A Study on the Usage of the BPMN Notation for Designing Process Collaboration, Choreography, and Conversation Models

Ivan Compagnucci[1], Flavio Corradini[1], Fabrizio Fornari[1*], and Barbara Re[1]
[1]Department of Computer Science, University of Camerino,
Via Madonna delle Carceri 7, Camerino, 62032, Italy

*E-mail: fabrizio.fornari@unicam.it

Being widely accepted by industries and academia, Business Process Model and Notation (BPMN) is the de facto standard for business process modeling. However, the large number of notation elements it introduces makes its use quite complex. This work investigates the usage of the BPMN notation by analyzing 54,500 models harvested from seven online collections. The study considers different model types introduced by the standard, such as process collaboration, choreography, and conversation. The analyses focus on the syntactic dimension of BPMN, investigating the usage of BPMN elements and their combinations. Syntactic violations of the standard, and of good modeling practices, are also investigated as well as possible relations with BPMN elements and modeling tools. The results of this study can guide further activities of educators, practitioners, researchers, and standardization bodies.

Keywords - *BPMN; Process Models; Notation Usage; Syntactic Errors; Modeling Practices.*

1 Introduction

Business Process Model and Notation (BPMN) was proposed by the Business Process Modeling Initiative in 2004. Since 2013 it has been recognized as a standard[1] and consistently maintained by the Object Management Group. In this study, we focus on version 2.0 of the notation[2] and we use the term BPMN to refer to such a version.

BPMN holds the promise of providing a standardized *lingua franca* easily understood by all business stakeholders (e.g., business analysts, technical developers, and business managers) involved in the de-

[1]BPMN 2.0.1 version is released as ISO/IEC 19510:2013 standard
[2]`https://www.omg.org/spec/BPMN/2.0` - Accessed on 9 March 2023

sign, implementation, monitoring, and management of the processes (Bork, Karagiannis, & Pittl, 2020; Genon, Heymans, & Amyot, 2010; Recker, Rosemann, Indulska, & Green, 2009). In particular, the adoption of BPMN has been successful thanks to its intuitive graphical nature, which foster the understanding of business processes by all the involved stakeholders. Moreover, the availability of supporting tools, in the market, positively influenced BPMN adoption.[3]

If on one side, the intuitive graphical nature of BPMN makes it easy to start with, on the other side, the huge amount of notation elements makes it challenging to master. BPMN includes over two hundred distinct graphical notation elements (see Section 2 for more details). As a result, several BPMN models present modeling issues, such as syntactic errors, poor understandability, etc. In such a setting, inspired by studies conducted on other standard notations such as UML (Dobing & Parsons, 2006; Petre, 2013), we want to shed light on how BPMN notation is used in practice.

To guide our study, we defined four research questions.

- *RQ1. How many and which notation elements are used to design BPMN models?*
- *RQ2. Which BPMN elements are used in combination?*
- *RQ3. Which syntactical errors occur more frequently, and how are they related to BPMN elements and modeling tools?*
- *RQ4. Which good modeling practices are violated, and how are they related to specific BPMN elements and modeling tools?*

To answer those questions, we first retrieved a set of **54,500** models from seven online BPMN collections, including *Process Collaboration*, *Choreography*, and *Conversation* models, and then we conducted an empirical analysis of such models. In this paper, for presentation purposes, we focus on reporting only the main data and results that contribute to answering the identified research questions. Additional materials, including all the data and the analysis results, are available online.[4]

Our work differs from previous studies in the field such as Compagnucci, Corradini, Fornari, and Re (2021); zur Muehlen and Recker (2013) by: (i) considering a broader collection of models, (ii) analyzing different types of models such as process collaborations, choreographies, and conversations, (iii)

[3]More than 70 tools support BPMN (`http://www.bpmn.org` - Accessed on 9 March 2023)

[4]`https://drive.google.com/drive/folders/1Ol6QBXVEs_jZ0vQ6o2oQlWyqesHSJqfL` - Accessed on 9 March 2023

investigating relations between syntactic errors and both elements and modeling tools. (iv) investigating relations between good modeling practices and both elements and modeling tools; (v) making available both data and tools used to conduct our study. The overall results of our study could be used to guide: educators in tailoring their teaching activities, practitioners in the use of the notation, researchers towards novel findings, and standardization bodies on future releases of the standard.

The rest of the paper is organized as follows. Section 2 presents background knowledge regarding BPMN. Section 3 describes all the steps adopted for conducting this research. Section 4 reports the results of the various analyses we conducted. Section 5 discusses the obtained results. Section 6 reports a comparison of our study with related works. Section 7 reports the implications of the study providing indications for further research activities. Section 8 reports some limitations of our study. Finally, Section 9 summarizes and concludes the paper.

2 Background

In this section, we report background information about BPMN elements, types of models, and available BPMN model collections.

2.1 BPMN Core and Extended Sets

The BPMN notation includes a core set of elements reported in Figure 1. These elements can be grouped in flow objects, connecting objects, swimlanes, and artifacts. The interested reader can refer to the official BPMN specification[5] for an exhaustive description.

The elements can be enriched with symbols (markers) to represent additional information. Markers are reported at the top left corner, or at the bottom center, of the element. Figure 2 reports a task with different combinations of markers (e.g., service, parallel multiple instance, compensation, loop). By treating each element with a unique combination of markers as separate elements, the maximum number of BPMN elements that can be detected is 267. This number corresponds to the ***theoretical complexity*** of the notation (Erickson and Siau (2004)), and it tells us that for designing a BPMN model we could use up to 267 different graphical elements.

While the size of a model corresponds to the total number of elements present in the model, the ***practical complexity*** refers to the number of different types of elements used (Erickson and Siau (2007), Siau, Erickson, and Lee (2005)). As an example, a model that presents one start event, one activity,

[5] `https://www.omg.org/spec/BPMN/2.0/PDF` - Accessed on 9 March 2023

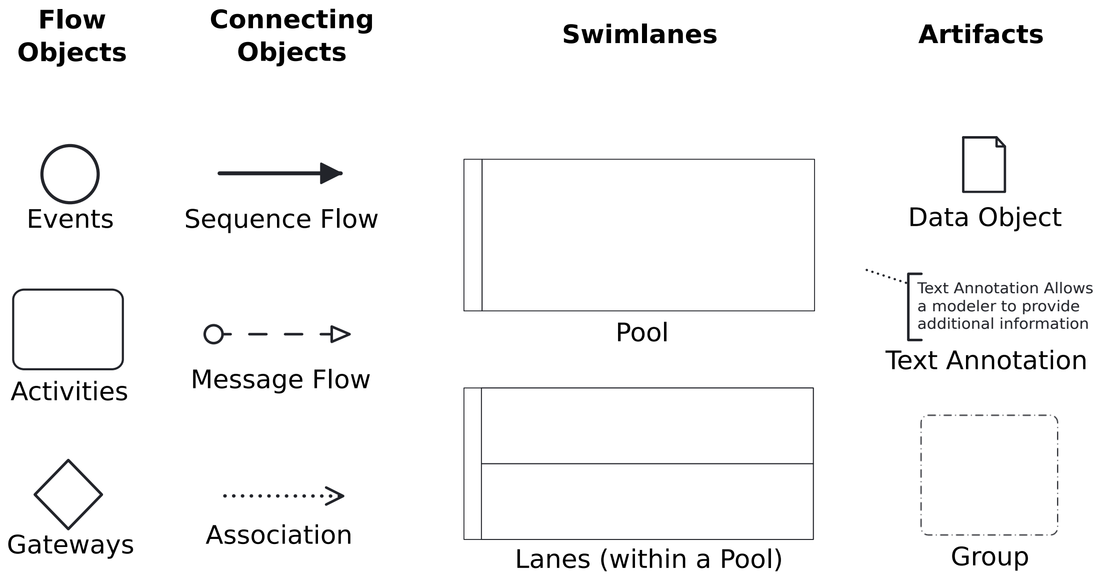
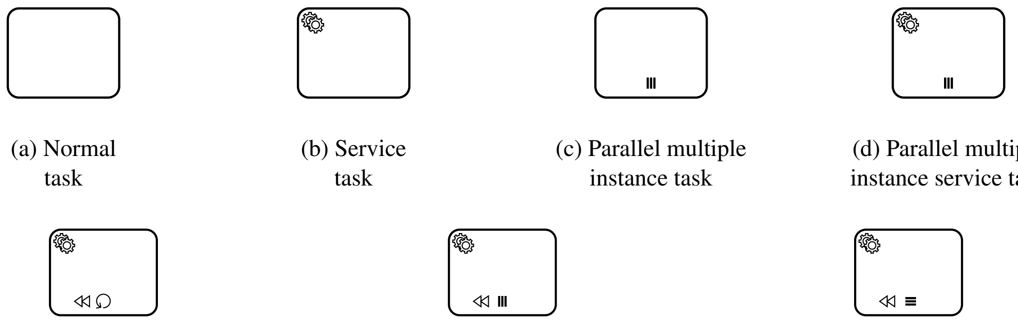


Figure 1: Core set of BPMN elements (The Object Management Group, 2011)



(a) Normal task (b) Service task (c) Parallel multiple instance task (d) Parallel multiple instance service task

(e) Compensate standard loop service task (f) Compensate parallel multiple instance service task (g) Compensate sequential multiple instance service task

Figure 2: Different definitions of the task element

one end event, and two sequence flows connecting the elements has a model size of five, since the total amount of elements is five, but a practical complexity of four, since it uses four different types of elements.

2.2 BPMN Models

A relevant aspect of the BPMN standard refers to the adoption of different model types such as: process, collaboration, choreography, and conversation. A process model describes how a single participant works, a collaboration model describes how multiple participants work and interact with each other, a choreography model describes the exchange of information between participants, and a conversation model provides a high-level representation of groups of messages (conversations) transmitted between process participants. In particular, of the 267 BPMN elements that are included in the standard, 242

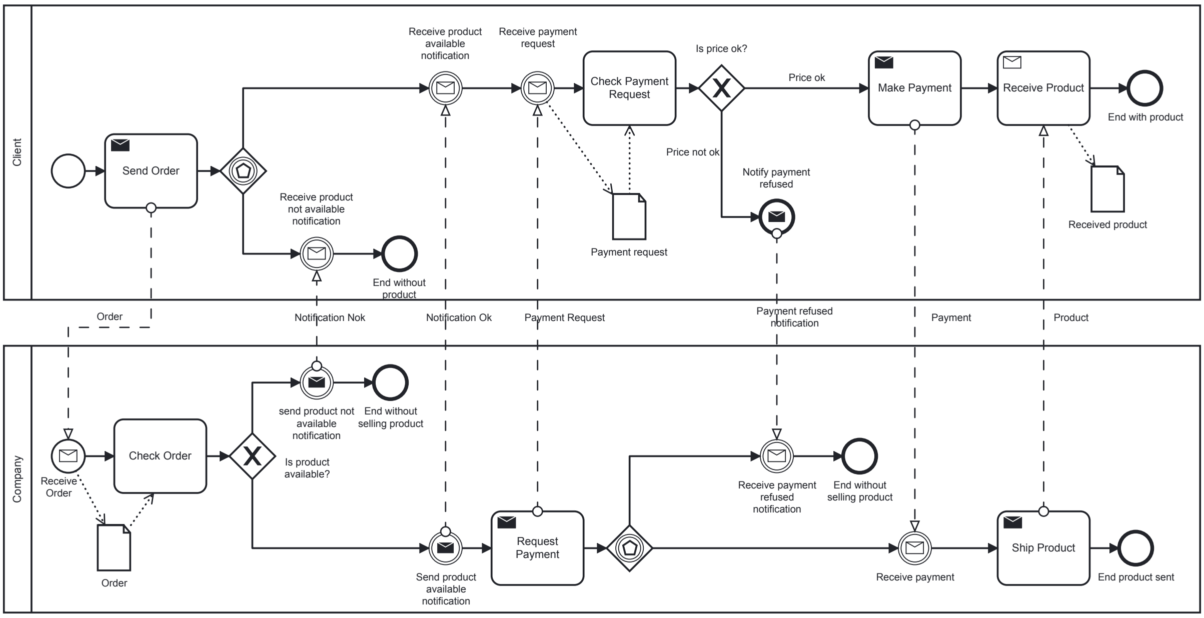


Figure 3: An example of process collaboration model

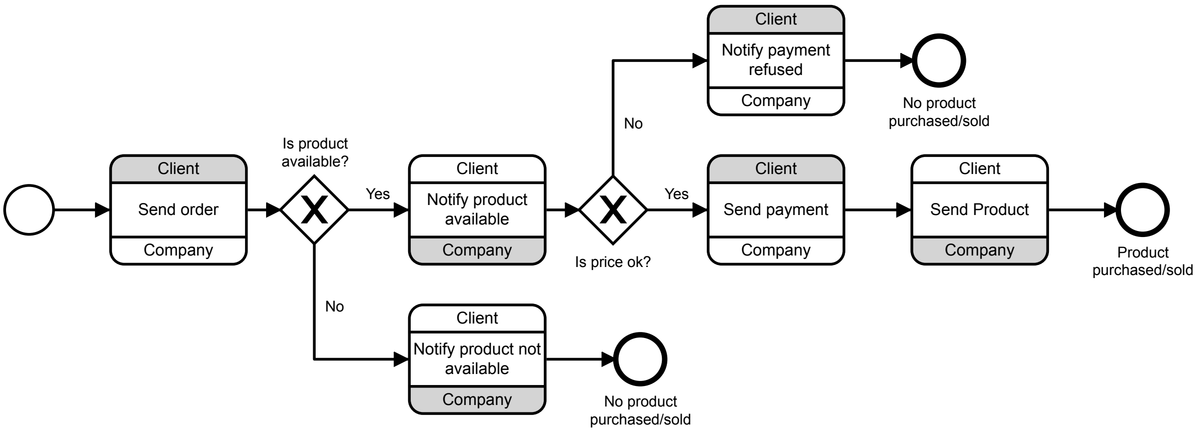


Figure 4: An example of a choreography model

can be used to design process models, 244 elements can be used to design collaboration models, 91 for choreography models, and 11 for conversation models. Those numbers correspond to the theoretical complexity of the different models.

Some elements are common in all types of models while others are specific to the type of models (e.g., choreography activity and conversation link are specific to choreography and conversation respectively). In addition, since the set of notation elements used for single processes is entirely included in that of collaborations[6], in the rest of the study, we will group processes and collaborations into a combined category "process collaboration". Figure 3 reports an example of a business process collaboration while Figure 4 and Figure 5 report the corresponding choreography and conversation models.

[6]The graphical notation for designing collaborations includes also *message flows* and a *message icon* that are not present in the notation for designing single processes.

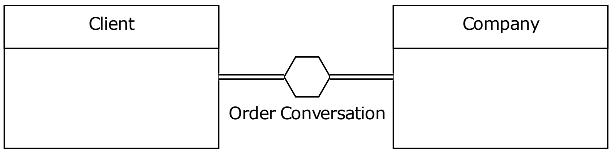


Figure 5: An example of a conversation model

The standard admits the possibility to combine elements of different models leading to the design of *combined models* (e.g., choreography elements can be combined with process elements, see pp. 363-364 of the BPMN specification[7]). This can impact the practical complexity of the models. It is also noteworthy that BPMN offers the capability to extend the notation by introducing changes to the already available elements by adding new graphical symbols, therefore leading to an increase in the notation's theoretical complexity.

2.3 BPMN Model Collections

For conducting our study we have identified seven BPMN collections from which we harvested models. Table 1 presents a summary of the repositories that have been considered.

The *BIT process library* collection[8] is composed of 850 models that have been made available to researchers by the IBM group for the definition and validation of an approach for checking business process properties (Fahland et al., 2009). The *Camunda BPMN* collection[9] includes 3,739 models created during BPMN training sessions. The *eCH-BPM* collection[10] is composed of 117 models designed involving BPM specialists and people working in Swiss public municipalities and federal agencies. The *GenMyModel* collection[11] includes 12,172 models derived from the GenMyModel platform. The *GitHub* collection[12] is composed of 17,314 models publicly available on GitHub that have been retrieved from a systematic mining approach (Heinze, Stefanko, & Amme, 2020). The *RePROSitory* collection[13] consists of 593 models that have been uploaded on the RePROSitory platform (Corradini, Fornari, Polini, Re, & Tiezzi, 2019). The *BPMAI* collection[14] includes 19,715 models mainly derived from modeling activities carried out by students within academic institutions and made available by the BPM Academic Initiative.

[7] BPMN specification: `https://www.omg.org/spec/BPMN/2.0/PDF` - Accessed on 9 March 2023
[8] `http://www.zurich.ibm.com/csc/bit` - Accessed on 9 March 2023
[9] `https://github.com/camunda/bpmn-for-research` - Accessed on 9 March 2023
[10] `http://www.ech-bpm.ch/de/process-library` - Accessed on 9 March 2023
[11] `https://app.genmymodel.com/explore` - Accessed on 9 March 2023
[12] `https://github.com/ViktorStefanko/BPMN_Crawler` - Accessed on 9 March 2023
[13] `http://www.pros.unicam.it/reprository` - Accessed on 9 March 2023
[14] `https://zenodo.org/record/3758705` - Accessed on 9 March 2023

Models Repositories	Number of Harvested Models	Models Providers	Models Origin
BIT Process Library	850	IBM Group	Industry
Camunda BPMN	3 739	Camunda Company	Training session
eCH-BPM	117	Swiss Government	Public administration
GenMyModel BPMN	12 172	GenMyModel's Users	Mixed
GitHub BPMN	17 314	GitHub's Users	Mixed
RePROSitory BPMN	593	RePROSitory's Users	Mixed
BPMAI	19 715	BPM Academic Initiative	Training session

Table 1: Overview of harvested models by repository

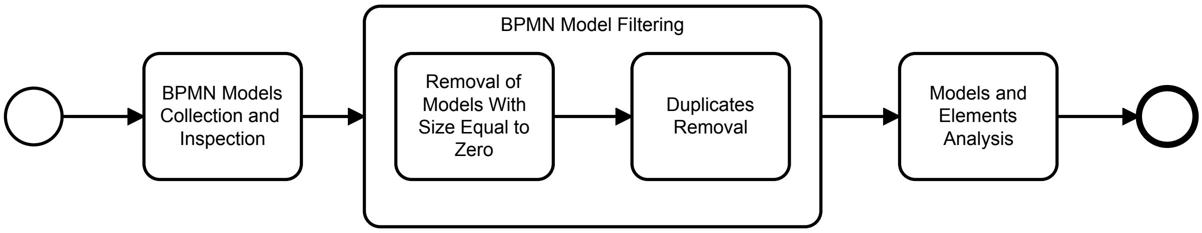


Figure 6: Graphical representation of the adopted methodology

3 Methods

To conduct our analysis, we harvested 54,500 BPMN models scattered among the collections introduced in Section 2. This section describes the methodology we used to conduct the study. We also describe the procedure for filtering out duplicate models and models with zero elements. Finally, we report on the type of analyses performed. Figure 6 reports an overview of the adopted methodology.

3.1 BPMN Model Inspection

In order to inspect the content of the model collections and to extract information related to the BPMN elements used, we developed and distributed an open-source java application named *BPMN Inspector*. We report the source code on a GitHub repository.[15] BPMN Inspector can detect and distinguish up to 267 graphical BPMN elements. The java application takes as input a set of BPMN models (*.bpmn*) and returns a comma-separated value file (*.csv*) containing for each analyzed model: the type and amount of elements present in the model, the model type (i.e., process collaboration, choreography, conversation), and the tool adopted for the design of the BPMN models. We based the elements detection on the XML tags defined in the BPMN 2.0 schema. As an example, the tag <*bpmn:startEvent*> identifies the presence of a start event while the tag <*bpmn:task*> identifies the presence of a task element.

For detecting the modeling editor, the tool relies on the value assigned to the "*exporter*" and "*tar-*

[15] `https://anonymous.4open.science/r/BPMN-Analytics-Repository/BPMNInspector` - Accessed on 9 March 2023

getNamespace" attributes from the *.bpmn* file. As an example, to a model that reports *targetNamespace="http://bpmn.io/schema/bpmn"* and *exporter="Camunda Modeler"* we assign *Camunda* as the modeling tool. If the exporter attribute is present, we use it as the main indicator otherwise, we rely on the targetNamespace attribute. If none of the two attributes are present, we assign "undefined" to the modeling tool in the .csv file.

3.2 BPMN Model Filtering

The filtering procedure allows for ensuring more truthful analyses. It involves two operations: discarding empty and duplicate models. Overall, from the initial set of 54,500 models, we removed 885 models with 0 elements through the filtering procedure. To identify duplicates, we generated a string for each model composed of three different information: (i) the number of occurrences of each element present in a model; (ii) the size of the *.bpmn* model file expressed in kilobytes; (iii) all the labels in the model. Figure 7 provides an example of such a string. We then checked for duplicated strings and removed the corresponding duplicated models. We removed 13,929 models resulting in a collection of 39,695 unique models. The final set consisted of 38,863 process collaboration models, 543 choreography models, and 289 conversation models.

3.3 Models and Elements Analysis

We conducted analyses based on the models derived from the filtering procedure to address the proposed research questions. Table 2 summarizes the analyses and the tool used. We analyzed a subset of those models according to the research questions. All the analyses were carried out on each type of model (e.g., process collaboration, choreography, and conversation) except for the analyses of the RQ4 that were performed only over valid process collaboration due to the limitation of the tool we used to address the research question.

RQ1 BPMN element usage. To determine the number and types of elements used, we first calculated the model size, i.e., the number of elements used to design each model and the number of occurrences of

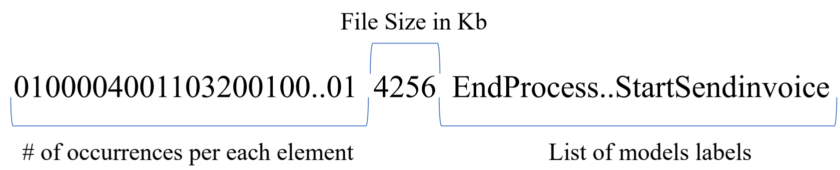


Figure 7: An example of the string obtained from a model

Scope	Analysis	Tools
RQ1 Element Usage	Number of BPMN elements used in a model Occurrences of specific BPMN elements Distribution of BPMN elements over models Variety of BPMN elements over models	BPMN Inspector
RQ2 Combined use of elements	Correlation between pairs of BPMN elements Combination frequency of BPMN elements	BPMN Inspector
RQ3 BPMN Syntactic validation	Syntactic errors in BPMN models	BPMN Validator
	BPMN element related syntactic errors Modeling tools related syntactic errors	BPMN Inspector
RQ4 BPMN Good modeling practices violations	Violation of BPMN good modeling practices	BEBoP
	BPMN element related good modeling practices violations Modeling tools related good modeling practices violations	BPMN Inspector

Table 2: Analyses performed over BPMN models

each of the 267 BPMN elements. To answer this question, we use the BPMN Inspector tool. Then, we calculated the distribution of BPMN elements over the analyzed models to discover whether elements were used in all the models or never used. Finally, we investigated the variety of BPMN elements over models, i.e., the number of different types of BPMN elements used for designing a model (which we also refer to as practical complexity).

RQ2 Combined use of BPMN elements. We first checked possible correlations among pairs of BPMN elements to gather information about possible combinations, relying on the Pearson correlation coefficient (ρ) (Benesty, Chen, Huang, & Cohen, 2009). To perform this analysis, BPMN Inspector relies on a python script that we share on GitHub.[16] Then we targeted groups of elements. We started with the pair of elements that are mostly used in combination, and then we proceeded by adding the element most frequently used with that combination, and so on. Since this type of analysis requires at least two elements, we considered only models with more than one element.

RQ3 BPMN syntactic validation. To check whether the collected models include syntactic errors, we compared each model with the BPMN XML schema provided by the OMG group.[17] We conducted the analysis using a validator we developed. The source code of the validator is reported on GitHub[18]. The validator detects whether a model is syntactically valid; if not, it lists the errors. Then, BPMN Inspector analyses the errors detected to check whether they could be related to specific BPMN elements or the modeling tools used to design the models. In particular, the error messages specify the type of BPMN element involved, making it possible to check if syntax errors are associated with the element.

[16] `https://anonymous.4open.science/r/BPMN-Analytics-Repository/CorrelationbetweenpairsofBPMNelements/BPMNPearsonCorrelation.ipynb` - Accessed on 9 March 2023- Accessed on 9 March 2023
[17] `https://wiki.xmldation.com`- Accessed on 9 March 2023
[18] `https://anonymous.4open.science/r/BPMN-Analytics-Repository/BPMNValidator` - Accessed on 9 March 2023

An example of an error message is *cvc-complex-type.4: Attribute "sourceRef" must appear on element "bpmn2:sequenceFlow"* which we can clearly relate to the sequenceFlow element.

RQ4 BPMN good modeling practices. To determine if the models we collected violated established good modeling practices (Corradini, Ferrari, et al., 2018), we used a tool called BEBoP[19] (understandaBility vErifier for Business Process models). BEBoP is a tool that verifies the understandability of business process models, ensuring that they have been designed according to established modeling practices. It automatically checks whether a model is syntactically correct and adheres to these practices. Then, we used BPMN Inspector to investigate whether modeling practice violations are related to specific BPMN elements or to the modeling tools used for designing the models.

4 Results on BPMN Elements Analysis

This section reports the most significant results from the analysis guided by the research questions. The interested reader can find all the data and the analysis results online.[20]

4.1 BPMN Element Usage

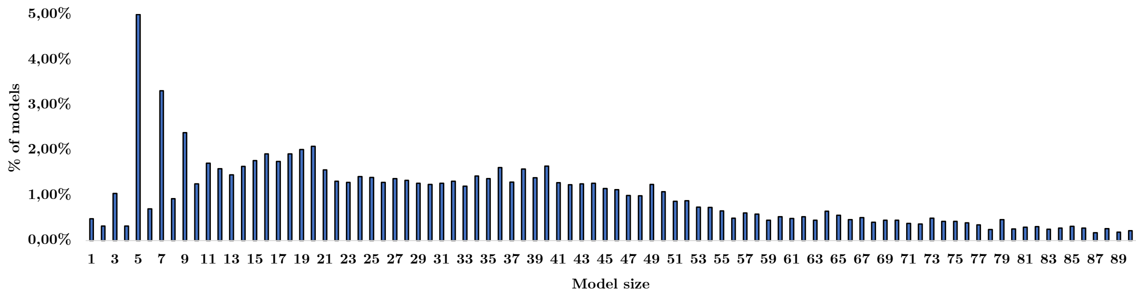
Focusing on the BPMN element usage, we report the analysis results on the number of BPMN elements in a model, the occurrences of specific BPMN elements, and the distribution and variety of elements over models.

Number of BPMN elements in a model. Figure 8 reports the percentage of process collaboration, choreography, and conversation models by size. The number of elements in process collaborations ranges from 1 to 3,310, with an average of 32 and a median of 41. On average, choreography models contain 26 elements, ranging from 2 to 159, with a median of 24. Finally, conversation models typically are composed of 17 elements, ranging from 2 to 129, with a median of 12.

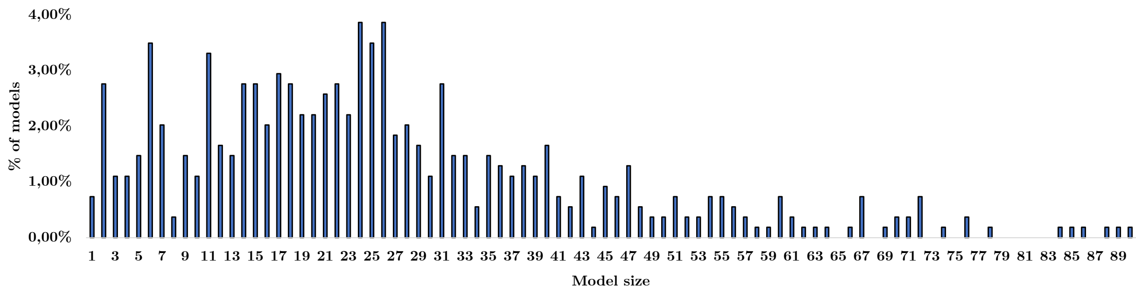
Occurrences of specific BPMN elements. Table 3 also reports a list of the BPMN elements ranked based on the number of occurrences in the analyzed models. The most used elements in process collaboration models are sequence flow, normal task, message flow, lane, end event, and exclusive gateway. Notably, these elements all belong to the BPMN core set. In choreography models, the most used elements are choreography participant, sequence flow, choreography task, and message. The most used elements in conversation models are conversation links, collapsed pools, and conversation.

[19]BEBoP tool: `https://pros.unicam.it/bebop` - Accessed on 9 March 2023

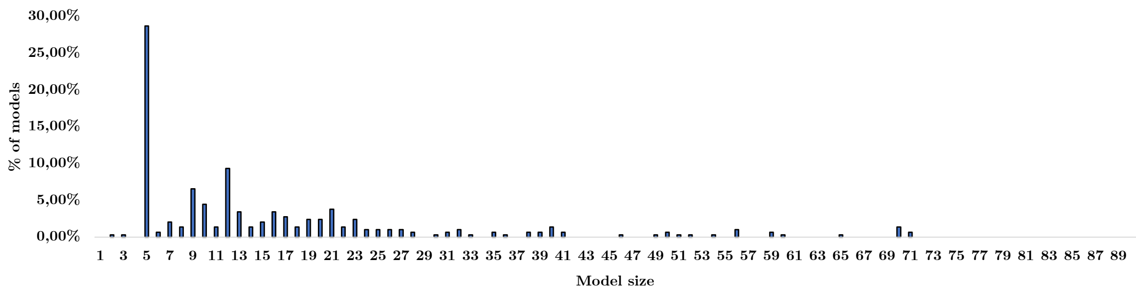
[20]`https://drive.google.com/drive/folders/1Ol6QBXVEs_jZOvQ6o2oQlWyqesHSJqfL` - Accessed on 9 March 2023



(a) Process collaboration



(b) Choreography



(c) Conversation

Figure 8: Percentage of models by model size

Rank (*r*)	Element (*e*)	Occur.	# of Models	Prob. Distr.
1	sequence flow	658 951	37 853	97,40%
2	normal task	262 616	28 450	73,21%
3	message flow	66 820	12 340	31,75%
4	lane	65 215	20 257	52,12%
5	end event	63 414	31 744	81,68%
6	exclusive gateway	56 200	15 951	41,04%
7	expanded pool	46 146	22 065	56,78%
8	start event	42 707	30 538	78,58%
9	parallel gateway	35 864	10 773	27,72%
10	association undirected	33 149	7 502	19,30%
11	user task	29 020	8 032	20,67%
12	text annotation	28 218	7 515	19,34%
13	association data input	26 028	4 878	12,55%
14	intermediate catch message event	23 466	7 996	20,57%
15	exclusive gateway (No Marker)	22 628	6 190	15,93%
16	association data output	20 730	5 931	15,26%
17	data object	19 338	5 228	13,45%
18	start message event	14 719	8 837	22,74%
19	collapsed subprocess	12 778	4 832	12,43%
20	intermediate throw message event	12 560	4 366	11,23%
21	conditional flow	12 239	2 679	6,89%
22	data input	11 882	3 175	8,17%
23	service task	11 571	3 833	9,86%
24	data output	9 749	3 477	8,95%
25	intermediate catch timer event	9 419	5 891	15,16%
26	call activity	6 553	1 716	4,42%
27	end message event	6 400	3 657	9,41%
28	send task	6 190	1 954	5,03%
29	data store	6 116	3 026	7,79%
30	event based gateway	6 065	4 569	11,76%
31	inclusive gateway	5 158	2 229	5,74%
32	script task	4 658	2 126	5,47%
33	intermediate event	4 152	1 688	4,34%
34	end terminate event	4 088	2 729	7,02%
35	association unidirectional	3 992	1 630	4,19%
36	expanded subprocess	3 947	1 826	4,70%
37	collapsed pool	3 864	2 591	6,67%
38	receive task	3 720	1 554	4,00%
39	manual task	3 141	1 077	2,77%
40	message	2 859	949	2,44%
41	group	2 472	1 004	2,58%
42	intermediate boundary timer event	2 351	1 474	3,79%
43	intermediate catch conditional event	2 269	1 290	3,32%
44	start timer event	2 048	1 540	3,96%
45	default flow	2 045	845	2,17%
46	intermediate boundary error event	1 829	1 246	3,21%
47	business rule task	1 374	888	2,28%
48	intermediate boundary message event	1 300	689	1,77%
49	intermediate throw link event	1 109	667	1,72%
50	intermediate catch signal event	1 072	570	1,47%
51	expanded event subprocess	948	460	1,18%
52	end error event	938	676	1,74%
53	start signal event	914	467	1,20%
54	start conditional event	896	744	1,91%
55	end cancel event	867	597	1,54%
56	intermediate catch link event	849	580	1,49%
57	intermediate throw signal event	668	390	1,00%
58	complex gateway	615	421	1,08%
59	association bidirectional	614	278	0,72%
60	intermediate boundary compensate event	515	320	0,82%

(a) Process collaboration

Rank (*r*)	Element (*e*)	Occur.	# of Models	Prob. Distr.
1	choreography participant	7 112	522	96,13%
2	sequence flow	6 485	517	95,21%
3	choreography task	3 418	527	97,05%
4	message	2 350	368	67,77%
5	exclusive gateway	907	341	62,80%
6	end event	740	458	84,35%
7	start event	484	447	82,32%
8	association undirected	376	122	22,47%
9	text annotation	294	60	11,05%
10	collapsed subchoreography	209	98	18,05%
11	parallel gateway	188	95	17,50%
12	event based gateway	117	79	14,55%
13	intermediate catch timer event	101	52	9,58%
14	expanded subchoreography	48	39	7,18%
15	default flow	46	32	5,89%
16	intermediate catch message event	43	16	2,95%
17	conditional flow	38	9	1,66%
18	association unidirectional	35	10	1,84%
19	group	34	10	1,84%
20	choreography participant multiple	31	18	3,31%
21	end terminate event	22	14	2,58%
22	start timer event	21	18	3,31%
23	intermediate throw link event	21	14	2,58%
24	intermediate boundary timer event	18	7	1,29%
25	choreography task loop	17	11	2,03%
26	intermediate catch link event	16	14	2,58%
27	intermediate catch signal event	16	8	1,47%
28	inclusive gateway	15	7	1,29%
29	intermediate catch multiple event	15	8	1,47%
30	start conditional event	13	13	2,39%
31	intermediate throw event	12	7	1,29%
32	choreography task parallel MI	12	9	1,66%
33	choreography task sequential mi	8	1	0,18%
34	intermediate catch conditional event	7	4	0,74%
35	intermediate boundary conditional event	7	3	0,55%
36	complex gateway	6	5	0,92%
37	collapsed subchoreography parallel MI	5	4	0,74%
38	collapsed call subchoreography	3	3	0,55%
39	expanded subchoreography parallel MI	3	3	0,55%
40	manual task	2	1	0,18%
41	start multiple event	2	2	0,37%
42	intermediate boundary message	2	2	0,37%
43	intermediate boundary cancel event	2	2	0,37%
44	user task	1	1	0,18%
45	exclusive gateway (No Marker)	1	1	0,18%
46	start signal event	1	1	0,18%
47	expanded subchoreography sequential MI	1	1	0,18%

(b) Choreography

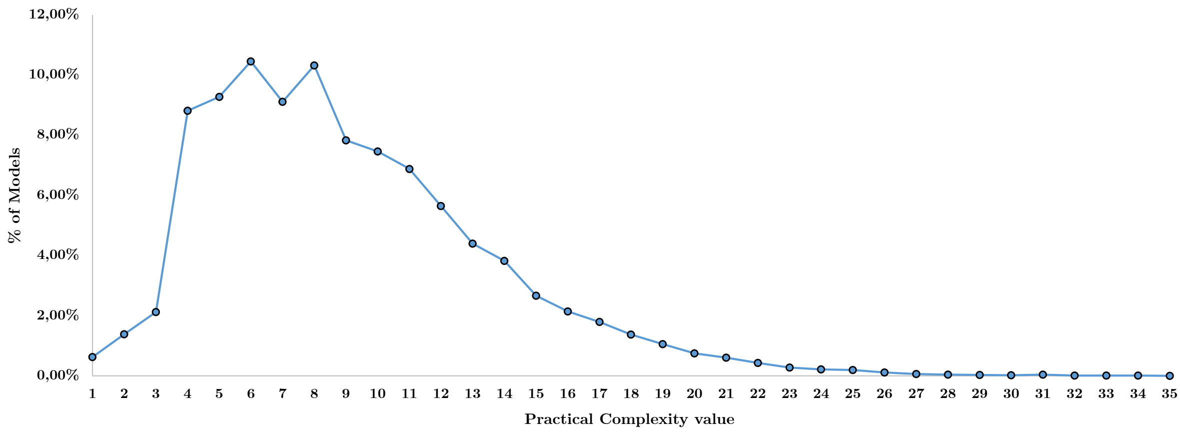
Rank (*r*)	Element (*e*)	Occur.	# of Models	Prob. Distr.
1	conversation link	2 244	285	98,28%
2	collapsed pool	1 346	284	97,93%
3	conversation	927	225	77,59%
4	message flow	339	50	17,24%
5	conversation subprocess	158	113	38,97%
6	text annotation	83	27	9,31%
7	group	81	18	6,21%
8	association undirected	58	21	7,24%
9	collapsed pool multiplicity	15	9	3,10%

(c) Conversation

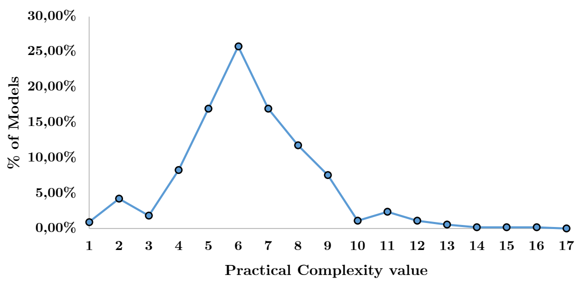
Table 3: Use of BPMN elements

Distribution of BPMN elements over models. Table 3 reports the number and the percentage of models including a given element (notice that the percentage is calculated over the total amount of analyzed models). In the process collaboration models, the most used elements are six: sequence flow, end event, start event, normal task, expanded pool, and lane. We recall that these elements all belong to the BPMN core set. In the choreography models, the most used elements are seven: sequence flow, choreography task, choreography participant, end event, start event, message, and exclusive gateway. The most used elements in the conversation model are conversation link, collapsed pool, and conversation. Among the elements that are less or never used, we found: call activities, ad-hoc subprocesses, and in general, all the elements with compensation, loop, and multiplicity markers. In particular, the overall results suggest that typed and marked elements are rarely used regardless of the model types (e.g., process collaboration, choreography, conversation). Moreover, for process collaboration models, the results reported that six elements, namely *sequence flow* 97.40%, *end event* 81.68%, *start event* 78.58%, and *normal task* 73.21%, and *expanded pool* 56.78%, and *lane* 52.12% are present in the majority of the models (+50%). Concerning choreography models, the elements that are more present in the models are *choreography task* 97.05%, *choreography participant* 96.13%, *sequence flow* 95.21%, *end event* 84.35% and *start event* 82.32%. For what concerns conversation models, there are three main elements almost always present: *conversation link* (98.28%), *collapsed pool* (97.93%), *conversation* (77.59%).

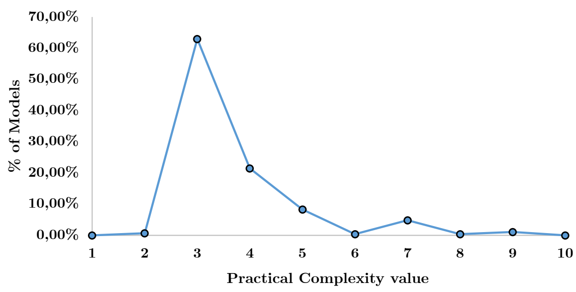
Variety of BPMN elements over models. We calculated the practical complexity for each BPMN model to measure the variety of the elements. Figure 9 shows the practical complexity of process collaboration, choreography, and conversation. 84.14% of collaboration models have a practical complexity between 4 and 14 elements. A process collaboration model is designed using an average of 8 elements. 87.29% of the choreography models have a practical complexity between 4 and 9 elements. A choreography model is designed using an average of 6,5 different elements. 92.73% of conversation models have a practical complexity between 3 and 6 elements. A conversation model is designed using an average of 3,5 different elements.



(a) Process collaboration



(b) Choreography



(c) Conversation

Figure 9: Practical complexity of BPMN models

4.2 Combined Use of BPMN Elements

Focusing on detecting possible relations between BPMN elements, we analyzed pairs and then groups of BPMN elements to find those most frequently used.

Correlation between pairs of BPMN elements. With this analysis, we inspected possible correlations in the usage of pairs of BPMN elements.[21] In Table 4, we report only the most significant pairs of elements that presented a positive correlation ($\rho \geq 0.60$). It is worth noticing that none of the pairs presented an inverse correlation. Considering process collaboration models, the *association data input* and the *data input* exhibit the strongest correlation, with a coefficient of $\rho = 0.96$. For what concerns choreography models, the strongest correlation is obtained by the pair formed by *choreography task* and *choreography participant* ($\rho = 0.89$). In conversation models, the strongest correlation is obtained by the pair formed by *conversation link* and *conversation element* ($\rho = 0.95$).

[21]Correlation does not refer to BPMN elements that are necessarily connected but that are both present in a model.

Element one	Element two	ρ
Process collaboration		
association data input	data input	0.96
association data output	data output	0.81
text annotation	association undirected	0.78
sequence flow	task	0.76
receive task	send task	0.75
intermediate throw link event	intermediate catch link event	0.72
end event	start event	0.65
association data input	data output	0.64
association data input	association data output	0.63
data input	data output	0.63
start event	expanded sub-process	0.61
data input	association data output	0.61
expanded pool	lane	0.60
Choreography		
choreography task	sequence flow	0.89
intermediate catch link event	intermediate throw link event	0.87
choreography task	choreography participant	0.87
choreography participant	sequence flow	0.85
sequence flow	exclusive gateway	0.83
text annotation	association undirected	0.80
choreography task	exclusive gateway	0.68
sequence flow	end event	0.61
Conversation		
conversation link	conversation element	0.95
association undirected	text annotation	0.88
group	collapsed pool	0.86
message flow	collapsed pool	0.73
group	message flow	0.71

Table 4: BPMN elements pairs correlation

Combination frequency of BPMN elements. After analyzing the correlation between pairs of BPMN elements, we focused on combinations of BPMN elements to find those most frequently used. The combinations are reported through Venn diagrams. We used dashed lines and colors to indicate different combinations.

We started by analyzing combinations within the core notation set that groups elements by families (e.g., exclusive, inclusive, and parallel gateways are grouped under the family named gateways). Figure 10 report the Venn diagram for the core notation. The most frequent combination is formed by activities and sequence flows in 93% of the models. If we add events, the resulting combination occurs in 91% of the models. The combination that considers activities, sequence flows, and events is the most frequent, and we can refer to it as the core combination. Adding gateways to the core combination results in a combination that occurs in 66% of models, respectively. If we add pools and lanes to the core set, the resulting combination appears respectively in 54% and 48% of the models. Adding message flows and

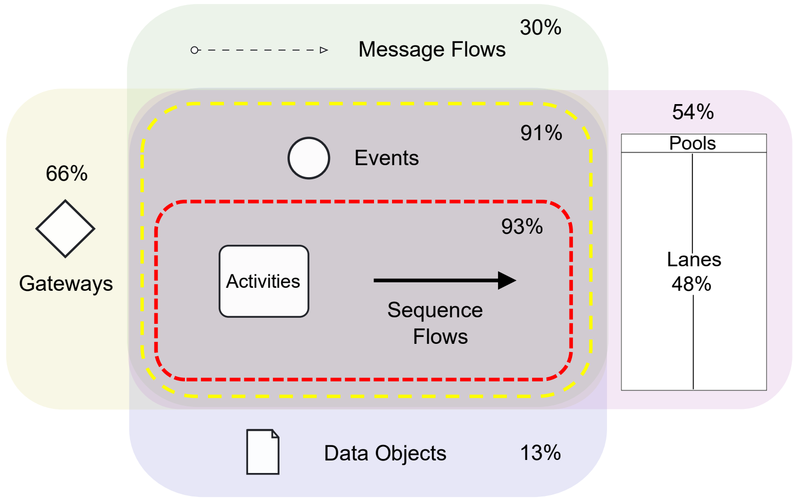
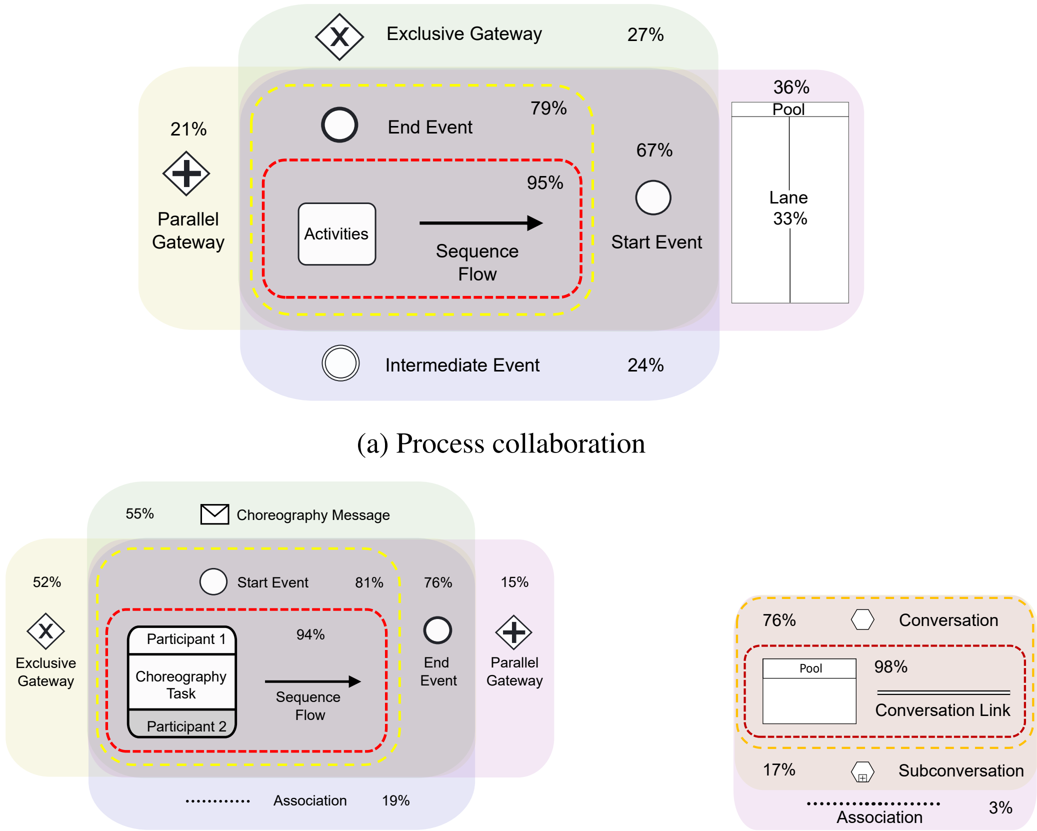


Figure 10: Combinations of BPMN core elements

data objects to the core set results in combinations that occur respectively in 30% and 13% of the models.

Regarding the most used combination of elements for process collaboration, choreography, and conversation, we report the corresponding results in Figure 11. The Venn diagram in Figure 11(a) depicts the most used combinations of elements for process collaboration models. The most frequent combination of elements is formed by activities and sequence flows, with a percentage of 95% models reporting such a combination. Adding the end event, the percentage of models that reports such a combination decreases to 79%. By adding the start event, the resulting combination is present in 67% of the models. Figure 11(b) shows the most used combinations of elements for choreography models. The most frequent combination of elements comprises choreography task and sequence flow, present in 94% of the models. These two types of elements are used in combination with the start event in 81% of the models. Adding the end event results in a combination present in 76% of the models. Choreography task, sequence flow, start event, and end event elements form what we can consider the core notation for choreography models. Finally, for the conversation models, the most used combinations of elements are reported in Figure 11(c). Most models (98%) include a pool and a conversation link, while 76% of the models also include the conversation element.



(a) Process collaboration

(b) Choreography

(c) Conversation

Figure 11: Most popular combinations of BPMN elements by type of model

4.3 BPMN Syntactic Validation

We inspected BPMN models to obtain information regarding Syntactic Errors (SE) affecting the design of a BPMN model. Table 5 overviews the number of syntactically correct and incorrect models grouped by the model type.

We found sixteen different errors occurring in the entire set of analyzed models. Table 6 lists those errors ordered from the most occurring to the least. The most frequent errors are raised when an undefined type definition is used for an element (SE 1) (e.g., an element is not declared in the BPMN standard schema) and when a mandatory attribute is missing (SE 2) (e.g., a connecting object does not have associated any target element, *targetRef* attribute is missing). Frequent errors in XML validation include unexpected attribute values outside their schema definition (SE 3) and duplicated ID values that should be unique (SE 4). Other common errors are the incorrect assignment of data types to attribute values (SE 5) and missing namespace definitions for XML nodes (SE 6).

Models	Process collaboration	Choreography	Conversation
Syntactically Correct	29 112 (75%)	306 (56%)	217 (75%)
Syntactically Incorrect	9 751 (25%)	237 (44%)	72 (25%)

Table 5: Number of syntactically correct and incorrect models

ID	Description	# of Errors
SE1	Undefined type definition is used for an element	34 330
SE2	A mandatory attribute is missing in the declaration of an XML node	29 211
SE3	The value of an attribute is different from those it could assume following the schema definition	8 185
SE4	Multiple occurrences of an ID which must be unique	5 853
SE5	The namespace definition of an XML node is missing	3 534
SE6	Wrong data type of an XML node attribute	3 043
SE7	The content of the element does not match the element type definition or pattern	729
SE8	A corresponding attribute "ID" is not present for attribute "IDREF"	492
SE9	A child element is not expected	339
SE10	Missing one or more mandatory child for the XML node	293
SE11	Wrong attribute name in an XML node	201
SE12	Invalid prefix of the namespace	31
SE13	Incomplete informations in choreography elements	30
SE14	The element is not defined in the namespace declared	18
SE15	The value of an attribute is not in the enumerations as is defined in the schema	3
SE16	Element not presented in the defined namespace	1

Table 6: Error types extracted in BPMN models

Table 7 depicts the most common syntactical errors divided by the model type. Based on the number of models analyzed, choreography models include the most errors on average.

Elements and syntactic violations. We identified 47 different BPMN elements linked to syntactic errors. The errors mainly related to BPMN elements are SE2, SE3, SE5, and SE6. Table 8 reports the elements affected more frequently by errors. The elements with the highest percentage of errors are message flows (3.69%) and associations (3.42%). They introduce errors mainly related to SE2, meaning that such connecting objects are often designed without a source or target element. While the other elements that are mainly related to errors are data input (2.30%) and call activities (2.19%), mostly affected by error SE3, which states that the element definition presents a value for an attribute is not coherent with the schema definition. Although a high amount of SE2 and SE3 errors is related to the sequence flow element, the total amount of sequence flows is so high that the error distribution is low (0.83%), the same can be said for tasks and SE3 errors.

Model Type	# of models per errors				Analyzed	Error
	SE1	SE2	SE3	SE4	model	distribution
Process collaboration	701	6381	605	392	38863	25%
Choreography	0	215	19	19	543	44%
Conversation	0	242	3	3	289	25%

Table 7: Syntactic errors and their distribution considering model type

BPMN Element	Errors by ID				# of Elements	Error distribution
	SE2	SE3	SE5	SE6		
sequence flow	**5 512**	**1 816**	5	21	**658 951**	0.83%
task	0	**1 093**	12	20	**262 616**	0.42%
association	**2 966**	121	0	9	84 982	**3.42%**
exclusive gateway	0	179	1	2	79 737	0.22%
message flow	**2 470**	112	0	7	66 820	**3.69%**
lane	0	193	2	0	65 215	0.29%
end event	0	172	2	16	63 410	0.29%
start event	0	192	0	20	42 707	0.49%
user task	0	283	0	27	29 020	1,06%
data object	0	25	0	0	19 338	0.12%
data input	0	274	5	0	11 882	**2.30%**
service task	0	132	1	10	11 571	1.23%
call activity	0	143	0	1	6 553	**2.19%**

Table 8: Errors distribution for BPMN elements

Modeling tools and syntactic violations. We inspected possible relations between modeling tools and syntactic errors. Table 9 reports the errors about the most adopted modeling tools, highlighting the most significant values. The set of models designed with jBPM (88.80%), GenMyModel (39.63%), Signavio (26.54%), and Eclipse (24.10%) are those presenting the highest percentage of errors. Especially, models designed with jBPM presented more occurrences of the SE1 error, while those designed with Signavio and GenMyModel presented more occurrences of the SE2 error.

Modeling Tools	Errors by ID				Analyzed models	Error distribution
	SE1	SE2	SE3	SE4		
Signavio	0	**16 168**	**1 248**	**1 211**	**22 019**	**26.54%**
GenMyModel	0	**7 230**	**3 827**	**3 375**	7 911	**39.63%**
Activiti	0	3	58	18	2 073	2.65%
Camunda	0	5	63	61	1 698	2.00%
ProM	0	0	0	0	830	0.00%
jBPM	**26 569**	33	**1 141**	34	598	**88.80%**
bpmn-js	0	2	6	1	398	6.28%
Yaoqiang	0	0	12	12	319	3.13%
Eclipse	20	0	75	13	166	**24.10%**
Flowable	0	0	0	0	52	0.00%

Table 9: Syntactic errors and their distribution considering modeling tools

4.4 BPMN Good Modeling Practices Violation

We report the analysis results to check whether the harvested models violate good modeling practices. We investigated possible relations between violating good modeling practices and using specific BPMN elements or modeling tools. The modeling practices predicate on different aspects of a model, such as the usage of the BPMN syntax, the assignment of proper labels to BPMN elements, the arrangement of BPMN elements, and the model's appearance. Table 10 reports evaluated good modeling practices. The first good modeling practices (G1-G5) predicate on general aspects of a BPMN model. In particular, the most violated are G1 and G2, which suggest avoiding overlapping elements (49.82%) and minimizing model size (45.36%).

Elements and good modeling practice violations. The rest of the reported good modeling practices (G6-G32) relate to specific families of BPMN elements: activities, events, gateways, connecting objects, swimlanes, and artifacts. Considering activities, it is worth mentioning that the modeling practice: *G6 - Provide activity descriptions* is violated in most models (95.96%). Concerning events, the most occurring violations are *G8 - Labeling start and end events* (58.89%) and *G9 - Use start and end events explicitly* (33.09%). Referring to gateways, *G15 - Labelling XOR gateways* (46.31%) and *G16 - Use explicit gateways* (35.52%). Concerning flows, we discovered a high percentage of violations for *G22 - Use default flows* (51.17%), and *G23 - Use linear sequence flows* (22.18%). Concerning the swimlanes, the most violated modeling practices are *G27 - Labelling lanes* (31.07%) and *G28 - Use pools consistently* (31.07%). Finally, artifacts elements present violations concerning *G31 - Labelling data objects* and *G32 - Associate data objects consistently*.

Scope	ID	Modeling Practice	Violation
General	**G1**	Avoid overlapping elements	49.82%
	G2	Minimize model size	45.36%
	G3	Keep a standard format	26.27%
	G4	Use a consistent process orientation	15.67%
	G5	Document minor details	5.91%
Activities	**G6**	Provide activity descriptions	95.96%
	G7	Labelling activities consistently	34.40%
Events	**G8**	Labelling start and end events	58.89%
	G9	Use start and end events explicitly	33.09%
	G10	Labelling message events	24.54%
	G11	Use end events consistently	22.29%
	G12	Labelling events	21.05%
	G13	Use start events consistently	8.51%
	G14	Restrict usage of terminate end event	5.29%
Gateways	**G15**	Labelling XOR gateways	46.31%
	G16	Use explicit gateways	35.52%
	G17	Mark exclusive gateways	14.64%
	G18	Labelling AND gateways consistently	8.58%
	G19	Minimize inclusive OR gateways	5.08%
	G20	Labelling converging gateways	4.77%
	G21	Use meaningful gateways	3.64%
Connecting objects	**G22**	Use default flows	51.17%
	G23	Use linear sequence flows	22.18%
	G24	Split and join flows consistently	11.73%
	G25	Use message flows	10.56%
	G26	Use linear message flows	4.96%
Swimlanes	**G27**	Labelling lanes	31.07%
	G28	Use pools consistently	24.20%
	G29	Labelling pools	15.81%
	G30	Use lanes consistently	9.13%
Artifacts	**G31**	Labelling data object	5.99%
	G32	Associate data objects consistently	3.77%

Table 10: Good modeling practices and their percentage of violation

Modeling tool	Modeling practice							Analyzed models
	General			Activities	Flows			
	G1	G3	G4	G6	G22	G23	G26	
Signavio	**59.09%**	**49.07%**	**17.86%**	**97.63%**	**67.20%**	**26.84%**	**20.32%**	12 661
GenMyModel	**64.41%**	1.32%	0.12%	**97.97%**	**60.25%**	0.00%	0.76%	4 327
Activiti	**33.57%**	9.03%	**26.68%**	**94.81%**	**25.59%**	**38.40%**	0.15%	2 005
Camunda	**27.66%**	4.74%	6.20%	**93.98%**	**36.23%**	8.02%	3.47%	1 645
ProM	**99.88%**	0.00%	0.00%	**100.00%**	**54.10%**	**99.88%**	0.00%	830
bpmn-js	17.52%	4.31%	4.58%	**88.68%**	**24.53%**	4.58%	2.70%	371
Yaoqiang	15.00%	**18.33%**	3.00%	**72.33%**	12.00%	**50.33%**	6.00%	300
Eclipse	11.11%	0.79%	2.38%	**88.10%**	7.14%	6.35%	0.79%	126
jBPM	11.94%	2.99%	1.49%	**92.54%**	4.48%	4.48%	0.00%	67
Flowable	11.76%	**35.29%**	9.80%	**92.16%**	**56.86%**	**23.53%**	0.00%	51

Table 11: Percentage of good modeling practices violations considering modeling tools

Modeling tools and good modeling practices violations. We inspected possible relations between modeling tools and good modeling practice violations. Table 11 reports the main relations we discovered, highlighting the most significant values. In particular, models designed with Signavio, GenMyModel, and Activiti present more modeling practice violations than others. The modeling practice *G6 - Provide activity descriptions* appears to be the most violated independently from the modeling tool used, while *G3 - Keep a standard format* and *G4 - Use a consistent process orientation* are not violated by models designed with ProM. Models designed with ProM, jBPM, and Flowable present no violations concerning the modeling practice *G26 - Use linear message flows*.

5 Discussion

In this section, we discuss the results obtained from the conducted analyses over the identified collection to answer the research questions.

5.1 BPMN Elements (RQ.1)

To answer ***"RQ.1 How many and which notation elements are used to design BPMN models?"*** we analyzed: the number, the occurrences, the type, the distribution, and the variety of BPMN elements used.

Concerning the ***amount of BPMN elements used in a model***, it results that 50.86% of the process collaboration models are designed with less than 34 elements. Most choreography models (51.20%) are designed with less than 25 elements. Finally, 55.36% of conversation models are designed with less than 13 elements. What surprised us was the presence of extremely small and extremely big models (i.e., models made of only a few elements or with thousands of elements). Models with small elements can be

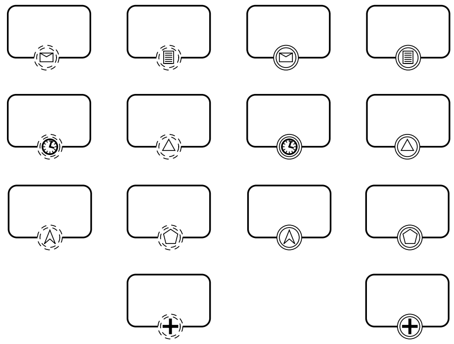


Figure 12: Process model without sequence flows

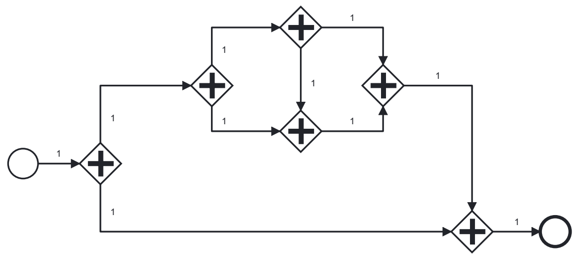


Figure 13: Process model without activities

regarded as "*testing models*". Users may have designed such models while practicing with a modeling tool or while testing BPMN-related tools (many of them come from the GitHub collection). Inspecting models with an extremely high amount of elements, we noticed three types of models: models that are difficult to visualize but that represent a realistic process[22]; models that have been hierarchically structured using sub-processes which elements are stored in the same BPMN model; spaghetti-like models usually discovered from the application of process mining techniques.

Concerning the analysis conducted, we inspected the number of occurrences of each element over the harvested models to determine ***which notation elements are actually used*** **in designing BPMN models.** We found that despite the process collaboration model's theoretical complexity of 244, 59 elements are never used. Concerning choreography models, which theoretical complexity is 91, 45 elements are never used. Considering conversation models, which theoretical complexity is 11, two elements are never used.

Analyzing the ***distribution of elements***, we discovered some models (2.60%) that did not present any sequence flow, which is the element used to connect the other BPMN elements. We looked into those models, for which we report an example in Figure 12. We can conclude that they do not correspond to realistic business process representations; they might have been used to conduct tests over BPMN tools. In addition, despite a business process consisting of at least one activity (Weske, 2019, p. 5) (Dumas, Rosa, Mendling, & Reijers, 2018, p. 6), we found 896 models that do not present one. During our manual analysis, we observed a common trend in the design of models that focus only on the control flow structure, as seen in Figure 13, or those that rely heavily on events rather than activities, as illustrated by the model in Figure 14. Some of those models are reported in scientific papers to describe approaches predicating only on process aspects.

[22] A model with over three hundred elements available on RePROSitory: `https://pros.unicam.it:4200/guest/modelDetails/24_1552924500764_425164153152522` - Accessed on 9 March 2023

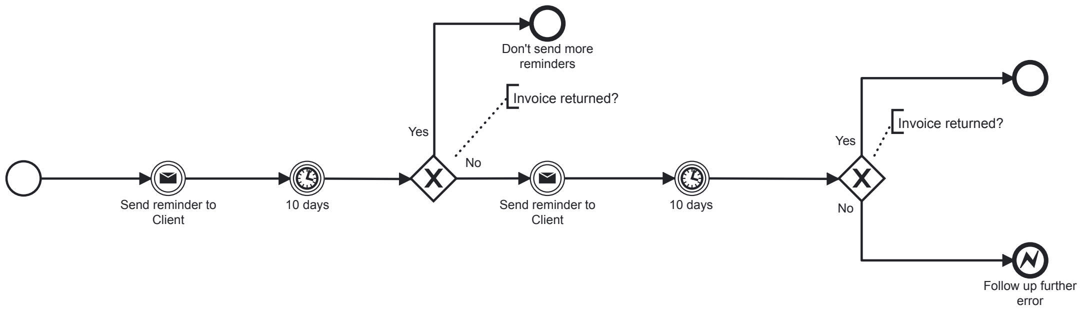


Figure 14: Process model without activities

Finally, considering the analysis on the ***variety of elements used***, why some are barely or never used cannot be determined a priori. It may be because some process concepts are not frequently used in real-world scenarios or because industrial models are not disclosed to the public. Designers with different backgrounds and application domains usually design different BPMN models that may require different notation elements (e.g., training and academic models may differ from those used in organizations). It is worth investigating further, requiring additional empirical studies and direct contact with BPMN practitioners and industrial partners. Finally, it is also worth noting that none of the analyzed BPMN models reported elements belonging to different modeling subsets, meaning no "combined models" were found. None of the modeling tools used to design the analyzed model can support the design of combined models.

5.2 Combinations of BPMN Elements (RQ.2)

To answer ***"RQ.2 Which BPMN elements are used in combination?"*** we analyzed the correlation between pairs of BPMN elements and groups of elements that are frequently used in combination.

Concerning ***correlations between pairs of BPMN elements***, our results show that some elements are highly correlated in designing BPMN models. The strongest correlations pairs of elements for process collaboration models are those related to: data input/output and association data input/output, which indicate whether an artifact is an input or output of a process; sequence flow and task, since both are fundamental elements in the design of a business process model. Additionally, receive task and send task are used to cover the process communication aspects, while the end event and start event allow for the initiation and completion of the process. For choreography models, the strongest correlation pair of elements is composed of choreography task and sequence flow, allowing to link them. Conversation models' strongest correlation pair of elements comprises conversation link and conversation since they represent the communication between process participants. No inverse correlation has been found, meaning that overall there is no clear evidence of alternate use of elements.

Considering the ***combination frequency of BPMN elements***, our findings show that BPMN models are designed around core sets of elements for each type of model that can be designed with BPMN, as illustrated in Figure 11. The core elements in process collaboration models comprise task, sequence flow, start event, and end event. The core set for choreography models comprises choreography task, choreography participant, sequence flow, start event and end event for the choreography type, and pool. The core set for conversation models comprises conversation links and conversation elements for the conversation type.

5.3 Syntactic Violations (RQ.3)

To answer ***"RQ3. Which syntactical errors occur more frequently, and how are they related to BPMN elements and modeling tools?"*** we analyzed syntactic errors in the harvested models, as well as relations with the BPMN elements and the modeling tools used.

The study we conducted shows that among the analyzed models, 10,060 are syntactically incorrect. More precisely, 25% of the process collaborations, 44% of the choreographies, and 25% of the conversations present syntactic errors. Although choreographies and conversations are designed less often than process collaborations, they present a considerable amount of errors. Among the ***syntactic errors*** the most frequent error (SE1) is raised when an element is used but is not defined in the standard BPMN schema, and the schema where the element is defined is not present in the standard BPMN schema. The second most frequent error (SE2) is raised due to an element that, by definition, must have an attribute associated, but such an attribute is missing. Figure 15 depicts an incorrect BPMN model that contains three syntactical errors due to the absence of the attribute required for linking the sequence flow element. Listing 1 provides an example where the "targetRef" attribute is absent in the XML node of the sequence flow. Another frequent error (SE3) occurs when a different value from those specified in the schema is assigned to an attribute of an element. In some models, we also detected many errors related to duplicated ID values used to identify BPMN elements that should instead be unique (SE4).

```
<bpmn:serviceTask id="Activity_18308bc" name="Send Reminder">
 <bpmn:outgoing>Flow_1x8xe15</bpmn:outgoing>
</bpmn:serviceTask>
<bpmn:sequenceFlow id="Flow_1x8xe15" sourceRef="Activity_18308bc"/>
```

Listing 1: XML definition of a sequence flow without the "targetRef" attribute.

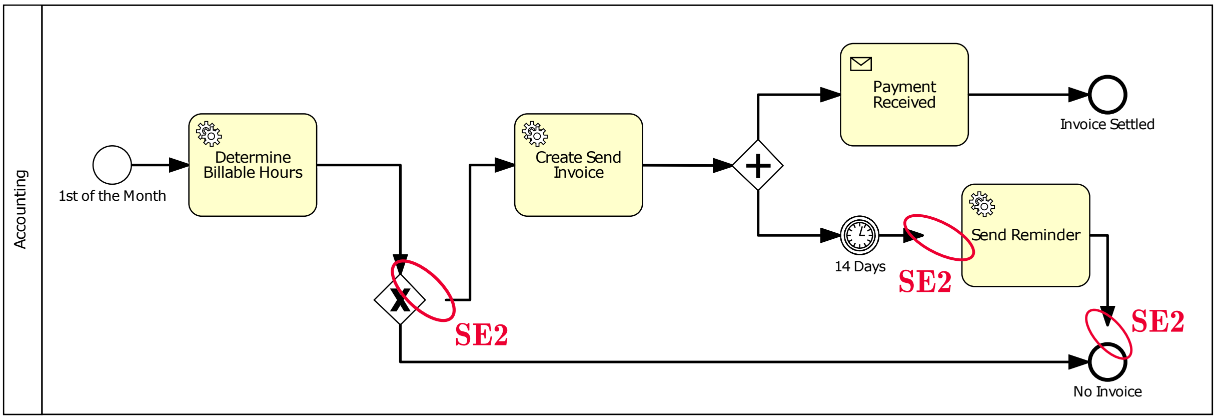


Figure 15: Syntactically incorrect model presenting sequence flows without target elements

From our investigation on ***possible relations between syntactic errors and BPMN elements used*** we noticed that despite in some cases syntactic errors that violate the BPMN XML schema could be related to BPMN elements, through the error message that directly refers to the element, clear relations exist only concerning the use of connecting objects that happen to be designed without a proper source or target element. The remaining errors do not appear strongly related to the element used.

From our investigation on ***possible relation between syntactic errors and BPMN modeling tools***, we noticed that the modeling tool's choice clearly impacts the presence of some syntactic errors. Signavio and GenMymodel were the most used modeling tools considering the analyzed models. However, models designed with Signavio (26.54%), GenMyModel (39.63%), jBPM (88.80%), and Eclipse (24.10%) presented a higher error distribution. Models designed with jBPM present an extremely high amount of SE1 errors. Such models are enriched with elements that are related to the Business Rules Management System called Drools, which can be integrated into the jBPM modeling tool, but they are defined in a different schema (the Drools schema), and when validated only with the BPMN XML schema they cause syntactic errors. The most frequent error in models designed with Signavio and GenMyModel corresponds to the missing mandatory XML node attributes (SE2). As mentioned, many of those errors are related to connecting objects that are designed without properly linking to a source or target element. This is a design that some modeling tools allow. The other modeling tools do not allow the design of connecting objects that are not properly linked to their source and target element, therefore they present low or close to zero SE2 errors.

Syntactic errors in BPMN models, such as SE3 and SE4, may occur when users assign attribute values to BPMN elements using modelers. For example, customized ID values that do not correspond to valid qualified names (SE3) or ID values that are already assigned to other elements (SE4) can cause errors in the model. Referring to SE4, it is not clear why they are so present in models designed with Signavio and GenMyModel since the most recent version of the tools does not allow customized IDs. This might be related to the previous version of the modelers that supported such behavior. This especially is true for models designed with the previous version of the Eclipse modeler that presented a known bug[23] fixed in the most recent versions.

Finally, it is worth noticing that most of the latest versions of BPMN modeling tools do not allow the design of models with syntactic errors (e.g., not allowing the design of sequence flows with no source or target). This is the case of Camunda, bpmn-js, GenMyModel, Activiti, ProM, jBPM, Yaoqiang, Eclipse, and Flowable. Signavio, on the other hand, allows to design, and save models that contain syntactic errors, warning first the user about such errors.

5.4 Modeling Practices (RQ.4)

To answer "***RQ4 Which good modeling practices are violated, and how are they related to specific BPMN elements and modeling tools?***" we analyzed which good modeling practices are violated by the harvested models, as well as possible relations with the BPMN elements and the modeling tools used. It is well known that when followed, the analyzed good modeling practices support the design of understandable business process models (Corradini, Ferrari, et al., 2018). To guide the discussion, Figure 16 reports a model that presents the main violations of good modeling practices. We added the IDs of the modeling practices in correspondence with the points that clearly show their violations so the reader can refer to such a model while reading the following discussion.

Concerning the analysis we conducted on *general good modeling practices*, we found that the most violated are those that suggest designing models without overlapping elements (G1) (e.g., the data object over the event), maintaining the number of elements used under a certain threshold (G2), keep a standard format without the usage of personalized colors (G3) (e.g., the task "Analyze Request" depicts a different color).

Several ***good modeling practices predicate over specific BPMN elements*** and our results show that several of them are often violated. For what concerns *Activities*, it results that they are rarely accompanied by a description (G6) that could provide more detail about the activity itself and help users of that model better understand the process it represents. In addition, designers often do not assign proper labels to activities, i.e., unique labels with one verb and one object (e.g., "Receive request", "Analyze request"),

[23] `https://www.eclipse.org/forums/index.php/t/1024694` - Accessed on 9 March 2023

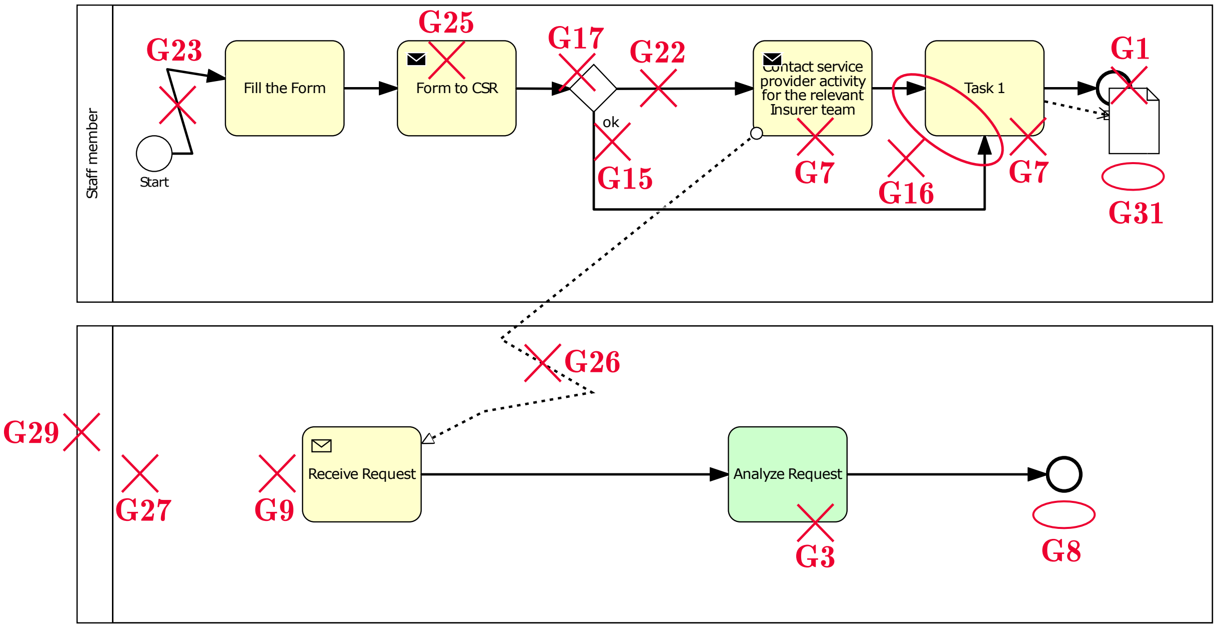


Figure 16: Syntactically valid model presenting good modeling practices violations

but they use fictitious labels (G7) (e.g., *"Task 1"*) or far too detailed labels (G7) (e.g., "Contact service provider activity for the relevant insurer team"). In addition, designers tend to label multiple activities with the same name, a practice admitted only for *Call Activities*.

Regarding *Events*, several models are designed without explicit start and end events (G9) (e.g., a start event before the task "Analyze request" is missing), and when they are represented, they are not properly labeled (G8) (e.g., lack of the label on the end event).

Modeling practices that predicate over the usage of *Gateways* are also frequently violated. BPMN introduces some ambiguities that can allow for the representation of a concept in multiple ways. For instance, the splitting and merging of control flow can be explicitly illustrated by designing gateways or implicitly represented by connecting multiple sequence flow to the same element. However, explicitly reporting gateways can reduce misinterpretations (G16), as illustrated by the missing merging exclusive gateway before the task "Task 1". When using an exclusive gateway to split a flow, giving it a clear label (G15) is important, which should be phrased as a question. Additionally, any sequence flows from diverging gateways should be labeled with their corresponding outcome conditions. On the other hand, exclusive gateways used to join the flow do not require a label. Figure 16 shows the diverging gateway is only labeled with a generic "ok" label, which is not recommended. Some editors enable the modeling of exclusive gateways without markers (G17), such as the one before the "Form to CSR send task" in the example model. However, this could confuse the model user, leading to errors and misunderstandings.

For what concerns *Connecting Objects*, good modeling practices suggest explicitly expressing the default flow after designing exclusive and inclusive gateways (G22), which is missing from the task "Form to CSR". This default flow prevents a process from being stuck at a decision point. In fact, the default flow will always evaluate as true if all the other sequence flow conditions are false. In addition, designing linear sequence flows (G23) and linear message flows (G26) is frequently violated. In particular, these modeling practices can be challenging to manage in models of big size, resulting in distorted and confusing cross-over flows and complicating model comprehension. We highlight such violations in Figure 16. Specifically in the sequence flow following the start event (G23) and in the message flow originating from the task "Contact service provider activity for the relevant insurer team"(G26).

Referring to *Swimlanes*, it emerges that they often lack labeling of their pools (G29) and lanes (G27) elements (as shown in Figure 16 concerning the lower pool). The designers should label lanes and pools to respectively specify the participants in the process as well as their internal roles (e.g., manager, associate), systems (e.g., an enterprise application), or internal departments (e.g., shipping, finance).

Artifacts in BPMN models are frequently subject to practice violations, often related to their labeling (G31). For example, some data object elements lack labels. When multiple instances of the same data object must be represented, a matching label followed by the relevant state in square brackets should be used. This approach ensures that the changes applied by the activities in the process are properly represented.

From our investigation on ***possible relations between good modeling practices violations and BPMN modeling tools***, we noticed that the choice of the modeling tool has a clear impact on some violations while it is not related to others. In particular, we found that models generated by ProM, a tool that automatically creates BPMN models from event logs or through format conversions rather than manual design, often violate modeling practices related to the visual appearance of the model. These violations include elements overlapping (G1), the use of linear sequence flow (G23), and the absence of default flow for exclusive gateways (G22). Nevertheless, of course, models automatically designed have the same format (G3) and use a consistent process orientation (G4). Instead, the modeling practices referring to message flows (G25 and G26) are not violated since the models designed with ProM do not present such BPMN elements.

Models designed with the other modeling tools also present several overlapping elements (G1). This suggests that such modeling practice is violated independently of the adopted tool. Furthermore, models designed using Signavio, Flowable, and Yaoqiang modeling tools violate the best practice of designing models with standard formats (G3) because these tools offer customization options for the appearance of BPMN elements. For instance, users can change labels' background color or font, leading to non-standard visual representations of the elements. Although no modeling tool enforces the design of horizontal models (G4), some of them, such as Signavio or Yaoqiang, allow the user to specify the model orientation so that the model is designed vertically or horizontally. In contrast, others such as Camunda, bpmn-js, Activiti, jBPM, and Flowable allow the design of pools only horizontally, limiting the possibility for users to design vertical process collaborations.

We noticed that independently from the modeling tool and whether it supports the assignment of a description to an activity or not, the modeling practice that suggests assigning a description to an activity (G6) is the most violated. Also, using default flows (G22) is a modeling practice that is often violated but independent of the modeling tool. The modeling practices that suggest using linear sequence flows (G23) and linear message flows (G26) present different percentages of violation based on the modeling tool used. In particular, some modeling tools, such as Signavio, Activiti, Yaoqiang, and Flowable, present more violations than others, in fact, they do not enforce the design of linear connecting objects, but they allow the user also to design "*zig-zag flows*", while others such as Camunda, bpmn-js, jBPM, and Activiti guide the user in designing linear flows, but the user can adjust them also in a non-linear manner adding the number of vertices used to design the flow. GenMyModel and Eclipse, instead, guide the user in designing linear flows and prevent the user from adding multiple vertices to the connecting objects.

6 Related Work

In this section, we compare some research works in the literature targeting statistical analyses concerning the usage of BPMN (zur Muehlen & Recker, 2013) and (Compagnucci et al., 2021). Table 12 reports a summary of the main characteristics of the studies.

The former study, zur Muehlen and Recker (2013), analyzed the usage of BPMN considering the first version of the notation (v1.0). The authors distinguished 50 notation elements; and analyzed 126 models, excluding non-sense and duplicate models. Given that the authors focus on BPMN 1.0, we can only compare our results to process models since the standard's choreography and conversation models have been introduced in version 2.0.

The study by Compagnucci et al. (2021), despite focusing on BPMN 2.0, does not perform a fine-

	Muehlen and Recker	Compagnucci et al.	Our work
BPMN models version	1.0	2.0	2.0
Number of BPMN models analyzed	126	25 590	54 500
Number of BPMN elements considered	50	85	267
Minimum number of models size	0	8	>0
Analysis on different model types	✗	✗	✓

Table 12: Related works characteristics comparison

Analysis Scope	Analysis Type	Muehlen and Recker	Compagnucci et al.	Our work
BPMN elements	Number of BPMN elements used	✓	✓	✓
	Occurrences of specific BPMN elements	✓	✓	✓
	Distribution of of BPMN elements over models	✓	✓	✓
	Variety of BPMN elements over models	✓	✓	✓
Combined use of BPMN elements	Correlation between pairs of BPMN elements	✓	✓	✓
	Combination frequency of BPMN elements	✓	✓	✓
BPMN syntactic validation	Syntactic errors in BPMN models	✗	✗	✓
	BPMN element related syntactic errors	✗	✗	✓
	Modeling tools related syntactic errors	✗	✗	✓
BPMN good modeling practices	Violation of BPMN good modeling practices	✗	✗	✓
	BPMN element related good modeling practices violations	✗	✗	✓
	Modeling tools related good modeling practices violations	✗	✗	✓

Table 13: Related works analyses comparison. Symbol "✓" indicates that the target work has performed the analysis. Symbol "✗" indicates that the target work has not performed the analysis

grain analysis considering BPMN elements as we do in this study. They distinguish just 85 elements against the 267 we included by looking at all the different markers related to elements. The authors analyzed a collection of 25,590 models, filtering out models with a size smaller than eight elements in an attempt to discard toy-example models.

This work focused on BPMN 2.0, analyzing 54,500 models from seven online repositories. We extracted data for 267 different BPMN elements reaching a high level of detail in the analysis of the BPMN notation. In addition, we analyzed the elements of the notation, considering the different types of models that BPMN 2.0 supports (i.e., process collaboration, choreography, and conversation models).

Table 13 compares the analysis conducted in the different studies. All the studies analyze the use of BPMN elements and their combinations. Differently from the others, in our study, we also inspect the syntactic validation of the BPMN models and their adherence to good modeling practices, further analyzing possible relations with the BPMN elements and the modeling tools used.

Referring to the ***number of BPMN elements used*** in the design of a model, zur Muehlen and Recker (2013) did not provide any data. While Compagnucci et al. (2021) present a similar trend to ours, the main difference is that they removed BPMN models with less than eight elements, which in our case count for more than 13% of the models. In the set of models analyzed by Compagnucci et al. (2021) and in ours are some models with an extremely high number of elements (in the order of hundreds and of thousands).

Concerning the ***occurrences and the distribution of notation elements***, it emerges that independently from the version of BPMN, the repository used, and the filtering procedure adopted, the most used elements are the same for all three studies. Such elements are sequence flow, normal task, message flow, lane, end event, exclusive gateway, expanded pool, start event, and parallel gateway. In this study and the study by zur Muehlen and Recker (2013), many text annotations have been detected, which were less present in the models analyzed by Compagnucci et al. (2021).

Concerning the ***variety of BPMN elements*** used in a model, our study, as well as the one by Compagnucci et al. (2021), highlight that a process collaboration model is usually designed using different types of BPMN elements in a range between four and fourteen. In the study by zur Muehlen and Recker (2013), models appear to be designed with different types of BPMN elements in a range between four and twelve. This is interesting since both our study and the one by Compagnucci et al. (2021) target models designed with BPMN 2.0 that presents a theoretical complexity much higher than BPMN 1.0 that was used to design models analyzed by zur Muehlen and Recker (2013). So, despite the standard BPMN 2.0 increasing the number of elements that can be used and, therefore, of concepts that can be represented and modeled, model designers have difficulties leveraging such an extensive set of elements.

Concerning **BPMN element correlations**, zur Muehlen and Recker (2013) detected some negative ones. However, this may indicate that the analyzed models are not representative enough. In fact, we cannot find any reasonable explanation for why a start or an end event should be used alternatively to an exclusive gateway. Regarding positive correlations, all three studies confirm that some notation elements are strongly correlated. For example, the send task and receive task, as well as the pool and message flows, are frequently used together to represent message exchange between two process participants. Moreover, end and start events are commonly used to indicate the beginning and end of a process. Finally, throw events and their related catch events are often combined.

For what concerns the **combined usage of BPMN elements** from all the analyzed works emerges that a de-facto core subset of elements is more often used, which includes: task, sequence flow, start event, and end event. Then other elements such as exclusive gateway, parallel gateway, intermediate event, pool, and message flow are combined with such a subset.

7 Implications

In this section, we report some implications from the analysis of the results grouped by possible interesting targets.

Implication for educators. BPMN teaching activities could be better oriented towards focusing first on those elements and combinations that result in being the most used ones. For instance, teaching first the emerged core sets and introducing later typed and marked elements with specific targeted lectures and examples (Figl, 2017). We suggest educators adopt, during training activities, a tool that prevents the creation of syntactically invalid models (i.e., bpmn-js, Camunda) but also introduces other modeling tools that illustrate the major differences that users may find. We also suggest educators focus their activities not only on explaining the usage of the BPMN notation but also on teaching modeling practices, such as those reported in literature (Corradini, Ferrari, et al., 2018) that, if followed, are proved to improve the understandability of the resulting model and may ease its maintainability. In addition, we suggest BPMN educators adopt an open repository platform (e.g., RePROSitory) for sharing training materials within their courses. In particular, BPMN models accompanied with proper descriptions could help clarify doubts about notation usage, fostering good modeling practices.

Implication for practitioners. We suggest modeling practitioners, especially novices, carefully select the modeling tool to use since some of them provide a form of support that guides the user in designing syntactically correct models without allowing them to insert syntactic errors and also makes it easier to follow good modeling practices. We also encourage practitioners to adopt the good practice of sharing BPMN models with metadata that provides additional information, such as the domain in which the model has been designed, the domain that it represents, or the designer's expertise. This practice of providing additional information about BPMN models enables further analysis of the usage of BPMN.

Implication for researchers. The research community could be guided toward finding why some BPMN elements are rarely or never used. Researchers may try to understand whether their usage depends on the designer's knowledge of the BPMN notation or whether it depends on the domain of the involved business process. To foster this kind of study, we encourage the research community to share BPMN models enriched with metadata that can help distinguish the origin of the models, the business process domain represented, and the designer's background. Other characteristics of BPMN models, such as workflow patterns (van der Aalst, ter Hofstede, Kiepuszewski, & Barros, 2003) and BPMN model metrics

(Recker et al., 2009), could also be investigated.

We want to stimulate the research community in trying to define a way to distinguish realistic business process models from so-called “toy-example” or “testing-models”. In doing so, from the analysis we conducted, we derived some indicators that may suggest the presence of not realistic business process models: *1.* model without any sequence flow (for process collaboration and choreography) and model without conversation link (for conversation models); *2.* process collaboration models without any activity, choreography model without any choreography task, conversation model without any pool; *3.* model size less than two; *4.* model size over three hundred elements; *5.* models presenting only one variety of elements.

In addition, modeling practices that predicate on choreography and conversation models are missing. Further investigation may be conducted to define modeling practices that target choreography and conversation models, providing tools to automatize their evaluation. Furthermore, the research community could benefit from establishing collaborations with companies that may allow them to disclose the investigated BPMN models. To prevent reserved information from being disclosed, models could undergo a process of anonymization while still preserving the utility for some studies, such as for reasoning about the kind of elements and modeling practices used in industries.

Finally, our results can support the development of BPMN-based tools that use machine learning techniques. To provide an example, Antinori et al. (2022) propose BPMN-Redrawer, a tool for redrawing BPMN models from images to .bpmn format using supervised machine learning algorithms for detecting BPMN elements. Our study can provide valuable input indicating the most probable/potential elements according to the frequency used, improving the average precision with which BPMN-Redrawer recognizes them.

Implication for the standardization bodies. The scarce usage of some BPMN elements could indicate a need for a better explanation. New releases of the standard and the documenting materials could include better explanations and examples for using those elements. Some BPMN elements, in particular, have attracted the research community’s attention and have already been discussed, such as the case of the “inclusive gateway”, also referred to as OR-Join, (Christiansen, Carbone, & Hildebrandt, 2010; Corradini, Muzi, Re, Rossi, & Tiezzi, 2022; Dumas, Großkopf, Hettel, & Wynn, 2007). The BPMN 2.0 specification provides a rather detailed but informal description of its semantics. The association of formal semantics to the standard, chosen between the already proposed ones (Corradini, Fornari, Polini, Re, & Tiezzi, 2018; Dijkman, Dumas, & Ouyang, 2008; Houhou, Baarir, Poizat, Quéinnec, & Kahloul, 2022; Kossak et al., 2014; Wong & Gibbons, 2011; Ye, Sun, Song, & Wen, 2008), could be a solution to disambiguate the interpretation of the standard.

It would be useful to investigate the possible categorization and extension of the BPMN elements based on the process application domain to be modeled. Some combinations of elements may be more frequently used in specific domains. For example, Bourr et al. (2021) identified a subset of elements for designing multi-robot systems, leveraging the role of the often overlooked timer, conditional, and signal events. For what concerns possible extensions of the BPMN standard, many research works have already contributed to introducing additional elements (Braun & Esswein, 2014; Compagnucci et al., 2020, 2022; Onggo et al., 2018; Strutzenberger, Mangler, & Rinderle-Ma, 2021; Zarour, Benmerzoug, Guermouche, & Drira, 2020). It would be interesting to study and define a standardization for such BPMN extensions so that the community can refer to the same extended notation when modeling specific domains.

8 Limitations

This section highlights the limitations of this study, referring to the quality of the models (i.e., unrealistic models, missing information), the modeling editor used (i.e., missing notation elements), and the tools used for the analyses (i.e., incompatibility with different types of models).

Some of the considered models show questionable characteristics, such as the absence of activities, the absence of sequence flow, and an extremely low or high number of elements. This may negatively impact results, especially when the number of models is limited. However, since our study was conducted on many models, such characteristics do not affect our results. In addition, we recognize that BPMN models can present custom elements derived from an extension of the standard. We inspected the BPMN tag "*<bpmn:extensionElements>*" (or "*<extensionElements>*") present in the analyzed models, and no additional graphical element has been detected. The majority of the identified extensions were related to additional metadata associated with already available BPMN elements, such as those referring to the coloring of an element.[24] Other metadata relates to tools that can be integrated with BPMN, such as the rule engine Drools.[25] It is also worth noticing that 3,550 models were missing the *exporter* and the *targetNamespace* attributes in their XML file. For this reason, we could not derive any information

[24]The tags *<extensionElements> <signavio:signavioMetaData metaKey="bgcolor" metaValue="#ffffff"/> </extensionElements>*, are used in Signavio to specify the background color of an element.

[25]`https://www.drools.org` - Accessed on 9 March 2023

about the editor used and therefore were classified as undefined. The analyzed models were harvested from different sources and did not present descriptive metadata. Therefore, another limitation of our study is that the analyses were performed on the entire collection regardless of the model's domain and the designer's background. This could affect the applicability of our findings in specific domains, such as industrial or healthcare. Considering the different levels of support of BPMN by modeling tools, the usage of some tools concerning others may, in some cases, impact the notation elements at the disposal of model designers. Therefore, BPMN models created using specific modeling tools may have an impact on the results of the analysis. We investigated this aspect, and the only differences that we found can be reconducted to *Camunda*, *bpmn-js* and *chor-js* that do not provide the possibility to design some specific elements: exclusive gateway without markers, data input and data output, multiple events, parallel multiple events, parallel event-based gateway, and exclusive event based gateway.

Finally, another limitation of our study relates to evaluating good modeling practices that we conducted only on process collaboration models due to the lack of a proper definition and supporting tool for modeling practices over choreography and conversation models.

9 Conclusions

In this paper, we investigated the use of the BPMN notation for the design of process collaboration, choreography, and conversation models. To achieve this objective, we defined four research questions that we answered by analyzing a collection of 54,500 models. The study relies on the functionalities of the BPMN Inspector and BPMN Validator that we developed on purpose. In addition, we used the BEBoP tool, which was already available. In particular, our study emphasizes the difference between the theoretical complexity of the BPMN notation (i.e., the total number of elements in the modeling notation) and the practical complexity (i.e., the number of different types of elements that are used for the design of BPMN models). Considering the combined use of elements, we noticed some elements are highly correlated, while no inverse correlation was detected. We also observed that BPMN models (independently of their type) are designed using a small subset of the notation elements. We explored syntactic errors affecting the models and their relations to using specific BPMN elements and the modeling tools used for designing such models. It emerged that most syntactic errors are related to connecting objects designed with a missing source or target element. Our study shows that the choice of a specific BPMN modeling tool can help designers avoid syntactic errors and influence adherence to certain good modeling practices, especially those referring to the appearance of the model. On the other hand, the violation of good modeling practices related to specific families of elements (i.e., gateways, activities) is not directly related to the choice of the editor used. We also compared our study with related works highlighting differences and similarities. Then, we formulated some suggestions directed to educators, practitioners, researchers, and standardization bodies. In conclusion, despite the results of our study highlighting that some subsets of the BPMN notation are predominant, it does not mean that the elements outside such subsets are to be disregarded. Indeed, in some specific domains, all the elements of the BPMN notation may find their applicability.

References

Antinori, A., Coltrinari, R., Corradini, F., Fornari, F., Re, B., & Scarpetta, M. (2022). BPMN-Redrawer: From Images to BPMN Models. In *International Conference on Business Process Management* (Vol. 3216, pp. 107–111). CEUR-WS.org.

Benesty, J., Chen, J., Huang, Y., & Cohen, I. (2009). Pearson correlation coefficient. In *Noise reduction in speech processing* (Vol. 2, pp. 1–4). Springer.

Bork, D., Karagiannis, D., & Pittl, B. (2020). A survey of modeling language specification techniques. *Information Systems*, *87*.

Bourr, K., Corradini, F., Pettinari, S., Re, B., Rossi, L., & Tiezzi, F. (2021). Disciplined use of BPMN for mission modeling of Multi-Robot Systems. In *Practice of enterprise modeling* (Vol. 3045, pp. 1–10). CEUR-WS.org.

Braun, R., & Esswein, W. (2014). Classification of domain-specific BPMN extensions. In *Conference on the practice of enterprise modeling, LNBIP.* (Vol. 197, pp. 42–57). Springer.

Christiansen, D. R., Carbone, M., & Hildebrandt, T. T. (2010). Formal Semantics and Implementation of BPMN 2.0 Inclusive Gateways. In *Web services and formal methods workshop, LNCS* (Vol. 6551, pp. 146–160). Springer.

Compagnucci, I., Corradini, F., Fornari, F., Polini, A., Re, B., & Tiezzi, F. (2020). Modelling Notations for IoT-Aware Business Processes: A Systematic Literature Review. In *BPM Workshops BP-Meet-*

IoT, LNBIP (Vol. 397, pp. 108–121). Springer.

Compagnucci, I., Corradini, F., Fornari, F., Polini, A., Re, B., & Tiezzi, F. (2022). A systematic literature review on IoT-aware business process modeling views, requirements and notations. *Software and Systems Modeling*, *14*(1), 1–36.

Compagnucci, I., Corradini, F., Fornari, F., & Re, B. (2021). Trends on the Usage of BPMN 2.0 from Publicly Available Repositories. In *International Conference on Perspectives in Business Informatics Research, LNBIP* (Vol. 430, pp. 84–99). Springer.

Corradini, F., Ferrari, A., Fornari, F., Gnesi, S., Polini, A., Re, B., & Spagnolo, G. O. (2018). A Guidelines framework for understandable BPMN models. *Data Knowledge Engineering*, *113*, 129-154.

Corradini, F., Fornari, F., Polini, A., Re, B., & Tiezzi, F. (2018). A formal approach to modeling and verification of business process collaborations. *Science of Computer Programming*, *166*, 35–70.

Corradini, F., Fornari, F., Polini, A., Re, B., & Tiezzi, F. (2019). Reprository: a repository platform for sharing business process models. In *Business Process Management* (Vol. 2420, pp. 149–153). CEUR-WS.org.

Corradini, F., Muzi, C., Re, B., Rossi, L., & Tiezzi, F. (2022). BPMN 2.0 OR-Join Semantics: Global and local characterisation. *Information Systems*, *105*, 101934.

Dijkman, R. M., Dumas, M., & Ouyang, C. (2008). Semantics and analysis of business process models in BPMN. *Information and Software Technology*, *50*(12), 1281–1294.

Dobing, B., & Parsons, J. (2006). How UML is used. *Communications of the ACM*, *49*(5), 109–113.

Dumas, M., Großkopf, A., Hettel, T., & Wynn, M. T. (2007). Semantics of standard process models with or-joins. In *On the Move to Meaningful Internet Systems, LNCS* (Vol. 4803, pp. 41–58). Springer.

Dumas, M., Rosa, M. L., Mendling, J., & Reijers, H. A. (2018). *Fundamentals of business process management*. Springer.

Erickson, J., & Siau, K. (2004). Theoretical and practical complexity of unified modeling language:delphi study and metrics analyses. In *International conference on information systems* (pp.

183–194). AIS eLibrary.

Erickson, J., & Siau, K. (2007). Theoretical and practical complexity of modeling methods. *Communications of the ACM*, *50*(8), 46–51.

Fahland, D., Favre, C., Jobstmann, B., Koehler, J., Lohmann, N., Völzer, H., & Wolf, K. (2009). Instantaneous Soundness Checking of Industrial Business Process Models. In *Business Process Management, LNCS* (Vol. 5701, pp. 278–293). Springer.

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

Genon, N., Heymans, P., & Amyot, D. (2010). Analysing the Cognitive Effectiveness of the BPMN 2.0 Visual Notation. In *Software Language Engineering, LNCS* (Vol. 6563, pp. 377–396). Springer.

Heinze, T. S., Stefanko, V., & Amme, W. (2020). Mining BPMN Processes on GitHub for Tool Validation and Development. In *Business Process and Information Systems Modeling, LNBIP* (Vol. 387, pp. 193–208). Springer.

Houhou, S., Baarir, S., Poizat, P., Quéinnec, P., & Kahloul, L. (2022). A First-Order Logic verification framework for communication-parametric and time-aware BPMN collaborations. *Information Systems*, *104*, 101765.

Kossak, F., Illibauer, C., Geist, V., Kubovy, J., Natschläger, C., Ziebermayr, T., … Schewe, K. (2014). A rigorous semantics for BPMN 2.0 process diagrams. Springer.

Onggo, B. S. S., Proudlove, N. C., D'Ambrogio, A., Calabrese, A., Bisogno, S., & Ghiron, N. L. (2018). A BPMN extension to support discrete-event simulation for healthcare applications: an explicit representation of queues, attributes and data-driven decision points. *Journal of the Operational Research Society*, *69*(5), 788–802.

Petre, M. (2013). UML in practice. In *Conference on software engineering* (pp. 722–731). IEEE Computer Society.

Recker, J., Rosemann, M., Indulska, M., & Green, P. (2009). Business process modeling: A comparative analysis. *Journal of the Association for Information Systems*, *10*, 1.

Siau, K., Erickson, J., & Lee, L. (2005). Theoretical vs. Practical Complexity: The Case of UML. *Database Management*, *16*(3), 40–57.

Strutzenberger, D. V., Mangler, J., & Rinderle-Ma, S. (2021). BPMN extensions for modeling continuous processes. In *Intelligent Information Systems, LNBIP* (Vol. 424, pp. 20–28). Springer.

The Object Management Group. (2011). *The BPMN Core Notation set.* (Online: `https:\\www.omg.org\bpmn\Samples\Elements\Core\BPMN\Elements.htm` (Accessed on 18 May 2022))

van der Aalst, W. M. P., ter Hofstede, A. H. M., Kiepuszewski, B., & Barros, A. P. (2003). Workflow patterns. *Distributed Parallel Databases*, *14*(1), 5–51.

Weske, M. (2019). *Business process management - concepts, languages, architectures*. Springer.

Wong, P. Y. H., & Gibbons, J. (2011). Formalisations and applications of BPMN. *Science of Computer Programming*, *76*(8), 633–650.

Ye, J., Sun, S., Song, W., & Wen, L. (2008). Formal Semantics of BPMN Process Models Using YAWL. In *Intelligent Information Technology Application* (Vol. 2, p. 70-74). IEEEXplore.

Zarour, K., Benmerzoug, D., Guermouche, N., & Drira, K. (2020). A systematic literature review on BPMN extensions. *Business Process Management*, *26*(6), 1473–1503.

zur Muehlen, M., & Recker, J. (2013). How Much Language Is Enough? Theoretical and Practical Use of the Business Process Modeling Notation. In *Advanced Information Systems Engineering, LNCS* (pp. 429–443). Springer.