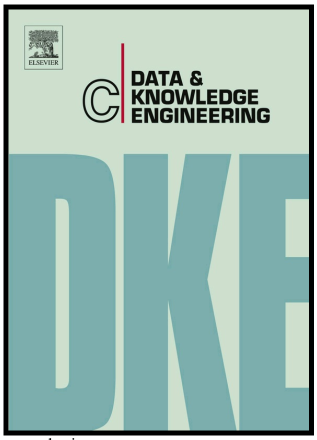
Author's Accepted Manuscript



www.elsevier.com

A Guidelines Framework for Understandable BPMN Models

Flavio Corradini, Alessio Ferrari, Fabrizio Fornari, Stefania Gnesi, Andrea Polini, Barbara Re, Giorgio O. Spagnolo

PII: S0169-023X(16)30341-X
DOI: https://doi.org/10.1016/j.datak.2017.11.003
Reference: DATAK1624

To appear in: *Data & Knowledge Engineering*

Received date: 28 November 2016
Revised date: 25 October 2017
Accepted date: 24 November 2017

Cite this article as: Flavio Corradini, Alessio Ferrari, Fabrizio Fornari, Stefania Gnesi, Andrea Polini, Barbara Re and Giorgio O. Spagnolo, A Guidelines Framework for Understandable BPMN Models, *Data & Knowledge Engineering*, https://doi.org/10.1016/j.datak.2017.11.003

A Guidelines Framework for Understandable BPMN Models

Flavio Corradini[1], Alessio Ferrari[2], Fabrizio Fornari[1], Stefania Gnesi[2], Andrea Polini[1], Barbara Re*[1], and Giorgio O. Spagnolo[2]

[1]University of Camerino - Camerino, Italy - {name.surname}@unicam.it
[2]ISTI-CNR, Pisa, Italy - {name.surname}@isti.cnr.it

Abstract

Business process modeling allows abstracting and reasoning on how work is structured within complex organizations. Business process models represent blueprints that can serve different purposes for a variety of stakeholders. For example, business analysts can use these models to better understand how the organization works; employees playing a role in the process can use them to learn the tasks that they are supposed to perform; software analysts/developers can refer to the models to understand the system-as-is before designing the system-to-be. Given the variety of stakeholders that need to interpret these models, and considering the pivotal function that models play within organizations, *understandability* becomes a fundamental quality that need to be taken into particular account by modelers. In this paper we provide a set of fifty guidelines that can help modelers to improve the understandability of their models. The work focuses on the Business Process Modelling Notation 2.0 standard published by the Object Management Group, which has acquired a clear predominance among the modeling notations for business processes. Guidelines were derived by means of a thoughtful literature review – which allowed identifying around one hundred guidelines – and through successive activities of synthesis and homogenization. In addition, we implemented a freely available open source tool, named BEBoP (understandaBility vErifier for Business Process models), to check the adherence of a model to the guidelines. Finally, guidelines violation has been checked with BEBoP on a dataset of 11,294 models available in a publicly accessible repository. Our tests show that, although the majority of the guidelines are respected by the models, some guidelines, which are recognized as fundamental by the literature, are frequently violated.

Keywords — Models Understandability; Business Process Modeling; BPMN; Modeling Guidelines; Model Quality; Tool.

1 Introduction

Graphical notations are often used to enhance textual or verbal communication, providing stakeholders with the possibility to actually *see* the subject of the discussion. This is particularly true within complex organizations, in which graphical notations can be used to represent Business Processes (BP) and hence visualize and reason about work practices. A BP consists of "activities that take one or more kinds of inputs creating an output, and that are performed in coordination in an organizational and technical environment" [Weske, 2012]. In addition, Business Process Management (BPM) supports stakeholders by providing methods, techniques and software to model, implement, execute and optimize work practices [Jeston and Nelis, 2014].

The literature shows that BP modeling has been identified as an important phase in BPM [Kalpic and Bernus, 2002], and the benefits of its use in practice are well recognized [Indulska et al, 2009]. At the same time the quality of the models resulting from the modeling phase is critical for the success of an organizations [Moreno de Oca et al, 2015]. In particular, the designed models must fit with the reality, and they must be considered *understandable* by all the stakeholders interested in the information they convey.

*Corresponding Author.

Understandability is certainly a complex non-functional quality, recognized as one of the most significant quality characteristics in the Business Process modeling community [Sánchez González et al, 2010] [Figl, 2017]. It depends on the context in which the models is "produced/consumed", and, not surprisingly, is affected by many factors, including suitable usage and arrangement of the notation elements [Genon et al, 2011]. Previous experiences and best practices can be fruitfully exploited in order to define guidelines that generally lead to the definition of more understandable BP models [Reijers et al, 2015]. It will be then the duty of the modeler to decide which guidelines to follow, and how much to take into consideration the other guidelines, or other non functional properties, in order to find the right balance. It is in fact generally the case that provided guidelines and other non functional properties can have negative correlations, that need to be mediated by the modeler.

Despite the availability of modeling guidelines, modelers can find in the literature only a few comprehensive understandability quality framework to which they can refer to improve their models [Sánchez-González et al, 2015] [Sánchez-González et al, 2013a]. Guidelines are in fact scattered among many different papers (e.g., [Silver, 2011; White, 2008; Mendling et al, 2010b]), which in general do not use a homogeneous template to describe them [Moreno de Oca and Snoeck, 2014]. The only exception in this sense is the work of Moreno de Oca and Snoeck [2014], which provides a first attempt towards a homogeneous guidelines framework[1]. Furthermore, even though commercial tools for checking a considerable number of guidelines are available (e.g., Signavio Process Editor[2]), no *open source* tool exists for this task. Given the need to adapt the guidelines to different contexts of model fruition [Genon et al, 2011], and reasonably assuming that the available guidelines are not exhaustive, having an open source software that can be tailored and extended by the BP community is particularly appealing.

The goal of this paper is to provide an homogeneous set of understandability guidelines, and to provide an open source tool to check their violation. To this end, we conducted a thoughtful review of existing guidelines to create a single and homogeneous reference framework that can be easily accessed and used by modelers. Furthermore, any time this was considered meaningful, we proceeded with the definition of suitable metrics and numeric thresholds that allow the adherence to the guidelines to be assessed. Algorithms were then provided in order to concretely check if a guideline is respected or not by a model. Such algorithms are included in an open source tool named BEBoP (understandaBility vErifier for Business Process models). The tool is implemented as a Web service and it can be accessed by users through its Web interface, or can be easily interrogated by other third parts modeling tools. The availability of an automatic tool to perform the checks makes the adoption of the guidelines much easier and effective [Gassen et al, 2015], [Haisjackl et al, 2015]. Indeed, modelers can improve the understandability of their models in an incremental way, by means of an iterative trial-and-error learning process, without having to memorize each single guideline.

It is worth mentioning that different classes of graphical notations to describe BPs have been investigated and defined. Even though the need for understandability is valid for any graphical notation, in this work we focus on BP models defined using BPMN 2.0[3]. This is a widely used OMG standard for modeling BPs [OMG, 2011]. BPMN defines three different diagrams to permit the representation of cooperating organizations. These diagrams, named *choreography*, *composition*, and *collaboration* differs for the abstraction level, and the details they focus on. The results we report here are mainly related to the representation of collaboration diagrams. Notably, in this work we are not limited to any subset of the notation.

In summary, the contribution of our work is threefold.

- It provides 50 BPMN *understandability modeling guidelines*. We collected, synthesized and homogenized a set of guidelines taken from 89 sources available in literature.

- It provides *metrics* and *thresholds* for the guidelines. The resulting comprehensive quality framework permits to concretely identify "bad smelling" models.

[1] The differences between the framework of Moreno de Oca and Snoeck [2014] and the one presented in this work will be detailed in Section 7.

[2] `http://www.signavio.com/products/process-editor/`

[3] We use BPMN or BPMN 2.0 interchangeably to refer to version 2.0 of the notation (Release Date: January 2011).

- It provides an open source tool named BEBoP to verify most of the listed guidelines.The proposed tool is offered as a service and can be easily integrated in any modeling tool.

This work was performed in the context of the EU funded Learn PAd project[4]. The project objective concerned the development of a learning platform for Public Administrations (PAs). The core idea of the project is to leverage BPMN models to convey process knowledge to PA stakeholders. In the context of Learn PAd, we developed and extensively applied the presented guidelines as well as BEBoP. In the current paper, we also present an excerpt of a model from one of the Learn PAd case studies, and we show the applicability of our approach – i.e., guidelines and tool – using the model as a reference example. In order to showcase the potential usage of the tool for empirical studies on BPMN model quality, we applied BEBoP to a collection of more than ten thousand real-world BPMN models provided by the BPM Academic Initiative (http://bpmai.org/) [Kunze et al, 2012]. Our tests show that, although the majority of the models adheres to most of the guidelines, particularly *relevant* guidelines are frequently violated in the dataset. Not surprisingly, we show that, the larger the models, the higher the average number of violations.

The rest of the paper is organized as follows. Section 2 describes the methodology adopted to define the guidelines. Sections 3 describes the proposed BP modeling guidelines, while Section 4 introduces the technical details of BEBoP. Section 5 illustrates the results of the application of the tool on a large data set, and then Section 6 shows the application of the guidelines into practice on a small scenario. Section 7 presents relevant papers coming from the literature review, and compares BEBoP with other tools available in the market. Finally, Section 8 closes the paper with some conclusions and opportunities for future works.

2 Methodology

Here we describe the methodology that we followed to derive the proposed understandability modeling guidelines. In particular, our effort has been structured over four main activities:

1. *Literature review:* to collect relevant papers;
2. *Categorization:* to classify the identified papers and simplify the following activities;
3. *Collection and synthesis:* to extract the set of guidelines and to compare the various guidelines in order to find relations among them (e.g., duplicates, overlaps);
4. *Homogenization:* to provide a unified description for each guideline.

Literature Review. The literature review, carried out to collect modeling guidelines, metrics and thresholds, was performed according to the *snowballing method* described by Webster and Watson [2002]. In particular, in order to identify the set of relevant research papers to consider in the next phases, the method suggests to start from an initial set of papers manually identified according to criteria described below. Successively, further relevant papers can be identified proceeding backward and forward in time, using respectively the related works section when available (*backward snowballing*), and the "cited by" functionality provided by digital libraries[5] (*forward snowballing*).

In our case we identified the initial set of papers running a manual search on conference proceedings and journals that, to the best of our personal knowledge, have an high reputation within communities for which the definition of understandability guidelines for Business Processes can certainly be considered as a relevant topic. In particular, the search was conducted accessing to proceedings and journal table of contents, to retrieve all the titles for papers published by a relevant set of scientific conferences and journals that could be considered primary venues by researchers working on topics related to Information Systems, Enterprise Modeling, Business Process Management and similar subjects. In particular, we considered the following list of conferences:

[4] http://www.learnpad.eu/

[5] Google Scholar in our case – http://scholar.google.com

- Advanced Information Systems Engineering (CAiSE);
- Business Information Systems (BIS);
- Business Process Management (BPM);
- On the Move to Meaningful Internet Systems (OTM);
- Business Informatics (CBI);
- Research Challenges in Information Science (RCIS);
- Practice of Enterprise modeling (PoEM).

We also considered the following journals:

- Data & Knowledge Engineering;
- Decision Support Systems;
- Information and Software Technology;
- Information Systems;
- Journal of Visual Languages and Computing;
- Journal of Systems and Software;
- International Journal of Information Technology & Decision Making;
- Journal of Management Information Systems;
- Industrial Management and Data Systems;
- Business & Information Systems Engineering.

An additional criterion that we considered to shape the *initial* set of papers refers to the publication date. In particular, we initially considered only papers published after the release of the BPMN 2.0 specification, i.e., January 2011. Certainly we can assume that the subject was already studied for the previous versions of the modeling notations, nevertheless we considered that, if relevant, the references published before 2011 would emerge later as a result of the snowballing procedure.

All the titles retrieved from the mentioned sources in the defined time frame were carefully considered, and in case they looked minimally promising with respect to our objectives, we downloaded the corresponding full paper. Successively in order to include or not a paper in the initial set, we proceeded reading the abstract, and both the "introduction" and the "conclusions" sections. To include a paper in the initial set we considered its potential in providing relevant information with respect to the definition of understandability guidelines. It is worth mentioning that according to our objectives we also considered useful papers not directly defining guidelines but somehow useful to clarify and better shape the objectives and characteristics of an understandability quality framework. As a result, the initial set was composed of 20 papers that were then reviewed in their entirety. Those papers are reported in bold in Table 1.

As a second step, the snowballing method asks to run backward and forward snowballing. This step was limited neither to the selected conferences and journals, nor to the considered time frame. This allows having a comprehensive picture on related research. We checked the reference list of the 20 identified research papers looking for possible relevant studies not considered so far. In order to decide if a paper was relevant or not we proceeded again as described for the initial set, i.e. with successive readings of title, abstract, introduction and conclusion. As a result, we identified 69 further relevant works (43 from backward and 26 from forward) to be included in the relevant set. Backward and forward snowballing search was done in an iterative way also on the papers identified in the second step. We stopped after two iterations since we did not find additional relevant research papers. After the steps reported above, the considered set included 89 papers.

Categorization. After the literature review, we run a categorisation activity to simplify the subsequent steps of collection and synthesis, and homogenization. Each identified paper was carefully read by at least two authors, that proposed to cluster them according to the five following categories (see Table 1 for details), obviously in case of misalignment we opted for the inclusion of the paper in all the suggested categories. As the reader can notice from Table 1, several papers refer to more than one category.

- **General**. It includes papers presenting high-level guidelines that impact on different aspects of the overall BPMN modeling practice.
- **Notation**. It includes papers presenting guidelines on the usage of the BPMN syntax.
- **Labelling**. It includes papers presenting guidelines for the assignment of proper labels to BPMN elements.
- **Patterns**. It includes papers presenting guidelines that suggest a specific arrangement of BPMN elements.
- **Appearance**. It includes papers presenting guidelines for a clear presentation of BPMN models.

Moreover we considered an orthogonal category, ***Metrics***, in which we included papers that indicate metrics and thresholds to evaluate the adherence of a model to the guidelines. Since this is a cross-cutting category, impacting on the other defined categories, we prefer to separate it from the others. Finally, another separate category is ***Quality***, which collects papers that do not define guidelines, but address or touch the issue of model quality either through literature reviews, or though vision papers. In this category we included papers that where reviewed to acquire a broader perspective on model quality.

Collection and Synthesis. After having categorized the papers, we ran a collection and synthesis activity. The collection task consisted in the extraction of the guidelines from the different papers in the sets resulting from the first five categories mentioned above. This step was carried out again manually and again each paper was considered by at least two authors that together decided to include or not a guideline in the set. Clearly after this step the same guideline might have been assigned to more than one set. We used a Web-based shared spreadsheet to list all the guidelines, and to take some notes on their characteristics. The spreadsheet included information regarding:

- The source paper to permit the easy access to the origin;
- The title and description of the guideline;
- Metrics related to the guidelines, when available;
- A note field in which to add comments and observations.

In this way each one of the authors had the possibility to read and make comments on each guideline. After one week we ran a remote plenary meeting in which each guideline was discussed to find matches and relations with other guidelines. In particular, we removed duplicated guidelines, and we re-wrote those facing the same or similar issues even though stated in different ways. It is worth mentioning that the plenary meeting was spread over two different half-days given the number of guidelines to be considered. The decision was taken at unanimity, and for some guideline the discussion was rather long. After this synthesis task, we reduced the list of guidelines to a set of 50 elements from the initial 100. It is worth mentioning that in this activity we did not try to derive a possible relevance measure for the identified guidelines. Indeed, we considered that all of them can have a different relevance according to the domain in which the model is built, and it is up to the modeler to decide which guidelines are more relevant in his/her context.

Table 1: Papers classification.

General	**[Laue and Awad, 2011] [Gschwind et al, 2014] [Figl and Laue, 2015] [Figl et al, 2013b]** [Reijers et al, 2010] [Weber et al, 2011] [La Rosa et al, 2011] [Reijers and Mendling, 2008] [Mendling et al, 2008] [Gruhn and Laue, 2009] [Leopold et al, 2015] [Sánchez-González et al, 2011] [Reggio et al, 2011] [White, 2008] [Dumas et al, 2012] [Mendling et al, 2007a] [Johannsen et al, 2014] [Silingas and Mileviciene, 2011] [Mendling and Strembeck, 2008] [Dumas et al, 2013] [Purchase, 2002] [Mendling et al, 2010b] [Silver, 2011] [Reijers et al, 2011a] [Moreno de Oca and Snoeck, 2014] [Mendling et al, 2012a] [Sánchez-González et al, 2013b] [Koehler and Vanhatalo, 2007]
Notation	**[Laue and Awad, 2011] [Gschwind et al, 2014]** [Claes et al, 2012] [Weber et al, 2011] [La Rosa et al, 2011] [White, 2008] [Mendling et al, 2008] [Krogstie, 2012] [Gruhn and Laue, 2009] [Kluza et al, 2013] [Leopold et al, 2015] [Moreno de Oca and Snoeck, 2014] [Signavio, 2014] [Reggio et al, 2011] [Dumas et al, 2012] [Mendling et al, 2007a] [Silingas and Mileviciene, 2011] [Sánchez-González et al, 2012] [Bosshart et al, 2014] [Koehler and Vanhatalo, 2007] [Mendling et al, 2010b] [Silver, 2011] [Reijers et al, 2011a] [Mendling et al, 2012a] [Sánchez-González et al, 2013b] [Leopold et al, 2016]
Labeling	**[Mendling et al, 2010b][Leopold et al, 2013] [Mendling et al, 2010c] [Pittke et al, 2014]** **[Leopold et al, 2017] [Overhage et al, 2012]** [La Rosa et al, 2011] [Kluza et al, 2013] [White, 2008] [Leopold et al, 2012] [Silingas and Mileviciene, 2011] [Mendling et al, 2012a] [Mendling and Recker, 2008] [Pittke et al, 2013] [Leopold et al, 2015] [Moreno de Oca and Snoeck, 2014] [Mendling et al, 2010a] [Mendling and Reijers, 2008] [Leopold et al, 2010] [Mendling and Strembeck, 2008] [Silver, 2011] [Signavio, 2014] [Sánchez-González et al, 2013b] [Leopold et al, 2016] [Koschmider et al, 2015] [Pittke et al, 2016] [Weber et al, 2011]
Patterns	**[Dumas et al, 2012]** [White, 2008] [Barjis et al, 2010] [Johannsen et al, 2014] [Silingas and Mileviciene, 2011] [Mendling et al, 2008], [Purchase, 2002] [Koehler and Vanhatalo, 2007] [Silver, 2011] [Moreno de Oca and Snoeck, 2014], [Weber et al, 2011]
Appearence	**[Bernstein and Soffer, 2015] [Kummer et al, 2016]** [Weber et al, 2011] [La Rosa et al, 2011] [Figl and Strembeck, 2015] [Leopold et al, 2015] [Moreno de Oca and Snoeck, 2014] [Signavio, 2014] [Reggio et al, 2011] [White, 2008] [Reijers et al, 2011b] [Gschwind et al, 2014] [Purchase, 2002] [Bosshart et al, 2014] [Mendling et al, 2010b] [Silver, 2011]
Metrics	**[Sánchez-González et al, 2010] [Overhage et al, 2012]** [Purchase, 2002] [Sánchez-González et al, 2012] [Sánchez-González et al, 2013b] [Mendling et al, 2012a] [Rolón et al, 2006] [Mendling, 2008] [Sánchez-González et al, 2011] [Kluza and Nalepa, 2012] [Mendling et al, 2007a] [Muketha et al, 2010] [Reynoso et al, 2009] [Kluza et al, 2014] [Melcher et al, 2010][Melcher and Seese, 2008]
Quality	**[Gassen et al, 2015] [Haisjackl et al, 2015] [Moreno de Oca et al, 2015]** **[Figl et al, 2010] [Snoeck et al, 2015] [Sánchez-González et al, 2015]** [van der Aalst, 2000] [Sharp and McDermott, 2009] [Becker et al, 2000] [Krogstie et al, 2006] [Moody et al, 2002] [Vom Brocke et al, 2010] [Mendling et al, 2007b] [Mendling et al, 2012b] [Krogstie, 2012] [Houy et al, 2014] [Sánchez-González et al, 2013a] [Davis, 2001] [Heggset et al, 2014] [Krogstie, 2015] [Petrusel et al, 2016] [Figl, 2017] [Figl et al, 2013a]

Homogenization. In the last activity, we rewrote all the guidelines in a homogeneous way, using a standardized *template*. Results are presented briefly in Section 3 and more details are included in the companion technical report [Corradini et al, 2015]. In particular, during this activity – carried out with the support of the shared spreadsheet used before, and through remote meetings – we also included metrics and thresholds for the guidelines, according to the indications provided by papers listed in Table 1, row *Metrics*. For some of the guidelines (six in total, all related to notation aspects), metrics and thresholds were not defined in the literature, while they were required to automatically assess the guidelines, in light of the development of an automatic guideline verification tool. In these cases we introduced appropriate metrics and thresholds. These specific metrics and thresholds require appropriate validation, which we leave as part of the future work. A complete list of metrics available in the literature is reported in Appendix B. It is also worth mentioning that in some situation the resulting guidelines results to be composite, in the sense that they require to check and relate different aspects. In some case only part of the guideline can be measured, as it is for instance the case for guideline 11. Nevertheless we preferred to include composite metrics when we considered that strictly related aspects would have been easier to be taken into account by the modeler – see [Corradini et al, 2015] for further details.

It is worth mentioning that we deliberatively decided to not undertake an effort toward a Systematic

Literature Review (SLR) based approach [Kitchenham, 2007]. Certainly this would have been a valid alternative. On the other hand we considered some shortcomings of SLR that could have affected our results. One of the most tricky problem generally influencing SLRs refers to the delay with which papers are indexed by digital libraries. In short we would probably have missed some papers published in 2016, and most of the papers published in 2017, unless the SLR had included a snowballing step or a similar procedure. Indeed the initial selection foreseen by the snowballing method is needed to solve such a problem since the analysis can also be performed manually when a selected venue is still not indexed, directly accessing the web sites of journals and conferences to retrieve the list of papers. Another aspect we considered refers to the definition of the search string. Indeed the engineering of the query to use in order to retrieve relevant research works is probably the most complex and tricky aspect in carrying on an SLR. In a previous paper we did an SLR on Business Process Flexibility [Cognini et al, 2016] and we discovered that there were available already 8 SLRs on very similar aspects and that only 4 papers out of more than 500 were shared by all SLRs. For the purpose of this work we considered quite difficult to define a "safe" search string, given that the objective was quite articulated, and guidelines could have been reported in a research work using different terminologies. In addition we thought that also papers treating understandability at a more abstract level could have been relevant in order to derive general principles about the topic. In summary, we considered that a possible imprecision in the definition of the search query would have led to the definition of a too much broad search, that would have correspondingly provided too many results impacting on the quality of the results applying SRL.

3 Understandability Guidelines Framework

The *BP Modeling Understandability Guidelines* that we defined are recommendations that a designer should follow to model more understandable BPMN models. Clearly the final decision related to the application is always subjective and it is part of the modeler's responsibility, also in light of the existence of negatively correlated guidelines. As already noted we decided to not provide a ranking for the guidelines, the set we provide is then flat. This decision was mainly due to the observation that each guideline can assume a different relevance depending from the application domain and in our study we did not have clear evidences to permit the derivation of a partial order relation among guidelines.

Here we provide further details on the results of the categorisation, collection and synthesis, and homogenisation activities. For each guideline we provide a template including the following fields (***italic*** indicates optional fields).

- An **ID** to uniquely identify the guideline.
- A **Name** to help the BP model designer in easily remembering the guideline.
- A **Description** to provide an explanation of the guideline to the BP model designer.
- **References** to report the original guideline's source and additional papers contributing to it definition allowing the users to access additional materials about the guideline itself.
- ***Metrics and Thresholds*** to assess the adherence of a model to a guideline. This field is optional, since, for some guidelines – e.g., patterns guidelines (see guideline 41, Table 5) or labelling guidelines (guideline 34, Table 4) – metrics and thresholds would not make sense, or would be trivial.
- An ***Example*** to graphically show the application of the guideline to a practical scenario, and to highlight the differences between good and bad modeling. This field is optional, since for some guidelines – e.g., guidelines considering model size (see guideline 2, Table 2) – graphical examples would not be more helpful than the textual description.

Following the categorisation we did for the paper (see Section 2), we found useful to organize the guidelines into categories.

- **General**. It includes guidelines that impact on different aspects of the overall BPMN modeling practice (9 guidelines from ID 1 to ID 9).
- **Notation**. It includes guidelines on the usage of the BPMN syntax (16 guidelines from ID 10 to ID 25).
- **Labelling**. It includes guidelines for the assignment of proper labels to BPMN elements (14 guidelines from ID 26 to ID 39).
- **Patterns**. It includes guidelines that suggest a specific arrangement of BPMN elements (3 guidelines from ID 40 to ID 42).
- **Appearance**. It includes guidelines for a clear presentation of the BPMN model (8 guidelines from ID 43 to ID 50).

This categorisation, besides being useful for presentation purposes, helps a modeler to reason and to focus on specific aspects of the model that he/she wants to maintain understandable. For example, if a modeler is particularly interested in improving the understandability of labels associated with the BPMN elements, he/she can directly refer to the category "Labelling"; if he/she is interested in improving the overall model appearance he/she can refer to the category "Appearance", *etc.*

For the sake of space, we do not list all the guidelines in this section. The interested reader can refer to Appendix A for a complete list including the name, the description and the source of the guidelines, or to the technical report by Corradini et al [2015], in which the guidelines are presented in their entirety with all the fields of the template. In the following, we provide example guidelines for the identified categories. Furthermore, in Section 6 we report an application scenario that refers other relevant guidelines.

General An example of a guideline for the category "General" is reported in Table 2. The guideline has ID equal to 2 and it is named "Minimize model size". It recommends the designer to minimize the number of elements that appear in a model; indeed, large BP models are difficult to read and then they are difficult to be understand. This guideline is mentioned by Mendling et al [2010b]. We associated a metric to the guideline referring to the number of elements in the model, and a threshold of 31 elements. This value has been taken from Mendling et al [2012a], where it was demonstrated that models with more than 31 elements generally result in difficulties in understanding.

Table 2: ID 2 “Minimize model size”.

Guideline Name	Guideline ID
Minimize model size	2
Description	
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.	
References	
[Mendling et al, 2010b] [Mendling et al, 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]	
Associated Metrics and Thresholds	
$MinimizeModelSize(x) = \begin{cases} 0 & if \quad SN \leq 31; \\ 1 & \quad otherwhise. \end{cases}$ Where: - $x \in$ Nodes of BPMN Model; and - SN is the number of nodes: number of activities and routing elements in a process model.	

Notation An example of a guideline for the category “Notation” is reported in Table 3. The guideline has ID equal to 18, and it is named “Split and join flows consistently”. It recommends the designer not to use gateways to join and split flows at the same time; this practice leads to better transparency and clarity of the process. The guideline was first proposed by Mendling et al [2010b], and then reaffirmed by Bosshart et al [2014] and Signavio [2014]. However, none of the these provides specific metrics and associated thresholds to assess the guideline. Hence, we defined these attributes as follows: to check if the guideline is violated, we need to check the number of incoming and outgoing edges from a considered gateway; if the number of incoming edges and the number of the outgoing edges are both greater than one, then the guideline is violated.

We also show an example model violating the guideline (Bad Modeling) and an example model that meets the guideline (Good Modeling). The Bad model depicts a XOR exclusive gateway which is used both to join and split the flow; the Good model instead represents separately the join and the split of the flows by means of two different XOR exclusive gateways.

Notably, in this category we also introduced guideline 15 with not explicit sources that suggests to reduce the usage of terminate-end-event if possible. The guideline has been inspired by our experience as teachers of BPM in university courses, and the observation that students have difficulties in interpreting models containing such notation elements. We also introduce guideline 23 in relation to the modeling of message exchanges. Also in this case the guideline has been inspired by our experience as teachers.

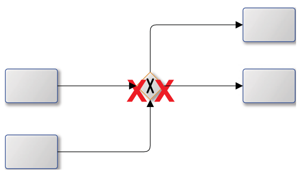
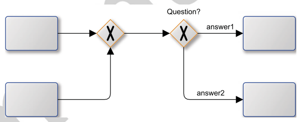
Table 3: ID 18 "Split and join flows".

Guideline Name	Guideline ID
Split and join flows consistently	18
Description	
The designer should not use any kind of gateways to join and split at the same time.	
References	
[Bosshart et al, 2014][Signavio, 2014][Mendling et al, 2010b][Mendling et al, 2012a] [Moreno de Oca and Snoeck, 2014][Gschwind et al, 2014][Krogstie, 2012][Silver, 2011]	
Associated Metrics and Thresholds	

$$splitJoinFlows(x) = \begin{cases} 0 \quad if \quad (Edges_{in}(x) = 1 \wedge Edges_{out}(x) > 1) \quad \vee \\ \quad (Edges_{in}(x) > 1 \wedge Edges_{out}(x) = 1) \\ 1 \quad otherwise \end{cases}$$

Where:
- $x \in$ Gateways; and
- $Edges_{in}(x)$ is the sum of incoming arcs of element x; and
- $Edges_{out}(x)$ is the sum of out coming arcs of element x.

Example

Bad Modeling	Good Modeling
	

Labelling An example of a guideline for the category "Labelling" is reported in Table 4. The guideline has ID equal to 34, and it is named "Labelling XOR Gateway". It suggests the designer to label XOR split gateways with an interrogative phrase and the outgoing sequence flows with a condition stated as outcome. The guideline is extracted from the work of Signavio [2014].

Notably, in this category we added guideline 39 with no source, since it directly descends from guideline 34 that suggests to include explicit conditions on XOR gateways. We considered useful to extend this rule to a notation element that is not represented as a gateway.

Table 4: ID 34 "Labeling guidelines".

Guideline Name	Guideline ID
Labeling XOR Gateway	34
Description	
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.	
References	
[Signavio, 2014][Leopold et al, 2015][Mendling et al, 2010c][Leopold et al, 2013] [Mendling and Strembeck, 2008][Leopold et al, 2013][Koschmider et al, 2015] [Leopold et al, 2010][Leopold et al, 2012][Pittke et al, 2013]	
Associated Metrics and Thresholds	
$labellingXorGateways(x) = \begin{cases} 0 & if \quad (getName(x)! = empty) \quad \wedge \\ & \quad (Diverging(x) = true) \\ 1 & otherwhise \end{cases}$ *where:* - $x \in$ Exclusive Gateways; and - $getName(x)$ returns the name of the Gateways; and - $Diverging(x)$ return true if gateway is diverging.	

Patterns An example of a guideline for the category "Patterns" is reported in Table 5. The guideline has ID equal to 41, and it is named "Use sub-processes". It recommends the designer to use sub-processes to group activities with the same purpose when: i) a set of consecutive activities has an owner different from the main process owner; ii) a set of consecutive activities has a different goal from the main process; iii) a process or a fragment must be re-used in another process. The usefulness of using sub-processes has also been described, among others (see Table 5), by Reijers et al [2010].

Table 5: ID 41 "Use sub-processes".

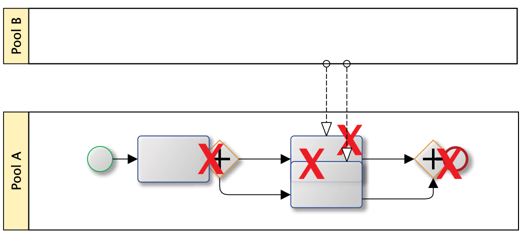
Guideline Name	Guideline ID
Use sub-processes	41
Description	
The designer should make use of sub-processes to group activities with the same purpose when: (i) a set of consecutive activities has an owner different from the main process owner; (ii) a set of consecutive activities has a different goal from the main process one; and (iii) a process or a fragment must be re-used in another process (use Call Activities in this case).	
References	
[Purchase, 2002][Silingas and Mileviciene, 2011][Moreno de Oca and Snoeck, 2014] [Johannsen et al, 2014]	

Appearance An example of a guideline for the category "Appearance" is reported in Table 6. The guideline has ID equal to 44, and it is named "Avoid overlapping elements". It tells the designer to avoid overlapping BPMN elements. This guideline appears in the works of Bosshart et al [2014] and Signavio [2014]. We also show an example of a model violating the guideline (Bad Modeling) and a model that meets the guideline (Good Modeling). The Bad model depicts two overlapping task elements and a message flow crossing both of them; the Good model instead shows a proper disposition of tasks and message flows. It is worth noticing that, in the Good model, we actually have two overlapping flows, which in principle violates the guideline. The apparently erroneous figure wants to suggest that the guideline should be considered as a *recommendation*, and the designer has to understand its spirit rather then following it without compromises. In some cases, as the one depicted, overlapping is not avoidable, and the compromise consists in minimizing the possible overlaps, while leaving overlapping elements when this does not disrupt the readability of the model.

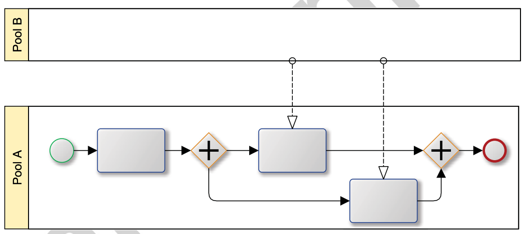
Table 6: ID 44 “Avoid overlapping elements”.

Guideline Name	Guideline ID
Avoid overlapping elements	44
Description	
The designer should avoid overlapping, or crossing, BPMN elements.	
References	
[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]	
Example	
Bad Modeling	**Good Modeling**

Bad Modeling



Good Modeling



4 Automatic Verification of the Guidelines: BEBoP

BEBoP (understandaBility vErifier for Business Process models) is a Java tool that supports BP designers in establishing whether their models comply with the understandability guidelines. The tool was developed as a Web service, and its basic graphical user interface can be accessed by the users through any Web browser[6]. The service can also be accessed by other software through its *RESTful* interface. Then it can be integrated as a plug-in in other existing tools, and eventually extended, for instance to permit the clustering of multiple invocations to check more than one model at the same time[7]. Currently, BEBoP is integrated within the Learn PAd Modeling Environment[8], which is a platform for designing BPMN models oriented to Public Administration stakeholders, for which understandability of models is a key quality aspect.

BEBoP reads a *.bpmn* file compliant with the OMG BPMN 2.0 standard, and produces a XML file that describes the guidelines that are not met and the BPMN elements violating them. Then, the XML file is loaded by the provided graphical user interface, which visualizes the violations of the guidelines in the model.

The tool allows to automatically verify 34 of the 50 guidelines. These guidelines are the ones that have associated metric and thresholds, or refer to the presence/absence of BPMN elements and their associated labels. In Table 8 we report the specific guidelines that are verified or not by the tool, while specific examples in the two classes are given in Sect. 6. Notably among the 34 measurable guidelines 16 are composite, and for 12 of them the associated metric refers only to a partial aspect (among these 10 are related to labeling).

[6] `http://pros.unicam.it/bebop-webinterface/`

[7] Source code available at: `https://PROSLab@bitbucket.org/PROSLab/bebop.git`, you can clone it using git

[8] Learn PAd modeling environment, available at: `https://goo.gl/cSgCzU`.

For each guideline, the tool implements an algorithm for its verification. Each guideline applies to specific model elements. Therefore, the implemented algorithm navigates the model elements that are relevant for the guideline and checks whether the elements comply to the guideline.

The results of the analysis are shown in Fig. 1. In particular, the BEBoP user can see the violated guidelines both in textual (left panel) and in visual form (right panel). The visualization of the model is enriched with red highlights on the elements that violate some guidelines[9]. On the left panel the complete list of guidelines is presented. Violated guidelines are marked in red (dark gray in B/W). Respected guidelines are marked in green (light gray in B/W). In addition the total number of satisfied guidelines is reported on the top bar. Fig. 2 shows a zoom on the left panel from Fig. 1. In particular, it is possible to inspect all the guidelines to read their descriptions, and, in case of defects, the list of elements that do not respect the guidelines. Overall, this *multi-modal* visualization allows the modeler to easily understand guidelines violations. On the one hand, seeing the defective elements directly on the model gives an immediate indication of the elements that need to be changed. When many guidelines are violated, the model might appear covered with red highlights, and single violations might not be immediately understandable from the graphical BP. In these cases, looking at the textual list of violated guidelines helps the modeler in going through each single violation, and in applying appropriate changes to the model.

To conclude the developed tool can be used by modelers to identify possible issues at first. Once the analysis reports guideline violations the modeler will have to use his/her expertise to produce a new version of the model that satisfies the properties, still respecting process objectives. The quality check should then be run again, and the loop will end once the modeler considers that the possibly remaining violations are not relevant.

5 Application of BEBoP to a Large Collection of BPMN Models

In this section, we apply BEBoP to a large collection of real-world BPMN models provided by the BPM Academic Initiative (http://bpmai.org/) [Kunze et al, 2012]. Our goal is to showcase the potential of our tool to support empirical research on the quality of BPMN models. In addition, we want to see which guidelines are frequently violated in practice, and which guidelines can be regarded as already established within the BPMN modeling community. The set-up and the results of the performed tests are described below.

5.1 Test Set-Up

The BPM Academic Initiative repository consists of 88 103 BPMN models including 86 357 collaborations. However, we restricted to the latest revision of the models with 100% connectedness (this measure refers to the size of the largest connected sub-graph against the size of the overall model). A model without this level of connectedness includes disconnected fragments, which typically means that the model has not been finalized. Including such models in our evaluation would have resulted in verification data that would be difficult to interpret.

This gave us a dataset of 16 032 models with a reasonable degree of quality. From these models we selected 11 294 models including more than 5 BPMN elements, to exclude toy examples from our evaluation. Considering our reference dataset, we performed a preliminary transformation step from .json (the repository format) to .bpmn (the format we manage), and then run the guidelines check with BEBoP.

5.2 Test Results

General remarks From the data collected after the runs of BEBoP we could observe that all the models violate at least one guideline, and indeed the most correct model violate 5 guidelines, while in average around 8 guidelines are violated by BPMN models. As we have already said violations cannot be considered errors by default. Indeed they could be explicitly related to decisions made by

[9]In the picture, dashed circles surround defective elements to ease visualization when this figure is printed in B/W

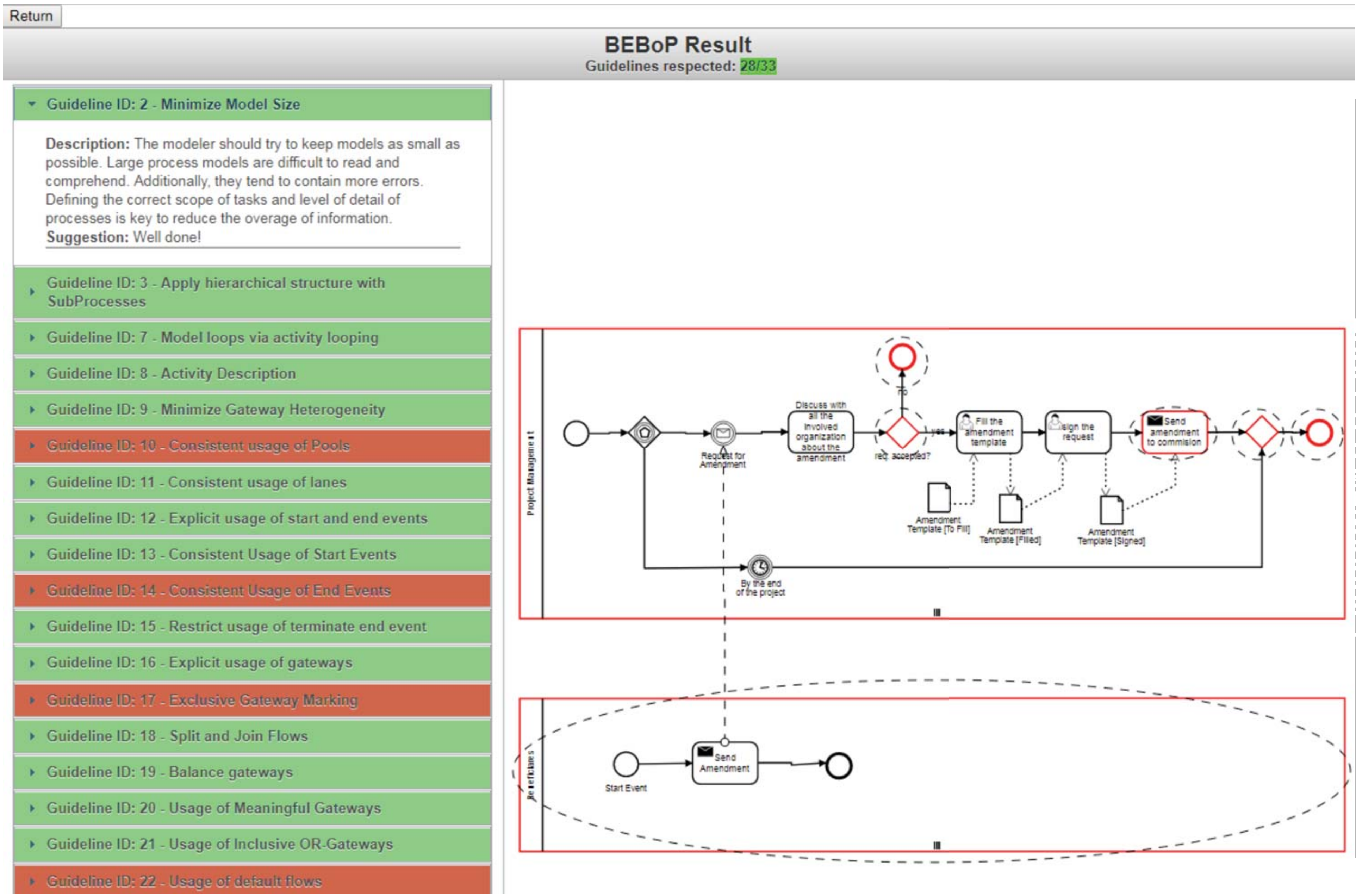


Figure 1: Screenshot of the results shown by BEBoP.

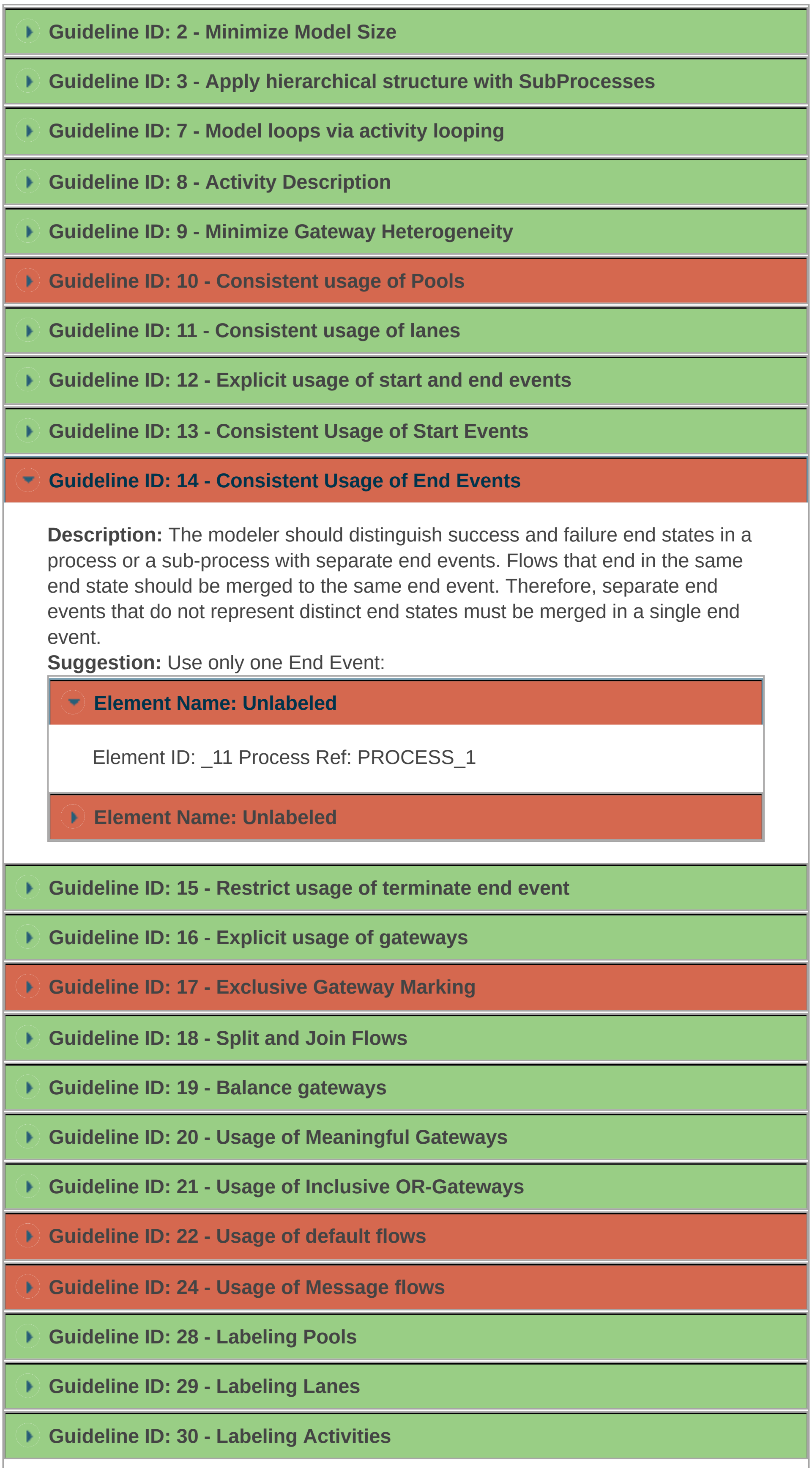


Figure 2: Zoom on the results of the guidelines verification component.

the modelers. On the other hand, in relation to the data we further discuss below, we can reasonably suspect that the number of violations would decrease if models could be automatically analyzed, and warning returned to the modelers. It is difficult to derive a ranking on the relevance of the guidelines from the observed violations. Indeed some of the guidelines are of interest only for a reduced number of models, for instance those on aspects related to communication that are relevant only for collaboration diagrams. Nevertheless we could observe the following data in relation to the guidelines related to quite general aspects that somehow could have affected any model.

Interestingly the guideline with the highest percentage of violation is **Provide activity description** (ID 8, 96.66%), which requires activities to be associated with natural language descriptions, followed by **Use default flows** (ID 22, 72.48%), **Labeling XOR Gateway** (ID 34, 50.72%), **Use explicit gateways** (ID 16, 48.05%), **Labeling converging gateways** (ID 36, 46.41%), **Labeling lanes** (ID 29, 41.15%). These are all guidelines that need to be further disseminated in the BPMN modeling practice, given the large amount of models that violate them. Some of these guidelines have little space in the literature, as, e.g., ID 22 introduced by White [2008], and ID 8, introduced in our work[10]. On the other hand, some guidelines that are largely mentioned in the literature are not widely adopted in practice. Representative examples are ID 34, cited by Mendling et al [2010c]; Leopold et al [2013], and others, and ID 16, referred, e.g., by Bosshart et al [2014]; Signavio [2014]. It is worth noticing that ID 16 is also one of the most *relevant* guidelines, since it is part of the 7 *fundamental* process modeling guidelines presented by Mendling et al [2010b].

Overall, we can say that, although the largest part of guidelines appear to be satisfied in the considered models, there is still a non negligible amount of them – including particularly relevant ones – that BPMN modelers should start to consider to improve the understandability of their models.

Impact of Model Size on Understandability Table 7 provides insights on the impact that the *size* of the models have on the violation of the guidelines. In particular, we classify the models in terms of the number of BPMN elements they contain (column *Size Class*), we report the number of models in each class (*Models*), and we provide the average number of elements for each model (column *AVG Elements*). For each guideline, we computed the percentage of models, within the specific *Size Class*, that violates it. In the computation, we considered only whether a guideline was violated or not by a model, and not how many times it was violated by the same model. The column *Guidelines* reports the average of these percentages. This gives an aggregate indicator of the degree of violation of the guidelines by the models belonging to a certain *Size Class*. Furthermore, we also computed the average of these percentages within each specific guideline category, namely *General*, *Notation*, *Labeling*, and *Appearance* – the *Patterns* category is not included since the guidelines in this category are not automatically verified by BEBoP. We report these values in the rightmost columns of the table. The last row of the table, Total (5-150), reports the same values defined for the other rows, but considering all the models, instead of partitioning them by *Size Class*.

Overall, we see that, when the size of the model increases, the percentage of guidelines that are violated increases consequently. Hence, as one might expect, the model size has a relevant impact on the degree of understandability of the model, and the larger the model, the higher the number of guidelines that tend to be violated. This is true both at the aggregate level (column *Guidelines*), and considering single categories of guidelines. The *Appearance* category is the one with the lowest percentage of violations (8,01%). However, also this category drastically increases from 8,01% to 34,31% (an increase of 26.03%) when the *Size Class* passes from 5-10 elements to 101-150 elements. The highest percentage of violations is observed for the *General* category, which reaches 57.06% for models of 101-150 elements. On the one hand, this is influenced by guideline 2 – belonging to the *General* category –, which, in our implementation, is violated when the model includes more than 31 elements. On the other hand, we observe an increase in the percentage of models violating guidelines also in categories other than *General*. This is a confirmation of the utility of reducing the size of the models to improve their overall quality, and, in particular, their understandability. Not surprisingly our data indicates that, when the size grows, also understandability defects tend to increase. Indeed

[10]Besides the limited literature support, guideline ID 8 may have also limited *tool* support, since some of the modeling environment do not allow to complement activities with descriptions.

the identification of model characteristics resulting to be correlated to model understandability could be an interesting aspect to explore. The dimension of the data set we used could enable such a study, providing indications on the validity or not of possibly identified correlations. For instance, it could be interesting to evaluate if a given complexity measure has an impact on understandability. We leave such a study as future work.

Size Class	*Models*	*AVG Elements*	*Guidelines*	*General*	*Notation*	*Labeling*	*Appearance*
5–15	6 604	10	13,22%	19,01%	11,57%	14,10%	8,01%
16–25	2 683	20	19,64%	19,96%	19,51%	20,89%	15,12%
26–45	1 432	33	25,62%	33,41%	24,23%	23,82%	25,72%
46–100	541	63	33,48%	53,97%	30,55%	27,26%	35,80%
101–150	34	114	37,61%	57,06%	35,92%	31,82%	34,31%
Total (5-150)	11294	18	17,36%	22,85%	16.04%	17,63%	13,36%

Table 7: Violation of the guidelines for increasing sizes of the models.

6 Understandability Guidelines in Practice

The work presented in this paper was carried out in the context of the Learn PAd project. The project involves an innovative holistic e-learning platform that aims to enhance, in a Public Administration (PA) context, the civil servants learning experience through the use of BP models. This platform enables process-driven learning and fosters cooperation and knowledge-sharing. Within the Learn PAd project BEBoP and the guidelines were used to improve the understandability of BPMN processes of PAs, and the approach was extensively applied in several Learn PAd scenarios. Here, we report the application of the guidelines to the European Project Budget Reporting scenario (EPBR). EPBR refers to a set of process performed by PAs that received a research grant from the European Union. For presentation purposes, we focus on the *Manage Amendment* BP model (a part or the EPBR), and we discuss the procedure we used to verify which understandability guidelines were met, or not. Then we describe how a modeler could eventually re-design the model by applying BP modeling guidelines. The considered scenario serves to clarify the process that we think can be applied to revise and improve models, in accordance to the guidelines.

6.1 Guidelines Application on the Manage Amendment Model

The *Manage Amendment* process is triggered when one or more partners of the EU-funded project (called beneficiaries) request an amendment to the project coordinator. Amendments are used to modify the grant agreement and its annexes. They can be requested at any time during the lifetime of the project. After the request, all the partners discuss the amendment; if all the partners agree with it, the request is sent to the EU Commission. In Fig. 3 we show the main steps, and involved documents, which are described in the following.

- *Request for Amendment.* The coordinator receives a request for amendment from one of the Beneficiaries (*Send Amendment* activity).
- *Discuss with all the Involved Organizations about the Amendment.* All the partners of the project discuss about the proposed amendment.
- *Fill the Amendment Template.* The coordinator fills the amendment template provided by the EU. The data object "Amendment Template" in the state "To Fill" is requested to perform this activity; it changes its state in "Filled" as soon as the activity is performed.

- *Sign the Request.* The coordinator signs the amendment request. The data object "Amendment Template" in the state "Filled" is requested to perform this activity and it changes its state in "Signed" as soon as the activity is performed.

- *Send Amendment to Commission.* The coordinator sends the amendment request to the EU. The data object "Amendment Template" in the state "Signed" is requested to perform this activity.

To verify the model we applied the following process, which can be regarded as the reference approach to follow when using BEBoP, and the guidelines in general. To identify which guidelines were met, we first scanned the model with BEBoP, and we collected its responses. Then, we proceeded by redesigning the model parts which were in conflict with the guidelines we consider relevant. Then the model was validated again using BEBoP, to check that the modifications did not introduce other undesired violations. Finally, we manually checked whether the guidelines that are not checked by BEBoP – 16 in total – were violated or not. In case of a violation, we proceeded with further adjustments.

Here, we discuss the model *before* and *after* the redesign (i.e., after the final adjustments), and we do not distinguish between automatic and manual verification steps. Instead, in the text of the following sections, in which we describe guidelines violations and adjustments, we will highlight the names of automatically verified guidelines in **bold**, and manually verified guidelines in ***italic***.

Fig. 3 shows the model before redesign. Dashed rectangles are used to highlight the parts of the model that did not comply with the guidelines. The number displayed corresponds to the ID of the guideline that resulted being violated. Fig. 4 reports the result of the model redesign. Ellipses highlight the parts that we modified to comply with the guidelines. The number displayed corresponds to the ID of the guideline whose violation is mitigated by the redesign.

Notably, guideline Balanced Gateways (ID 19) is violated both by the initial model, and by the refactored one. Indeed this is a simple case of contrasting guidelines in which to keep the design simple the modeler considers more important to satisfy guideline Use end events consistently (ID 14) on the consistent usage of end event, sacrifying then guideline Balanced Gateways (ID 19).

6.1.1 XOR Gateways

One of the detected guidelines violations involves XOR gateways that are represented both with and without marker; the representation of the same element in two different ways causes confusion to the model user. This goes against guideline **Mark exclusive gateways** (ID 17) which tells to the designer to make use of only Exclusive Gateways with the "X" marker. In addition, the guideline **Use default flows** (ID 22) tells the designer to associate a default flow with each XOR and OR gateway. Also this guideline was violated since no default flow was defined for all the XOR gateways. In general, this can cause issues since the reader does not get a clear understanding of which is the default BP flow, i.e., the flow that is executed when everything goes as expected, without exceptions or errors. To solve the described guidelines violations, we enforced the use of the "X" marker on each XOR gateway, and we assigned a default flow for the XOR gateway "Amendment request accepted?".

6.1.2 Pools and Message Flows

Proceeding with the model re-design, we focus on resolving guideline violations concerning the pool named "Beneficiaries" and the send task named "Send Amendment to Commission". The former is placed in an open pool, and violates guideline **Use pools consistently** (ID 10), which requires *to use a black-box pool to represent external participant/processes.* Hence, in Fig. 4, we substituted the pool with a black-box pool. Instead, the send task called "Send Amendment to Commission" is supposed to be associated with message flow towards the European Commission, but the message flow is not displayed. This violates guideline **Use message flows** (ID 24) which recommends to *represent message flows for each send or receive task.* To address this issue, a new pool is introduced, to represent the European Commission, and a message flow is added.

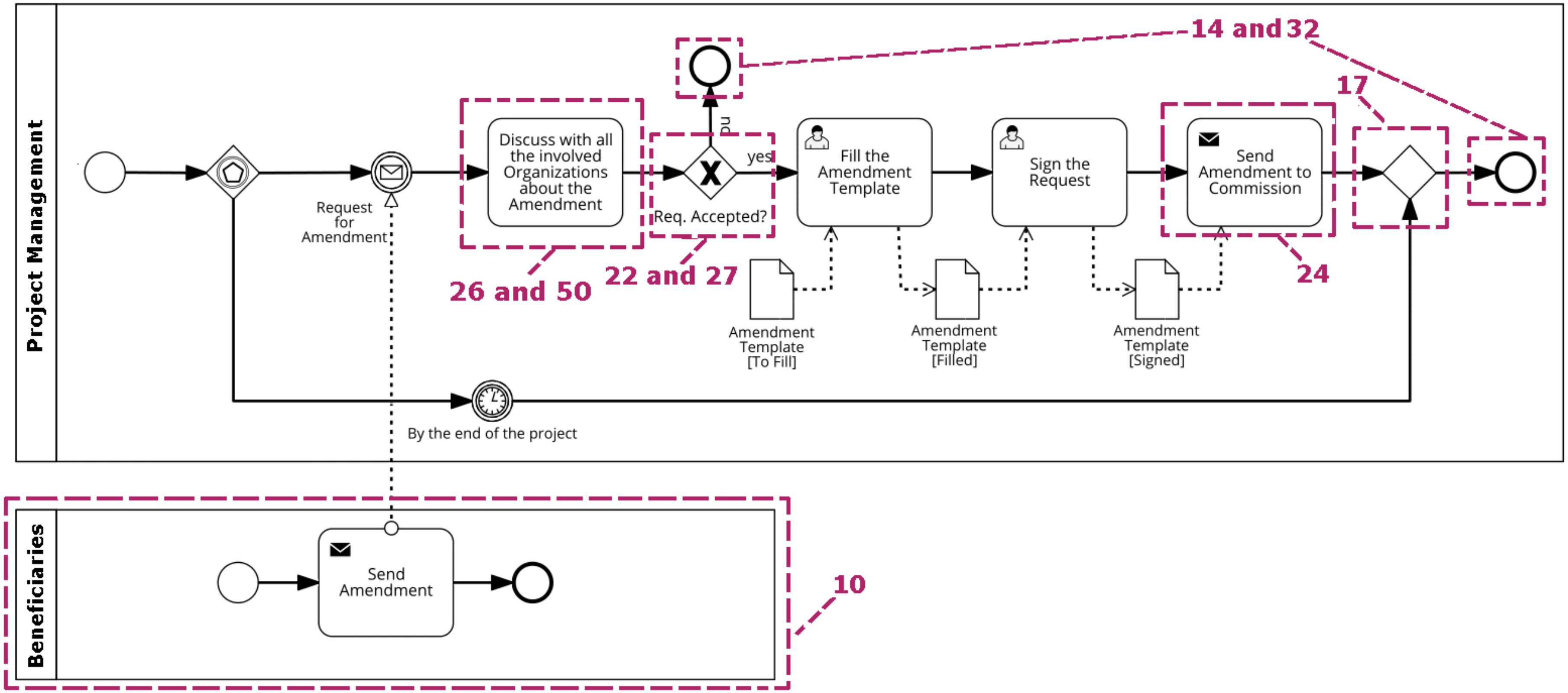


Figure 3: Manage Amendment model before re-design (it highlights guidelines violations).

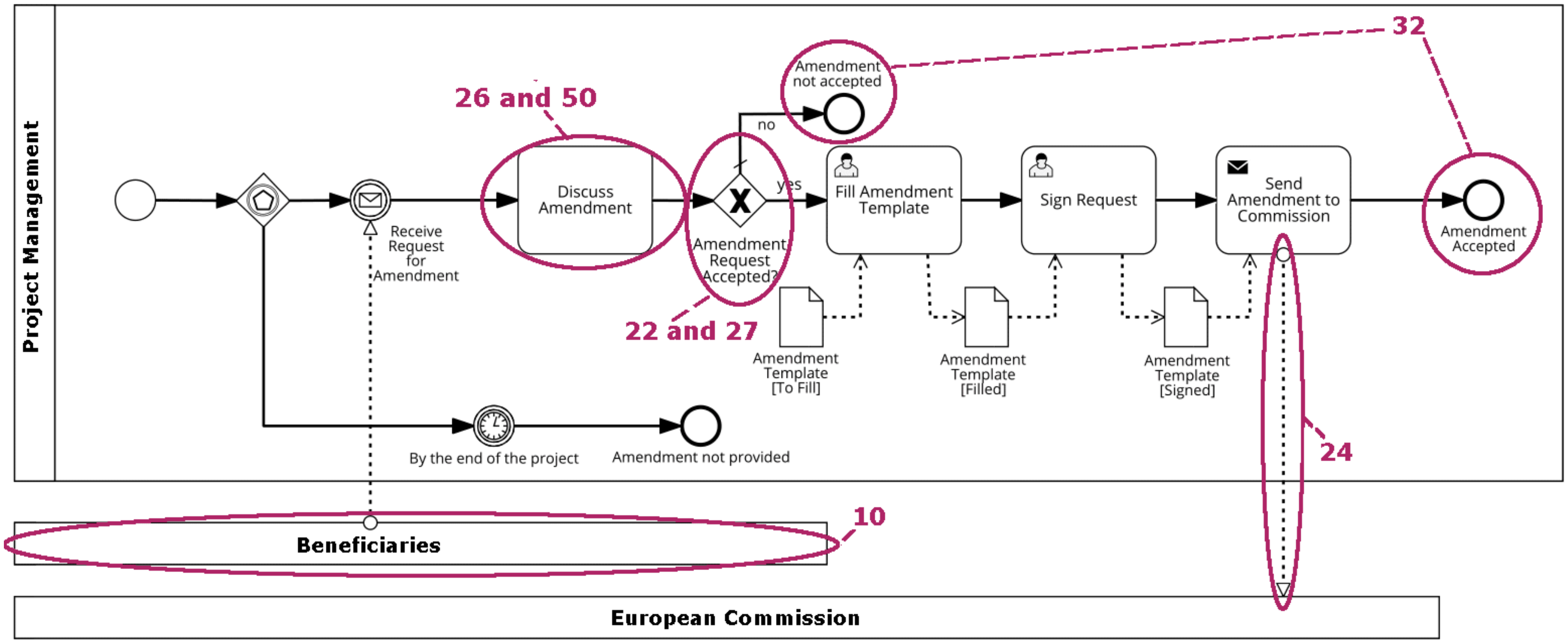


Figure 4: Manage Amendment model after re-design (it highlights guidelines application).

6.1.3 Details and Standard Format

Another guideline violation refers to the amount of details reported in the activity labeled "Discuss with all the Involved Organizations about the Amendment". This goes against the guidelines ***Document minor details*** (ID 26) and ***Keep a standard format*** (ID 50), which respectively tell the designer to leave details to documentation, keeping labels simple, and to keep a unique format along models avoiding the use of different font size, colors and boxes dimensions. Guideline 26 is not met because the activity name ("Discuss with all the Involved Organizations about the Amendment") introduces too many details, which the designer does not need. Guideline 50 is not met because the box representing the activity has a different size compared to the others. This representation can cause a model reader to question whether this representation implies a higher relevance for the specific activity with respect to others. We solved these violations by changing the activity label in "Discuss Amendment", moving the details inside the activity description, and resizing the activity.

6.1.4 Balance and Labelling

Another guideline that is not met involves the XOR Gateway with label "Req. accepted?". In particular, the guideline ***Use a labeling convention*** (ID 27) is violated. This is due to the label "Req. accepted?" that includes an abbreviation. We solved this guideline violation by rewriting the label as "Amendment Request accepted?" which explicitly expresses the type of request to be accepted.

6.1.5 End Events

The two end events in Fig. 3 are highlighted by BEBoP as sources of guidelines violation. Both events are without a label, and this goes against guideline **Labelling start and end events** (ID 32) which recommends to provide labels when the model includes more than one end event. In addition, according to guideline **Use end events consistently** (ID 14) events that represent the same end state should be merged in a single end event. To check this guideline, BEBoP reports a violation in any case in which more than one end event is present in the model. It is then responsibility of the model designer to check whether such violation is a true violation (i.e., in our case, if the two end events represent the same state) or not. In our case, to solve this violation the modeler has to establish if those events correspond to the same event or if they are actually distinct. In the former case the events must be merged in one, in the latter case a proper label has to be associated to those events to properly distinguish them. In our case, the events are actually distinct so we associate them different and proper labels. This example shows that some guidelines, such as ID 14, are verified by BEBoP only for the portion of the guideline that is algorithmically feasible, but the actual violation of the guideline has to be evaluated by a human assessor. Moreover, the designer explicitly added a new end event labelled "Amendment not provided" to properly distinguish all the different possible terminations of the process. To do that the designer made an explicit choice between guidelines ID 14 and ID 4 that, in the considered example, are in contrast to each other such as having different final end events reduce the symmetry of the model. Notably, not all the guidelines violations must be solved by the modeler, but he/she will have to establish if a particular violation is actually necessary: in this case having three end events violates guideline 14 but it is necessary for guaranteeing a correct understanding of the process.

6.1.6 Activity Description

The only guideline violation that we cannot directly see in the model provided in Fig. 4 is named **Provide activity description** (ID 8). This guideline suggests the designer to associate each activity with a brief description. Activity descriptions are attributes of the BPMN elements that are *hidden* at this level of representation. However, some activities description were actually missing in our initial model, so we added them. As an example we can refer to activity "Send Amendment"; it initially had no description associated with, so we added the following one: "In this activity the coordinator sends the amendment request to the EU Commission)". A description associated with each activity helps to maintain the model clear without introducing too many details in the model itself, but allowing the model reader to access those details if needed.

7 Related Works

In the last years, much effort has been devoted to the definition of understandability guidelines and related metrics and thresholds. As already discussed in Section 2 several research papers target this topic in a slightly different manner. In addition, commercial and open source tools are available to check at least a subset of the guidelines available in the literature. In Section 7.1, we report those contributions that have as a target the derivation of consistent sets of guidelines possibly in relation to understandability. With respect to such works we highlight the major differences with what presented in this paper. In Section 7.2 we summarize the related works on metrics and thresholds that we leveraged to define our guidelines, and to implement our tool. Finally, in Section 7.3, we compare existing software for guidelines verification with BEBoP.

7.1 Understandability Guidelines Frameworks

In the literature there are few attempts to derive a unified guideline framework for the BPMN notation. Nevertheless, there are some works that, even though they have a different objective, such as model quality in general, can be clearly related to what is presented here.

Probably one of the first work that tries to provide an understandability guidelines framework has beed defined by Mendling et al [2007b]. This work reports the results of a study on the dimension of BP models. The authors concluded that the size of the model is a dominant aspect for understandability. In successive studies, they defined a set of process modeling guidelines which can be applied to multiple notations. The guidelines improve the quality of the models, in that their application makes the model more understandable and less prone to errors [Mendling et al, 2010b, 2012a]; the guidelines are defined for Event Process Chain (EPC) but are also applicable to BPMN. Interestingly in Sánchez-González et al [2015] the authors apply the 7 guidelines defined in [Mendling et al, 2010b] to a real complex scenario. The included discussion could be the base for the identification of general strategies to be defined in order to refactor models, so that they will abide to the defined guidelines. The use case methodology applied could be fruitfully adopted to evaluate also the guidelines not included in [Mendling et al, 2010b].

Regarding BPMN 2.0, a relevant work that specifically focuses on guidelines is provided by Silingas and Mileviciene [2011]; the authors analyzed six BPMN models, and identified the *bad smells* – i.e., modelling approaches negatively impacting on model quality – that they contained. Then, they suggested best practices/patterns to comply with the bad smells, and showed how refactoring the original versions led to better quality models. The cases dealt with mixed style names, large process diagrams, inconsistent use of gateways, inconsistent use of events, inconsistent use of loops, and poor diagram layout. The approach of Silingas and Mileviciene [2011] differs from ours since it is example driven and based on the practical experience of the authors, while our approach is driven by literature. Furthermore, Silingas and Mileviciene [2011] focus only on a subset of the potential understandability defects, while our effort is mainly devoted to have a comprehensive view of the guidelines available in the literature.

On the application of guidelines, an interesting contribution is given by Leopold et al [2016]. The authors focus on quality issues of 585 BPMN 2.0 process models from industry, highlighting which guidelines (collected from specific works, [Allweyer, 2009], [Silver, 2011], [White, 2008]) are not followed. Leopold et al [2016] also developed a tool to verify the guidelines, with the goal of understanding the most frequent understandability defects in real-world models. Clearly the goal of our work is different, since we want to provide an homogeneous set of understandability guidelines, rather than empirically identify the most commonly violated, so to provide a kind of relevance list for the studied guidelines. It is worth mentioning that the tool developed by Leopold et al [2016] has not been released to the research community, while BEBoP is freely available. In addition the guidelines considered in Leopold et al [2016] are a proper subset of the ones identified by us.

Another relevant work is provided by Silver [2011], who suggests the use of an approach called *method and style* to help the model designer in designing BPMN models that are correct, complete, and clear. Although this work provides clarification on the usage of BPMN elements and it specifies good practices of modeling, it does not associate the concepts of metrics and thresholds to them. The main difference with our work is that the authors do not suggest a way to verify the application of those best practices that is needed to automatically check if the model fits with the guidelines or not.

The closest work to the one presented in the current paper is the one by Moreno-Montes de Oca and colleagues [Moreno de Oca et al, 2015; Moreno de Oca and Snoeck, 2014], and for this reason we provide here a quite detailed comparison. The objective of these papers is to provide modeling guidelines oriented to quality in general, even though most of the guidelines are directly related to understandability aspects (roughly 24 out of 27 of the guidelines described by Moreno de Oca and Snoeck [2014]). Notably, in our work we include almost all the understandability guidelines already identified by Moreno-Montes de Oca and colleagues. In addition, we also consider as a possible understandability problem the usage of OR-Join elements, and we suggest to reduce them when possible (ID 23 in their list, ID 21 in our list). In Moreno de Oca and Snoeck [2014] the usage of OR-joins is mainly considered a possible source of semantic issues, rather than an understandability problem. Another aspect that is not considered in Moreno de Oca and Snoeck [2014], refers to the exchange of messages in collaboration diagrams. In this respect only a labelling related guideline (ID 27 in their list, part of ID 30 in our list) is reported. Our list instead includes 3 additional guidelines on this aspect (ID 23, 24, 46). Some interesting difference we found with [Moreno de Oca and Snoeck, 2014] refers to the clear highlighting of some relevant paths. In particular, in our guidelines we suggest to have different end events for each erroneous termination (ID 12 and 14 our list), while in [Moreno de Oca and Snoeck, 2014] only one end-event is suggested in this case (ID 5 their list). In our framework we favored the multiple erroneous end-node way of modeling that should permit to convey useful information to the reader. At the same time, our guidelines suggest to clearly identify the "happy path" (ID 5) which is in line also with the typical description of scenarios in requirements engineering. This suggestion is not reported in Moreno de Oca and Snoeck [2014]. On the other hand there are 7 guidelines (ID 7-11, 17, 24) in Moreno de Oca and Snoeck [2014] that we do not consider in our list. The first 5 in this set refer to the number of occurrences for specific notation elements. We thought that these guidelines are well subsumed by the general one suggesting not to use more than 31 elements (n. 1 in their list, n. 2 in our list). The other two guidelines the we do not include are ID 17 and 24. The first one is considered a bad modeling style for us and is somehow subsumed by our 7th guideline, the second one seemed to us rather obvious and difficult to apply. We assume that modelers try to keep models simple and they are not in general able to recognize when they are "over-modeling". On the other hand, we think that many other guidelines will help the modeler to possibly spot when the model smells of "over-modeling". Moreover, the two works provide similar definitions for the metrics of shared guidelines even though Moreno de Oca and Snoeck [2014] do not provide any tool for the automatic check of the guidelines framework.

The discussion above summarizes the main differences between the two works at a macroscopic level. Other minor differences are not reported here. It is also worth mentioning that all the guidelines we identified cannot be fully traced to guidelines listed by Moreno de Oca and Snoeck [2014] – differences in the granularity of some guidelines exist, so the matching among the two lists is not one to one.

Finally, it is worth to mention the work of Figl [2017] which aims at categorizing and summarizing systematically existing findings on the factors influencing the comprehension of processes. This work is relevant since it focus on aspects influencing the comprehensions of models and it can certainly be a source of inspiration to define new guidelines.

7.2 Metrics and Thresholds

Considering BP modeling metrics, we found relevant the work by Rolón et al [2006] and the one by Reynoso et al [2009]. They focus on the same topic which is the definition of metrics that can be applied in order to quantify the understandability and modifiability of conceptual BPMN 1.0 models. Another work by Mendling [2008] presents a set of metrics that captures various aspects of the structure and the state space of a process model. Although the cited work focus on BPMN 1.0, since the 2.0 version was not released yet, we can confirm that the metrics can be also applied on BPMN 2.0. More generally, works that provide a collection of metrics can be found in the literature [Muketha et al, 2010] [Kluza and Nalepa, 2012] [Kluza et al, 2014]. These contributions supported the definition of our list of metrics reported in Appendix B.

Finally, BP modeling *thresholds* are also an investigated topic in the literature. On this area, we found relevant the works by Sánchez-González et al [2010] and Sánchez-González et al [2011]. They studied structural metrics and their connection with the quality of process models. Then, they determined thresholds values to distinguish different levels of process model quality. Also Mendling et al [2012a] derived thresholds and established that size and complexity are general driving forces of error probability in modeling. In our work, we took advantage of the thresholds defined in the literature, and we associated them to modeling guidelines.

7.3 Comparison of Different Business Process Verification Tools

Several tools exist that allows to verify quality aspects of BP models. Among such tools, the most widely used are Signavio 10.1[11], No Magic MagicDraw 18[12], Bizagi Process Modeler 3.0[13] and Camunda Community Edition[14]. All these tools provide BP model editing capabilities besides BP model verification capabilities. In this section, we compare the verification capabilities of BEBoP with those provided by these platforms. The evaluation of the tools was performed by designing *ad hoc* models violating specific guidelines in the defined set, and by testing those models in each tool to check whether the tool was able to spot out the guideline violations.

Table 8 summarizes the comparison performed. The first and the second column show the ID and the name of the guideline. The other columns respectively show, for each tool, if the guideline is automatically checked. The cell of the table is empty if the guideline is not automatically checked, and is marked with "yes" otherwise.

With respect to the 50 guidelines, our tool allows to automatically verify 34 guidelines, Signavio 22, MagicDraw 8, Bizagi 7, and Camunda 5. Only one of the guidelines that is checked by the other tools is not considered by BEBoP, namely guideline 44 - "Avoid overlapping elements" – see Table 6. BEBoP does not check this guideline since it is not a model editor, as the other tools. Although it can access element positions from the BPMN XML file, we considered that the actual layout shown to the user also depends on the editor. Indeed, most of the editors are not fully compliant with the BPMN 2.0 standard, and appear to partially use the information related to the position of the elements. Hence, a violation found in the XML file might not correspond to an actual violation in the graphical representation.

It is also important to highlight that Signavio, MagicDraw and Bizagi are proprietary tools, instead our tool is released under GPL license, and it is a Web service. Hence, it is ready to be integrated easily in several architectures, and can be extended by the BP community with novel guidelines.

8 Conclusions and Future Work

This paper aims at providing means for improving the design of BP models, with a specific focus on the understandability of models. We tackle this issue by providing BP modeling guidelines that suggest appropriate design decisions that lead to more understandable models. To achieve our goal we identify, consolidate and homogenize a set of guidelines from the literature. As output of this activity, we provide a list of 50 understandability modeling guidelines that when possible are associated to metrics and thresholds. To our knowledge, this is the most complete list available. The guidelines are general purpose, and this lets the modelers free to chose those that are more relevant according to the modeling purpose – e.g., process learning, information system development.

[11] `http://www.signavio.com/products/process-editor/`
[12] `http://www.nomagic.com/products/cameo-business-modeler.html`
[13] `http://www.bizagi.com/en/products/bpm-suite/modeler`
[14] `https://camunda.org/download/modeler/`

Table 8: Comparison of BEBoP with other tools

Guideline ID	Guideline Name	Tool Name				
		BEBoP	**Signavio**	**MagicDraw**	**Bizagi**	**Camunda**
1	Validate models	yes	yes	yes	yes	yes
2	Minimize model size	yes	yes			
3	Apply hierarchical structure with sub-processes	yes	yes			
4	Apply symmetric modeling					
5	Highlight the "happy path"					
6	Minimize concurrency					
7	Model loops via loop activities	yes				
8	Provide activity descriptions	yes	yes	yes		
9	Minimize gateway heterogeneity	yes				
10	Use pools consistently	yes	yes			
11	Use lanes consistently	yes				
12	Use start and end events explicitly	yes	yes		yes	yes
13	Use start events consistently	yes				
14	Use end events consistently	yes		yes		
15	Restrict usage of terminate end events	yes				
16	Use explicit gateways	yes	yes	yes		
17	Mark exclusive gateways	yes				
18	Split and join flows consistently	yes	yes			
19	Balance gateways	yes	yes			
20	Use meaningful gateways	yes	yes			
21	Minimize inclusive OR gateways	yes				
22	Use default flows	yes				
23	Use messages consistently					
24	Use message flows	yes		yes		
25	Use task types consistently					
26	Document minor details					
27	Use a labeling convention					
28	Labelling pools	yes	yes	yes	yes	yes
29	Labelling lanes	yes	yes			
30	Labelling activities	yes	yes	yes	yes	yes
31	Labelling events	yes	yes		yes	
32	Labelling start and end events	yes	yes			
33	Labelling message event	yes	yes			
34	Labelling XOR gateway	yes	yes		yes	
35	Labelling AND gateways	yes				
36	Labelling converging gateways	yes				
37	Labelling data objects	yes	yes			
38	Labelling synchronised end/split					
39	Include loop marker annotations	yes		yes		
40	Reduce the number of redundant activities					
41	Use sub-processes					
42	Use sub-processes to scope attached events					
43	Design neat and consistent models					
44	Avoid overlapping elements		yes		yes	yes
45	Use linear sequence flows	yes	yes			
46	Use linear message flows	yes	yes			
47	Use a consistent process orientation	yes	yes			
48	Organize artifacts flows					
49	Associate data objects consistently					
50	Keep a standard format					
	Total	34	22	8	7	5

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.