Towards systematic usage of labels and icons in business process models. In 13th International Workshop on Exploring Modeling Methods for Systems Analysis and Design June 16-17, 2008, Montpellier, France

Mendling J, Reijers HA (2008) The impact of activity labeling styles on process model quality. In: SIGSAND-EUROPE, pp 117–128

Mendling J, Strembeck M (2008) Influence factors of understanding business process models. In: Business Information Systems, Springer, LNBIP, vol 7, pp 142–153

Mendling J, Neumann G, Van Der Aalst W (2007a) Understanding the occurrence of errors in process models based on metrics. In: On the Move to Meaningful Internet Systems, LNCS, vol 4803, Springer, pp 113–130

Mendling J, Reijers HA, Cardoso J (2007b) What makes process models understandable? In: Business Process Management, Springer, LNCS, vol 4714, pp 48–63

Mendling J, Verbeek HMW, van Dongen BF, van der Aalst WM, Neumann G (2008) Detection and prediction of errors in EPCs of the SAP reference model. Data & Knowledge Engineering 64(1):312–329

Mendling J, Recker J, Reijers HA (2010a) On the usage of labels and icons in business process modeling. International Journal of Information System Modeling and Design 1(2):40–58

Mendling J, Reijers HA, van der Aalst WM (2010b) Seven process modeling guidelines (7pmg). Information and Software Technology 52(2):127–136

Mendling J, Reijers HA, Recker J (2010c) Activity labeling in process modeling: Empirical insights and recommendations. Information Systems 35(4):467–482

Mendling J, Sanchez-Gonzalez L, Garcia F, La Rosa M (2012a) Thresholds for error probability measures of business process models. Journal of Systems and Software 85(5):1188–1197

Mendling J, Strembeck M, Recker J (2012b) Factors of process model comprehension—findings from a series of experiments. Decision Support Systems 53(1):195–206

Moody DL, Sindre G, Brasethvik T, Sølvberg A (2002) Evaluating the quality of process models: Empirical testing of a quality framework. In: International Conference on Conceptual Modeling, Springer, LNCS, vol 2503, pp 380–396

Muketha GM, Ghani AAA, Selamat MH, Atan R (2010) A survey of business process complexity metrics. Information Technology Journal 9(7):1336–1344

Moreno de Oca I, Snoeck M (2014) Pragmatic guidelines for business process modeling. Technical Report 2592983, KU Leuven, Faculty of Economics and Business

Moreno de Oca I, Snoeck M, Reijers HA, Rodríguez-Morffi A (2015) A systematic literature review of studies on business process modeling quality. Information and Software Technology 58:187–205

OMG (2011) Business Process Model and Notation (BPMN V 2.0)

Overhage S, Birkmeier DQ, Schlauderer S (2012) Quality marks, metrics, and measurement procedures for business process models. Business & Information Systems Engineering 4(5):229–246

Petrusel R, Mendling J, Reijers HA (2016) Task-specific visual cues for improving process model understanding. Information and Software Technology 79:63–78

Pittke F, Leopold H, Mendling J (2013) Spotting terminology deficiencies in process model repositories. In: Enterprise, Business-Process and Information Systems Modeling, Springer, LNBIP, vol 147, pp 292–307

Pittke F, Leopold H, Mendling J (2014) When language meets language: Anti patterns resulting from mixing natural and modeling language. In: International Conference on Business Process Management - Workshops, Springer, LNBIP, vol 202, pp 118–129

Pittke F, Leopold H, Mendling J (2016) Automatic detection and resolution of lexical ambiguity in process models. IEEE Transactions on Software Engineering 41(6):526 – 544

Purchase HC (2002) Metrics for graph drawing aesthetics. Journal of Visual Languages & Computing 13(5):501–516

Reggio G, Leotta M, Ricca F (2011) “Precise is better than light” a document analysis study about quality of business process models. In: International Workshop on Empirical Requirements Engineering, IEEE, pp 61–68

Reijers H, Mendling J, others (2011a) A study into the factors that influence the understandability of business process models. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on 41(3):449–462

Reijers HA, Mendling J (2008) Modularity in process models: Review and effects. In: International Conference on Business Process Management, Springer, LNCS, vol 5240, pp 20–35

Reijers HA, Vanderfeesten IT (2004) Cohesion and coupling metrics for workflow process design. In: International Conference on Business Process Management, LNCS, vol 3080, Springer, pp 290–305

Reijers HA, Mendling J, Dijkman R (2010) On the usefulness of subprocesses in business process models. Tech. rep., Eindhoven - BPMcenter.org

Reijers HA, Freytag T, Mendling J, Eckleder A (2011b) Syntax highlighting in business process models. Decision Support Systems 51(3):339–349

Reijers HA, Mendling J, Recker J (2015) Business Process Quality Management. In: Handbook on Business Process Management 1, Springer, pp 167–185

Reynoso L, Rolón E, Genero M, García F, Ruiz F, Piattini M (2009) Formal Definition of Measures for BPMN Models. In: Software Process and Product Measurement, LNCS, vol 5891, Springer, pp 285–306

Rolón E, Ruiz F, García F, Piattini M (2006) Applying software metrics to evaluate business process models. CLEI-Electronic Journal 9(1)

Rolón E, Sanchez L, Garcia F, Ruiz F, Piattini M, Caivano D, Visaggio G (2009) Prediction Models for BPMN Usability and Maintainability. IEEE, pp 383–390

Sánchez-González L, García F, Mendling J, Ruiz F (2010) Quality assessment of business process models based on thresholds. In: On the Move to Meaningful Internet Systems, LNCS, vol 6426, Springer, pp 78–95

Sánchez González L, García Rubio F, Ruiz González F, Piattini Velthuis M (2010) Measurement in business processes: a systematic review. Business Process Management Journal 16(1):114–134

Sánchez-González L, Ruiz F, García F, Cardoso J (2011) Towards thresholds of control flow complexity measures for BPMN models. In: Proceedings of the 2011 ACM Symposium on Applied Computing, pp 1445–1450

Sánchez-González L, García F, Ruiz F, Mendling J (2012) Quality indicators for business process models from a gateway complexity perspective. Information and Software Technology 54(11):1159–1174

Sánchez-González L, García F, Ruiz F, Piattini M (2013a) Toward a quality framework for business process models. International Journal of Cooperative Information Systems 22(01)

Sánchez-González L, Ruiz F, García F, Piattini M (2013b) Improving Quality of Business Process Models. In: International Conference on Evaluation of Novel Approaches to Software Engineering, CCIS, vol 275, Springer, pp 130–144

Sánchez-González L, García F, Ruiz F, Piattini M (2015) A case study about the improvement of business process models driven by indicators. Software & Systems Modeling 16(3):759–788

Sears A (1993) Layout appropriateness: A metric for evaluating user interface widget layout. Software Engineering, IEEE Transactions on 19(7):707–719

Sharp A, McDermott P (2009) Workflow modeling: tools for process improvement and applications development. Artech House

Signavio (2014) Modeling Guidelines Manual. URL `https://www.signavio.com/resources/other-resources/modeling-guidelines-manual/`

Silingas D, Mileviciene E (2011) Refactoring BPMN Models: From 'Bad Smells' to Best Practices and Patterns. BPMN 20 Handbook Second Edition: Methods, Concepts, Case Studies and Standards in Business Process Management Notation pp 125–134

Silver B (2011) BPMN method and style: with BPMN implementer's guide. Cody-Cassidy

Snoeck M, de Oca IM, Haegemans T, Scheldeman B, Hoste T (2015) Testing a selection of BPMN tools for their support of modelling guidelines. In: 8th IFIP WG 8.1. Working Conference on The Practice of Enterprise Modeling, LNBIP, vol 235, pp 111–125

Vanderfeesten I, Reijers HA, Mendling J, van der Aalst WM, Cardoso J (2008) On a quest for good process models: the cross-connectivity metric. In: Advanced Information Systems Engineering, Springer, LNCS, vol 5074, pp 480–494

Vom Brocke J, Rosemann M, others (2010) Handbook on business process management. Springer

Weber B, Reichert M, Mendling J, Reijers HA (2011) Refactoring large process model repositories. Computers in Industry 62(5):467–486

Webster J, Watson RT (2002) Analyzing the past to prepare for the future: Writing a literature review. MIS Quarterly 26(2):xiii–xxiii

Weske M (2012) Business Process Management. Springer, Berlin, Heidelberg

White SA (2008) BPMN modeling and reference guide: understanding and using BPMN. Future Strategies Inc.

A Guidelines List

In the following (Tables 9) we report all the defined guidelines with their ID, Names, Description and Source. For major details please refer to Corradini et al [2015]. We remind that the guidelines are grouped in the following categories: *General*, guidelines that impact on different aspects of the overall BPMN modeling practice (9 guidelines from ID 1 to ID 9); *Notation*, guidelines on the usage of the BPMN syntax (16 guidelines from ID 10 to ID 25); *Labelling*, guidelines for the assignment of proper labels to BPMN elements (14 guidelines from ID 26 to ID 39); *Patterns*, patterns guidelines that may be applied in the arrangement of BPMN elements (3 guidelines from ID 40 to ID 42); *Appearance*, guidelines for having a clear presentation of the BP model (8 guidelines from ID 43 to ID 50).

The guidelines for which a quantitative analysis is possible have the ID represented in bold in the following tables, and the check is implemented in the provided tool. It is worth mentioning that for some of the guidelines only qualitative aspects are considered. Indeed we decided to not provide quantitative metrics that somehow resulted to be too much complex to compute, and we thought that it is much reasonable to leave the judgement to the modeler, with respect to the satisfaction of such guidelines. For instance in this class are included those metrics related to natural language aspects that would have required the usage of complex NLP techniques in order to derive a measure.

ID	Name	Description
1	Validate models	The designer should create models which comply with the BPMN standard. Once the process logic has been defined, the designer should validate a model ensuring that the model is syntactically correct. **Sources**: [Silver, 2011; Leopold et al, 2015; Laue and Awad, 2011]
2	Minimize model size	The designer should try to keep models as small as possible. Large models tend to contain more errors. Additionally they are difficult to read and comprehend. Defining the correct scope of tasks and level of detail of models is the key to reduce the overage of information. **Sources**: [Mendling et al, 2010b, 2012a; Dumas et al, 2013; Weber et al, 2011; Dumas et al, 2012; Reijers et al, 2011a; Purchase, 2002; Mendling et al, 2008; Sánchez-González et al, 2013b; Leopold et al, 2015; Moreno de Oca and Snoeck, 2014; Gschwind et al, 2014; Mendling and Strembeck, 2008; Johannsen et al, 2014; Sánchez-González et al, 2011]
3	Apply hierarchical structure with sub-processes	The designer should create a hierarchical model structure. BPMN sub-processes are used to split the process into layers. The designer can expand the sub-processes later to expose details of lower levels of hierarchy. A process model will contain multiple layers, but internally the integrity of a single model has to be maintained. **Sources**: [Silver, 2011; Weber et al, 2011; Sánchez-González et al, 2013b; Johannsen et al, 2014; Mendling et al, 2007a; Reijers et al, 2010; Reijers and Mendling, 2008]
4	Apply symmetric modeling	The designer should model as structured as possible. Symmetric structures increase understandability of models for both experienced and inexperienced users. Well-structuredness, means that for every node with multiple outgoing arcs (a split) there is a corresponding node with multiple incoming arcs (a join), such that the set of nodes between the split and the join form a single-entry-single-exit (SESE) region. **Sources**: [Mendling et al, 2010b; Koehler and Vanhatalo, 2007; Mendling et al, 2008; Moreno de Oca and Snoeck, 2014; Laue and Awad, 2011; Mendling et al, 2012a, 2007a; Dumas et al, 2012]
5	Highlight the "happy path"	The designer should make the process logic visible in the model. The "happy path" - a sequence of activities that will be executed if everything goes as expected without exceptions - should be easily identified when reading a model. The designer should model the happy path first and then the alternative flows. **Sources**: `http://www.bpmnquickguide.com/viewit.html`
6	Minimize concurrency	The designer should minimize the level of concurrency which means to reduce the use of parallel gateways and ad-hoc sub-processes. Concurrency, which is represented by parallel gateways, may generate ambiguity, especially if the activities in parallel are "manual tasks" and only one person is responsible for those. In this case there will be no parallelization but it is up to the person to decide the tasks execution order. **Sources**: [Mendling et al, 2010b, 2012a, 2007a; Moreno de Oca and Snoeck, 2014; Gruhn and Laue, 2009]
7	Model loops via loop activities	The designer should model a loop via activity looping (with the loop marker) instead of using a sequence flow looping; this, where possible, and if this practice actually contributes to simplify the model. **Sources**: [Moreno de Oca and Snoeck, 2014; Mendling et al, 2012a]
		Continued on next page

ID	Name	Description
8	Provide activity descriptions	The designer should provide a brief description for each activity in the model. **Sources**: [Signavio, 2014]
9	Minimize gateway heterogeneity	The designer should minimize the heterogeneity of gateway types. The use of several type of gateway may cause confusion. **Sources**: [Mendling et al, 2010b, 2012a; Moreno de Oca and Snoeck, 2014]
10	Use pools consistently	The designer should define as many pools as processes and/or participants. Use a black-box pool to represent external participant/processes. The modeled pools need to be in relation with each other and have to be linked to the main pool through message exchange. **Sources**: [Bosshart et al, 2014; Signavio, 2014]
11	Use lanes consistently	The designer should model internal organizational units as lanes within a single process pool, not as separate pools; separate pools imply independent processes. The designer should create a lane, in a pool, only if at least one activity or intermediate event is performed in it. **Sources**: Bosshart et al [2014]; Signavio [2014]
12	Use start and end events explicitly	The designer should explicitly make use of start and end events. The use of start and end events is necessary to represent the different states that begin and complete the modeled process. Processes with implicit start and end events are undesirable and could lead to misinterpretations. **Sources**: [Bosshart et al, 2014; Signavio, 2014; Mendling et al, 2010b, 2012a; White, 2008; Moreno de Oca and Snoeck, 2014; Claes et al, 2012]
13	Use start events consistently	The designer should include, in the model, only one start event. Where necessary, alternative instantiations of the process should be depicted with separate start events and using a event-based start gateway. **Sources**: [Bosshart et al, 2014; Signavio, 2014; Mendling et al, 2010b, 2012a; Sánchez-González et al, 2013b; Moreno de Oca and Snoeck, 2014; Mendling et al, 2007a; Claes et al, 2012]
14	Use end events consistently	The designer should distinguish success and failure end states in a process or a sub-process with separate end events. Therefore, separate end events that do not represent distinct end states must be merged in a single end event. **Sources**: [Bosshart et al, 2014; Signavio, 2014; Mendling et al, 2010b, 2012a, 2007a; Claes et al, 2012]
15	Restrict usage of terminate end event	The designer should use terminate events only when strictly necessary. They are used to model situations where several alternative paths are enabled and the entire process have to be finished when one of them is completed. The designer should use other end events rather than the terminate end event (e.g. a generic end event), to guarantee that the executions of the reaming process paths or activities will not be stopped.
16	Use explicit gateways	The designer should split or join sequence flows always using gateways. The designer should not split or join flows using activities or events. This includes that an activity can have only one incoming sequence flow and only one outgoing sequence flow. **Sources**: [Bosshart et al, 2014; Signavio, 2014; Mendling et al, 2010b, 2012a; Leopold et al, 2015; Silingas and Mileviciene, 2011]
17	Mark exclusive gateways	The designer should use the Exclusive Gateway with the marker “X” instead of using it without marker. **Sources**: [Bosshart et al, 2014; Signavio, 2014]
18	Split and join flows consistently	The designer should not use gateways to join and split at the same time. **Sources**:[Bosshart et al, 2014; Signavio, 2014; Mendling et al, 2010b, 2012a; Moreno de Oca and Snoeck, 2014; Gschwind et al, 2014; Krogstie, 2012; Silver, 2011]
19	Balance gateways	The designer should always use the same type of gateway for splitting and joining the flow. In particular, the designer should ensure that join parallel gateways have the correct number of incoming sequence flows especially when used in conjunction with other gateways; this is related to ensuring the soundness property. Do not apply this guidelines on Event-based or Complex Gateways. **Sources**: [Mendling et al, 2010b, 2012a; White, 2008; Mendling et al, 2008; Laue and Awad, 2011; Mendling et al, 2007a; Dumas et al, 2012]
20	Use meaningful gateways	The designer should not represent gateways that have only one incoming and only one outgoing sequence flow. Gateways with only one incoming and one outgoing sequence flow do not provide any added value. **Sources**: [Bosshart et al, 2014; Signavio, 2014; Mendling et al, 2010b, 2012a; Weber et al, 2011; Koehler and Vanhatalo, 2007]

Continued on next page

ID	Name	Description
21	Minimize inclusive OR gateways	The designer should minimize the use of inclusive gateways (OR). Inclusive OR-splits activate one, several, or all subsequent branches based on conditions. They need to be synchronized with inclusive OR-join elements, which are difficult to understand in the general case. **Sources**: [Mendling et al, 2010b; White, 2008; Mendling et al, 2007a; Sánchez-González et al, 2013b; Gruhn and Laue, 2009]
22	Use default flows	Where possible, after an exclusive and an inclusive gateway, the designer should express the default flow. One way for the modeler to ensure that the process does not get stuck at a gateway is to use a default condition for one of the outgoing sequence flow. This default sequence flow will always evaluate to true if all the other sequence flow conditions turn out to be false. **Sources**: [White, 2008]
23	Use messages consistently	The designer could represent message exchange with different elements. A clearer usage of those elements would be: • Send Task, can be used to express that the sending of a message requires an effort such as: making a phone call, sending an email, delivering a document, accessing a data store to retrieve data, etc. • Receive Task, can be used to express that the receiving of a message requires an effort such as: answering a phone call, checking the email, collecting documents, storing data on a data store, etc. • Intermediate Throwing Event, can be used to express that the sending of a message does not require particular effort e.g. the message is automatically processed by a system. • Intermediate Catching Event, can be used to express that the receiving of a message does not require particular effort e.g. the message is received and automatically processed by a system. • For other cases of message exchange, the designer should use the remaining Message events such as: Message Start Event (if the process starts after receiving a message); Message Event SubProcess Interrupting/Non- interrupting (if a received message starts a sub-process); Message Boundary Interrupting/Non-interrupting (if a message is received by a sub-process); Message End Event (if the process or sub-process, ends after sending a message).
24	Use message flows	The designer should represent message flows for each message events and send or receive tasks. If in a sub-process are present more message flows to the same pool, the designer should show in the top-level process maximum two message flows: one for all outgoing message flow and one for all incoming message flow with that pool. **Sources**: [OMG, 2011]
25	Use task types consistently	The designer should distinguish task types e.g. manual task, user tasks and service tasks. **Sources**: [Silver, 2011; Krogstie, 2012]
26	Document minor details	The designer should leave details to documentation keeping labels simple and limiting the use of text annotations. **Sources**: [Mendling et al, 2010a]
27	Use a labeling convention	The designer should not use short names or abbreviations. The designer should always use keywords that are meaningful to the business; he should not use the element type in its name. The name should emphasize the goal, and details of activity can be captured in comments or documentation. The designer should not use conjunctions in names raise name abstraction level or split into two subsequent/alternative activities. **Sources**: [Overhage et al, 2012; Weber et al, 2011; Leopold et al, 2015; Silingas and Mileviciene, 2011; Moreno de Oca and Snoeck, 2014; Mendling and Strembeck, 2008; Koschmider et al, 2015; Leopold et al, 2010, 2012; Pittke et al, 2013; Mendling and Reijers, 2008]
28	Labelling pools	The designer should label pools using the participants name. An exception can be done for the main pool: it can be labeled using the process name. If a pool is present in a sub-process, the name of the pool must be the same of the upper-level process pool which includes the sub-process activity. This means that the pool of the upper-level process and the pool of the sub-process needs to be the same. **Sources**: [Signavio, 2014; Mendling and Strembeck, 2008; Leopold et al, 2013; Koschmider et al, 2015; Leopold et al, 2010, 2012; Pittke et al, 2013]
		Continued on next page

ID	Name	Description
29	Labelling lanes	The designer should always assign a label to lanes. The label should identify the responsible entity for the process. Lanes are often used for representing things as internal roles (e.g., manager, associate), systems (e.g., an enterprise application), or internal departments (e.g., shipping, finance). **Sources**: [Signavio, 2014; Mendling and Strembeck, 2008; Leopold et al, 2013; Koschmider et al, 2015; Leopold et al, 2010, 2012; Pittke et al, 2013]
30	Labelling activities	The designer should label activities with one verb, and one object. The verb used should use the present tense and be familiar to the organization. The object has to be qualified and also of meaning to the business. The designer should not label multiple activities with the same name, except for same Call Activities used many time in the process. Send and receive verbs should be present only for sending and receiving activities. **Sources**: [Silver, 2011; Signavio, 2014; Mendling et al, 2010b, 2012a, 2010a; Sánchez-González et al, 2013b; Leopold et al, 2015; Mendling et al, 2010c; Leopold et al, 2013; Silingas and Mileviciene, 2011; Moreno de Oca and Snoeck, 2014; Mendling and Strembeck, 2008; Koschmider et al, 2015; Leopold et al, 2010, 2012; Pittke et al, 2013]
31	Labelling events	The designer should model all events with a label representing the state of the process: • Events of type message, signal, escalation, and error events should be labeled with a past participle using an active verb; • Link events should be labeled with a noun; • Timer events should be labeled with time-date or schedule; • Conditional events should be labeled with the condition that triggers them. **Sources**: [Signavio, 2014; Leopold et al, 2015; Mendling et al, 2010c; Leopold et al, 2013; Mendling and Strembeck, 2008; Koschmider et al, 2015; Leopold et al, 2010, 2012; Pittke et al, 2013]
32	Labelling start and end events	The designer should not label start untyped and end untyped event if there is only one instance of them. The designer should use labeling when multiple start and end events are used. Label them according to what they represent using a noun. Do not repeat names. **Sources**: [Signavio, 2014; Mendling et al, 2010c; Leopold et al, 2013; Mendling and Strembeck, 2008; Koschmider et al, 2015; Leopold et al, 2010, 2012; Pittke et al, 2013]
33	Labelling message events	The designer should draw a message flow whenever he uses a message event, and he should label the event. When a focus on the message itself is required, the designer can represent a message icon and label it with the name of the message. **Sources**: [Signavio, 2014; Mendling and Strembeck, 2008; Leopold et al, 2013; Koschmider et al, 2015; Leopold et al, 2010, 2012; Pittke et al, 2013]
34	Labelling XOR gateways	The designer should label XOR split gateways with an interrogative phrase (do not label XOR join-gateways). Sequence flows coming out of diverging gateways should be labeled using their associated conditions stated as outcomes. **Sources**: [Signavio, 2014; Leopold et al, 2015; Mendling et al, 2010c; Leopold et al, 2013; Mendling and Strembeck, 2008; Koschmider et al, 2015; Leopold et al, 2010, 2012; Pittke et al, 2013]
35	Labelling AND-gateways	The designer should omit labels on AND-splits and joins (and sequence flows connecting them); they add no new information, so it is best to omit them. **Sources**: [Signavio, 2014; Leopold et al, 2015; Mendling et al, 2010c; Leopold et al, 2013; Mendling and Strembeck, 2008; Koschmider et al, 2015; Leopold et al, 2010, 2012; Pittke et al, 2013]
36	Labelling converging gateways	The designer should not label converging gateways. When the convergence logic is not obvious, the designer should associate a text annotation to the gateway. **Sources**: [Signavio, 2014; White, 2008; Leopold et al, 2015; Mendling and Strembeck, 2008; Leopold et al, 2013; Koschmider et al, 2015; Leopold et al, 2010, 2012; Pittke et al, 2013]
37	Labelling data object	The designer should label data objects using a qualified noun that is the name of a business object. The designer should label multiple instances of the same data object (which are really data object references) using a matching label followed by the applicable state in square brackets. **Sources**: [Signavio, 2014; Mendling and Strembeck, 2008; Leopold et al, 2013; Koschmider et al, 2015; Leopold et al, 2010, 2012; Pittke et al, 2013]

Continued on next page

ID	Name	Description
38	Labelling synchronized end/split	The designer should use gateways and sub-processes consistently. The designer should match the labels of sub-process end states with the labels of a gateway immediately following the sub-process; this allows to have a clear vision on how sub-process and process are linked together. **Sources**: [Signavio, 2014; Mendling and Strembeck, 2008; Leopold et al, 2013; Koschmider et al, 2015; Leopold et al, 2010, 2012; Pittke et al, 2013]
39	Include loop marker annotations	The designer should associate a text annotation to a loop represented with a loop marker so to express the condition (which alternatively is hidden).
40	Reduce the number of redundant activities	The designer should integrate activities (without boundary events) that can be performed by the same person. The designer can represent these activities as a single activity or he can represent them in a sub-process. A set of consecutive activities in the same lane (or in a pool without lanes) may indicate missing participant details, too much detail, or a misalignment in scope. **Sources**: [Weber et al, 2011; Moreno de Oca and Snoeck, 2014]
41	Use sub-processes	The designer should make use of sub-processes to group activities with the same purpose when: • A set of consecutive activities has an owner different from the main process owner; • A set of consecutive activities has a different goal from the main process one; • A process or a fragment must be re-used in another process (use Call Activities in this case). **Sources**: [Purchase, 2002; Silingas and Mileviciene, 2011; Moreno de Oca and Snoeck, 2014; Johannsen et al, 2014]
42	Use sub-processes to scope attached events	The designer should use a sub-process with attached event to clearly define the scope of an event. If the response to the handling of an exception (in the use of boundary events) is the same for every activity within a contiguous segment of the process, the designer should not attach the same boundary event to all the activities and he should not represent the same exception flows multiple times. The correct way, the designer should model it, is to enclose that segment in a sub-process and attach a single boundary event to the sub-process boundary. **Sources**: [Silingas and Mileviciene, 2011]
43	Design neat and consistent models	The designer should keep the model as neat and consistently organized as possible by following the following list of advices: Maximize the number of orthogonally drawn connecting objects; Make your models long and thin (instead of square): maximize the number of connecting objects respecting workflow direction; Minimize the drawing area; Adapt the size of objects such that elements have enough space; Use a uniform style for flow layout. **Sources**: [Weber et al, 2011; Moreno de Oca and Snoeck, 2014]
44	Avoid overlapping elements	The designer should avoid overlapping, or crossing, BPMN elements. **Sources**: [Bosshart et al, 2014; Signavio, 2014; Leopold et al, 2015; Moreno de Oca and Snoeck, 2014; Gschwind et al, 2014; Figl and Strembeck, 2015; Kummer et al, 2016]
45	Use linear sequence flows	The designer should use linear sequence flows without useless foldings; it helps to maintain the model clear. **Sources**: [Bosshart et al, 2014; Signavio, 2014; Purchase, 2002; Moreno de Oca and Snoeck, 2014; Gschwind et al, 2014; Figl and Strembeck, 2015; Kummer et al, 2016]
46	Use linear message flows	The designer should use linear message flows without useless foldings; it helps to maintain the model clear. **Sources**: [Bosshart et al, 2014; Signavio, 2014; Moreno de Oca and Snoeck, 2014; Figl and Strembeck, 2015; Kummer et al, 2016]
47	Use a consistent process orientation	The designer should draw pools horizontally and use consistent layout with horizontal sequence flows, and vertical message flows and associations. **Sources**: [Bosshart et al, 2014; Signavio, 2014; Bernstein and Soffer, 2015; Leopold et al, 2015; Moreno de Oca and Snoeck, 2014; Figl and Strembeck, 2015; Kummer et al, 2016]
48	Organize artifacts flows	The designer should group artifacts flows, if there are several artifacts. The designer should pick a point on the boundary of an activity and have all the flows connected to that point. If there are multiple flows for the same artifact, the designer should group the flows. **Sources**: [Bosshart et al, 2014; Signavio, 2014; White, 2008; Kummer et al, 2016]
49	Associate data objects consistently	The designer should associate data objects only to activities. In particular the designer should not associate a data object with a sequence flow if the sequence flow is connected to a gateway. The designer should always model the association with a direction. **Sources**: [White, 2008]

Continued on next page

ID	Name	Description
50	Keep a standard format	The designer should keep a unique format along diagrams and focus on a clean and friendly look and feel. Using different font sizes, colors, boxes sizes or overlapping labels might make the diagrams reading a challenge. The designer should not model further properties with different colors, in order to make diagrams recognizable. **Sources**: [Bosshart et al, 2014; Signavio, 2014; Leopold et al, 2015; Kummer et al, 2016]

Table 9: Business Process Modeling Understandability Guidelines.

B Business Process Model Metrics

Here we list all the metrics related to BPMN models that we collected from the literature, reporting their Name, Description, Source (the publications that present them) and Year (the year of the publication) (Tables 10, 11, and 12). For major details please refer to Corradini et al [2015].

NAME	DESCRIPTION	SOURCE	YEAR
NT	Number of tasks.	Rolón et al [2006]	2006
NCD	Number of complex decision.	Rolón et al [2006]	2006
NDOin/ NDOout	Number of data objects which are input/outputs of activities.	Rolón et al [2006]	2006
NID	Number of inclusive decision.	Rolón et al [2006]	2006
NEDDB	Number of exclusive data-based decision.	Rolón et al [2006]	2006
NEDEB	Number of exclusive event-based decision.	Rolón et al [2006]	2006
NL	Number of lanes.	Rolón et al [2006]	2006
NMF	Number of message flows.	Rolón et al [2006]	2006
NP	Number of pools.	Rolón et al [2006]	2006
NPF	Number of parallel forking.	Rolón et al [2006]	2006
NSFA	Number of sequence flows between activities.	Rolón et al [2006]	2006
NSFE	Number of sequence flows from events.	Rolón et al [2006]	2006
NSFG	Number of sequence flows from gateways.	Rolón et al [2006]	2006
CLA	Connectivity level between activities. Total Number of Activities / Number of Sequence Flows between these Activities. $CLA = TNA/NSFA$	Rolón et al [2006]	2006
CLP	Connectivity level between participants. $CLP = NMF/NP$	Rolón et al [2006]	2006
PDOPin/ PDOPout	Proportion of data objects as incoming/outgoing products and total data objects. $PDOPIn = NDOIn/TNDO$; $PDOPOut = NDOOut/TNDO$	Rolón et al [2006]	2006
TNT	Total number of tasks. $TNT = NT + NTL + NTMI + NTC$	Rolón et al [2006]	2006
PDOTout	Proportion of data objects as outgoing product of activities of the model. $PDOTOut = NDOOut/TNT$	Rolón et al [2006]	2006
PLT	Proportion of pools/lanes and activities $PLT = NL/TNT$	Rolón et al [2006]	2006
TNCS	Total number of collapsed sub-processes. $TNCS = NCS + NCSL + NCSMI + NCSC + NCSA$	Rolón et al [2006]	2006
TNA	Total number of activities. $TNA = TNT + TNCS$	Rolón et al [2006]	2006
TNDO	Total number of data objects in the model. $TNDO = NDOIn + NDOOut$	Rolón et al [2006]	2006
TNG	Total number of gateways. $TNG = NEDDB + NEDEB + NID + NCD + NPF$	Rolón et al [2006]	2006
TNEE	Total number of end events. $TNEE = NENE + NEMsE + NEEE + NECaE + NECoE + NELE + NEMuE + NETE$	Rolón et al [2006]	2006
TNIE	Total number of intermediate events. $TNIE = NINE + NITE + NIMsE + NIEE + NICaE + NICoE + NIRE + NILE + NIMuE$	Rolón et al [2006]	2006
TNSE	Total number of start events. $TNSE = NSNE + NSTE + NSMsE + NSRE + NSLE + NSMuE$	Rolón et al [2006]	2006
TNE	Total number of events. $TNE = TNSE + TNIE + TNEE$	Rolón et al [2006]	2006
CFC	Control-flow Complexity metric. It captures a weighted sum of all connectors that are used in a process model.	Cardoso [2007]	2005
NOA	Number of activities in a process.	Cardoso et al [2006]	2006
NOAC	Number of activities and control-flow elements in a process.	Cardoso et al [2006]	2006
NOAJS	Number of activities, joins, and splits in a process.	Cardoso et al [2006]	2006

Table 10: Business Process Model Complexity Metrics - Part 1.

NAME	DESCRIPTION	SOURCE	YEAR
HPC_D	Hasted-based Process Complexity (process difficulty).	Cardoso et al [2006]	2006
HPC_N	Hasted-based Process Complexity (process length).	Cardoso et al [2006]	2006
HPC_V	Halsted-based Process Complexity (process volume).	Cardoso et al [2006]	2006
NoI or Fan-in	Number of activity inputs. The fan-in of a procedure A is the number of local flows into procedure A plus the number of data structures from which procedure A retrieves information.	Cardoso et al [2006]	2006
NoO or Fan-out	Number of activity outputs. The fan-out of a procedure A is the number of local flows from procedure A plus the number of data structures which procedure A updates.	Cardoso et al [2006]	2006
Length	Activity length. The length is 1 if the activity is a black box; if it is a white box, the length can be calculated using traditional software engineering metrics that have been previously presented, namely the LOC (line of code) and MCC (McCabe's cyclomatic complexity).	Cardoso et al [2006]	2006
IC	Interface complexity of an activity metric. $IC = Length * (NoI * NoO)^2$, where the length of the activity can be calculated using traditional Software Engineering metrics such as LOC (1 if the activity source code is unknown) and NoI and NoO are the number of inputs and outputs.	Cardoso et al [2006]	2006
NOF	Number of control flow connections (number of arcs).	Cardoso et al [2006]	2006
TNSF	Total number of sequence flows.	Rolón et al [2009]	2009
CC	Cross-connectivity metric. It is the ratio of the total number of arcs in a process model to the total number of its nodes.	Vanderfeesten et al [2008]	2008
ICP	Imported Coupling of a Process metric. It counts, for each (sub-) process, the number of message/sequence flows sent by either the tasks of the (sub-) process or the (sub-) process itself.	Khlif et al [2009]	2009
ECP	Exported Coupling of a Process metric. It counts, for each (sub-) process, the number of message/sequence flows received by either the tasks of the (sub-) process or the (sub-) process itself.	Khlif et al [2009]	2009
W	Cognitive Weight. It measures the cognitive effort to understand a model; it can indicate that a model should be re-designed	Gruhn and Laue [2006]	2006
MaxND	Maximum Nesting Depth, where the nesting depth of an action is the number of decisions in the control flow that are necessary to perform this action.	Gruhn and Laue [2006]	2006
(Anti)Patterns for BPM	It counts the usage of anti-patterns. In a BP model, it can help to detect poor modeling.	Gruhn and Laue [2006]	2006
CP	Coupling metric. The metric calculates the degree of coupling. Coupling is related to the number of interconnections among the tasks of a process model. The higher coupling value of the process, the more difficult it is to change the process and the higher probability that there will be errors in the process.	Reijers and Vanderfeesten [2004]	2004
Cohesion	Cohesion measures the coherence within the parts of the model.	Reijers and Vanderfeesten [2004]	2004
CNC	Coefficient of Network Complexity or Connectivity coefficient. It is the ratio of total number of arcs in a process model to its total number of nodes. It is calculated as: $CNC = NOF/NOAJS$.	Latva-Koivisto [2001]	2001
MeanND	Mean Nesting Depth, where the nesting depth of an action is the number of decisions in the control flow that are necessary to perform this action.	Gruhn and Laue [2006]	2006
CI	Complexity Index , or reduction complexity. It is defined as the minimal number of node reductions that reduces the graph to a single node.	Latva-Koivisto [2001]	2001
RT	Restrictiveness Estimator. It is an estimator for the number of feasible sequences in a graph. RT requires the reachability matrix rij, i.e. the transitive closure of the adjacency matrix, to be calculated. $RT = \frac{2\sum r_{ij} - 6(N-1)}{(N-2)(N-3)}$	Latva-Koivisto [2001]	2001

Table 11: Business Process Model Complexity Metrics - Part 2.

NAME	DESCRIPTION	SOURCE	YEAR
S_N	Number of nodes. It is the number of activities and routing elements in a process model.	Mendling [2008]	2008
$\Pi(G))$	Separability. It is the ratio of the number of cut-vertices divided by the total number of nodes in the process model.	Mendling [2008]	2008
$\Xi(G)$	Sequentiality. It is the degree to which the model is constructed out of pure sequences of tasks. The sequentiality ratio is the number of arcs between none-connector nodes divided by the number of arcs.	Mendling [2008]	2008
diam	Diameter. It is the length of the longest path from a start node to an end node.	Mendling [2008]	2008
$\wedge$	Depth. It is the maximum nesting of structured blocks in a process model.	Mendling [2008]	2008
GM or MM	Gateway Mismatch or Connector Mismatch. It is the sum of gateway pairs that do not match with each other, e.g. when an AND-split is followed up by an OR-join.	Mendling [2008]	2008
GH or CH	Gateway Heterogeneity or Connector Heterogeneity. It defines the extent to which different types of connectors are used in a process model.	Mendling [2008]	2008
Φ	Structuredness. It relates to how far a process model can be built by nesting blocks of matching join and split connectors. The degree of structuredness can be determined by applying reduction rules and comparing the size of the reduced model to the original size.	Mendling [2008]	2008
CYC	Cyclicity. It captures the number of nodes in a cycle and relates it to the total number of nodes	Mendling [2008]	2008
TS or Concurrency	Token Splits or Concurrency. It captures the maximum number of paths in a process model that may be concurrently activate due to AND-splits and OR-splits; it sums up the output-degree of AND-joins and OR- joins minus one.	Mendling [2008]	2008
$\Delta(G)$	Density. It is the ratio of the total number of arcs in a process model to the theoretically maximum number of arcs.	Mendling [2008]	2008
ACD or AGD	Average Connector Degree or Average Gateway Degree. It is the average of the number of both incoming and outgoing arcs of the gateway nodes in the process model.	Mendling et al [2007b]	2007
MCD or MGD	Maximum Degree of a Connector or Maximum Gateway Degree. It is the maximum sum of incoming and outgoing arcs of these gateway nodes.	Mendling et al [2007b]	2007
ECaM	Extended Cardoso Metric. It is a Petri net version of metric that generalizes and improves the original CFC metric proposed by Cardoso. It focuses on the syntax of the model and ignores the complexity of the behavior.	Lassen and van der Aalst [2009]	2009
ECyM	Extended Cyclomatic Metric. It is directly adapted from McCabe Cyclomatic. It focuses on the resulting behavior and ignore the complexity of the model.	Lassen and van der Aalst [2009]	2009
SM	Structuredness Metric. It recognizes different kinds of structures in the process model and scores each structure by giving it some "penalty" value. The sum of these values is the Structuredness Metric (SM).	Lassen and van der Aalst [2009]	2009
DSM	Durfee Square Metric. It is based on h-index. It equals d if there are d types of elements which occur at least d times in the model (each), and the other types occur no more than d times (each)	Kluza and Nalepa [2012]	2012
PSM	Perfect Square Metric. It is based on the g-index. Given a set of element types ranked in decreasing order of the number of their instances, the PSM is the (unique) largest number such that the top p types occur (together) at least p^2 times.	Kluza and Nalepa [2012]	2012
Layout complexity	It evaluates the usability of different screen designs based on the Shannon formula.	Sears [1993]	1993
Layout appropriateness	It is the efficiency of a screen in terms of cost involved in completing a collection of tasks.	Comber and Maltby [1996]	1996
Layout measure	It is a group of measures that quantify layout of models: number of edge crossing, number of non-rectilinear edges, overlapping area, etc.	Eichelberger and Schmid [2009]	2009

Table 12: Business Process Model Complexity Metrics - Part 3.