# Evaluating Support for Implementing BPMN 2.0 Elements in Business Process Management Systems

Carlos Francisco Habekost Dos Santos[1][a], Marcelo Fantinato[2][b],
José Palazzo Moreira de Oliveira[1][c] and Lucineia Heloisa Thom[1][d]

[1]*Institute of Informatics, Federal University of Rio Grande do Sul, Porto Alegre, Brazil*

[2]*School of Arts, Sciences and Humanities, University of São Paulo, São Paulo, Brazil*

*{cfhsantos, palazzo, lucineia}@inf.ufrgs.br, m.fantinato@usp.br*

Keywords: BPMN, BPMS, Implementation, Notational Elements.

Abstract: Business Process Model and Notation (BPMN) provides an extensive set of notational elements, such as activities, events and gateways, which enable the representation of a wide variety of business processes. Business Process Management Systems (BPMSs) can implement BPMN to allow the execution of business processes. However, not all BPMN elements are implemented in a BPMS. The purpose of this paper is to present the results of an evaluation of the BPMN 2.0 elements conducted to identify those whose implementation is supported in BPMSs. We evaluated four BPMSs, comparing the elements implemented in such BPMSs with their respective definitions in the BPMN specification, considering the requirements to implement them. As a result, we found that only 34.18% of the BPMN 2.0 elements are implemented in the investigated BPMSs. We also identified that BPMSs implement BPMN elements only partially, adapting their original definition. In addition to the results of our evaluation, our contribution is to provide developers with an approach to evaluating the support of BPMN elements in a BPMS. Following the steps proposed in this paper, we identified the limitations and devised a method for implementing the remaining BPMN elements not yet implemented.



## 1 INTRODUCTION

Business Process Management (BPM) provides a set of techniques for the analysis, implementation, enactment and continuous improvement (i.e., evolution) of business processes in different types of organizations (Weske, 2014; Dumas et al., 2018). Business processes[1] reflect the operation of organizations and allow controlling the development of services and products that need to be delivered to customers.

Processes can be graphically represented via Business Process Model and Notation (BPMN) (OMG, 2013). BPMN has become a standard for process modeling, disseminated by the Object Management Group (OMG). BPMN 2.0 is an ISO standard since 2013 (ISO, 2013).

BPMN enables to describe real-world processes, which can be supported by BPMN tools (Cortes-Cornax et al., 2015). For business practitioners, BPMN enables internal communication of business procedures, business-IT alignment and collaboration among business partners (Salles et al., 2013; Arevalo et al., 2016; Salles et al., 2018).

A process model in BPMN is called *process diagram*, and two or more processes interacting with each other can form a *collaboration diagram*. *Collaboration diagrams* can comprise the following set of basic notational elements[2]: *activities*, *events*, *gateways*, *groups*, *annotations*, *data objects*, *databases*, *pools*, *lanes*, *sequence flows*, *message flows* and *associations*).

BPM life cycle phases include process *modeling*, *configuration*, *implementation*, *execution* and *validation* (Weske, 2014; Dumas et al., 2018). During modeling, we can model a process in BPMN (OMG, 2013). The process modeled in BPMN can then be implemented, i.e., be further detailed at a granular level that allows its automated execution. These ac-

[a] https://orcid.org/0000-0003-3291-5208
[b] https://orcid.org/0000-0001-6261-1497
[c] https://orcid.org/0000-0002-9166-8801
[d] https://orcid.org/0000-0002-0620-9302

[1]For the sake of simplicity, in this paper, we use only the term *process* to refer to the expression *business process*.

[2]For the sake of simplicity, in this paper, we use only the term *element* to refer to the expression *notational element*.



Santos, C., Fantinato, M., Moreira de Oliveira, J. and Thom, L.
Evaluating Support for Implementing BPMN 2.0 Elements in Business Process Management Systems.
DOI: 10.5220/0009356501930200
In *Proceedings of the 22nd International Conference on Enterprise Information Systems (ICEIS 2020) - Volume 2*, pages 193-200
ISBN: 978-989-758-423-7

tivities can be supported by a BPM System (BPMS). A BPMS aims to coordinate an automated process model so that all work is done at the right time and by the right resource (Weske, 2014; Dumas et al., 2018).

Automating a BPMN process model requires providing the model with implementation details that are not required when it comes to only describing process models for business purposes. These implementation details are required to remove ambiguities and need to be accurate so that the process model can be interpreted by a BPMS (Gassen et al., 2015; Dumas et al., 2018; Santos et al., 2019). For example, if a process model contains abstract tasks, one must specify their attributes, documents and resources.

The lack of detailed information on implementation aspects in BPMN represents a technical challenge to transforming a process model into an executable version. As a result, each BPMS supplier uses its proprietary format to map and transform a process model into an *executable model* (Geiger et al., 2013). Consequently, the adherence of current BPMSs, considering the BPMN elements they implement, to the BPMN specification is unknown.

In this paper, we present the results of an *evaluation of BPMSs considering their degree of implementation of BPMN 2.0 elements*. Our goal is to measure and show the completeness of BPMN implementations in BPMSs. Supported by our approach, BPMS developers can identify missing elements in their BPMSs to implement them as well as verify whether the present elements are implemented following the BPMN 2.0 specification so that nonconforming elements can be adjusted.

For example, considering the BPMN element *message flow*, a BPMS supplier can verify whether its BPMS implements it. If the element is missing, the BPMS developer may implement it. On the other hand, if this element is already implemented, the BPMS developer can verify the element behavior. If the element implementation does not follow the BPMN 2.0 specification, the developer must adjust it to comply with the official specification.

The protocol for evaluating the BPMN elements implemented in BPMSs is summarized by these steps: *(i)* We chose free BPMSs to be evaluate; *(ii)* We defined as scope the BPMN elements used for *collaboration diagrams* as defined in the BPMN 2.0 specification (ISO, 2013); *(iii)* We modeled a process in each selected BPMS verifying whether the elements were implemented in the BPMSs and for the implemented elements, we verified whether they followed the BPMN 2.0 specification; *(iv)* To systematize this evaluation, we set a score for the implemented elements of each BPMS according to the level of adherence to the BPMN specification.

Our main contribution is to present an evaluation of whether and how BPMSs implement BPMN elements, with respect to the BPMN 2.0 specification. Some insights related to this evaluation could be obtained, including: the profile of BPMN elements commonly unimplemented in BPMSs, the limitations of current BPMN implementations in BPMSs and future research directions focusing on improving BPMSs and the BPMN specification.

The remainder of this paper is structured as follows: Section 2 presents the related work; Section 3 details the protocol defined to select and evaluate the BPMSs; Section 4 depicts the results obtained; and Section 6 summarizes the approach, discusses its limitations and concludes the paper.

## 2 RELATED WORK

In this section, we analyzed studies related somehow to model transformation targeting at execution or simulation models.

Table 1 summarizes the related work discussed in this section. In summary, we found three types of approaches: *(a)* BPMS limitations, i.e., studies whose goal is to evaluate overall BPMS limitations; *(b)* model transformation limitations, i.e., studies whose goal is to evaluate limitations on the model transformation; *(c)* model transformation proposal, i.e., studies whose goal is to propose new approaches for model transformation.

Table 1: Comparison of Related Work [*(a)* BPMS Limitations; *(B)* Model Transformation Limitations; *(C)* Model Transformation Proposal].

| Authors & year | *(a)* | *(b)* | *(c)* |
|---|---|---|---|
| Börger (2012) | X | | |
| Cetinkaya et al. (2012) | | | X |
| Peralta et al. (2014) | | | X |
| Geiger et al. (2013) | | X | |
| Bocciarelli et al. (2014) | | | X |
| Kluza et al. (2015) | | | X |
| Meidan et al. (2017) | X | | |

Börger (2012) evaluated the BPMN 2.0 specification and found many behavioral issues that the specification leaves open. The issues described include the BPM life cycle concept that does not characterize the mechanism of interruption and compensation for transactions; the expression “evaluate” is not clear, because it is not defined when a condition defined in any part of the process model can be evaluated; a general notion of state is missing and hence the definition of data dependent conditions is only poorly supported. As a consequence of these issues, according to these authors, the BPMS suppliers typically implement only subsets of the so-called standard and still are often only partially compatible with each other.

Meidan et al. (2017) conducted a survey to evaluate BPMSs and highlight each phase of the BPM life cycle fully supported by them. The survey combined Systematic Literature Review (SLR) and quality models. SLR was used to select the BPMSs. The following BPMSs were selected: Activiti, Bonita, jBPM, ProcessMaker, uEngine and Camunda. To evaluate the selected BPMSs, the authors considered as quality models criteria related to modeling, design, deployment, execution & control and analysis. As the main result, these authors presented an evaluation presenting the BPMS closest to the aims of the BPM life cycle when compared to other BPMSs.

Geiger et al. (2013) studied the issues in BPMN serialization that arise due to the complexity and inconsistency of the BPMN specification. *Serialization* is to translate data structures or objects into a format that can be stored and reconstructed later. The authors considered the Web Services Business Process Execution Language (WS-BPEL) as a model serialization. WS-BPEL is a serialization format for an executable service-based process model. The BPMN specification provides a mapping from the process model in BPMN to WS-BPEL. However, these authors depict that the BPMN specification does not provide the correct serialization format, making it difficult to turn a process model executable. The approach provides two contributions: a list of relevant constraints in the BPMN specification to turn a process model executable and the respective serialization.

Cetinkaya et al. (2012) proposed an approach to transform a process model into a simulation model. These authors developed a framework that minimizes the gap between these models. The framework was developed based on Discrete Event System Specification (DEVS), which is a mathematical formalism used to represent systems. Models represented in DEVS are called *atomic models*. These atomic models are defined with the following information: the set of input values, the set of output values, the set of states, the internal transition function, the external transition function, the output function and the time advance function. For each element in BPMN, there is a corresponding element target in DEVS. As the main result of this transformation, one can obtain a model that can be executable at the simulation level.

Peralta et al. (2014) proposed metrics to analyze and measure a process model, aiming to improve and facilitate its implementation in the cloud. These authors considered as metrics: number of activities, the total number of precedence dependencies between activities, connectivity level between activities, number of split nodes of parallel, number of XOR-split, number of OR-split, number of AND-join, number of XOR-join and number of OR-join. These metrics were applied in the process model, in a semi-automated way, and resulted in a description of its needs to implement a process model in the cloud.

Bocciarelli et al. (2014) proposed an approach to transform process models to the simulation level. To this end, these authors developed a Java-based tool that, taking as input a BPMN 2.0 process model, obtain, as a result, a simulation model, called *eBPMN*. The eBPMN core includes a set of BPMN elements, adapted to the simulation model. According to these authors, this model allows the process model to be executed with details that are needed to simulate it.

Kluza et al. (2015) focused on the semantic of the process model. To achieve this, the authors applied ontology concepts on free BPMSs (Activi, jBPM and Camunda) to increase the semantic representation. According to these authors, applying semantic modeling allows disambiguation of data description and control of its integrity. As a result, this approach allows data transformation, during translating a process model into an execution model.

The approaches described above have different types of goals, including: evaluating the BPMN 2.0 specification, evaluating BPMSs in terms of their BPMN implementation, identifying limitations to transforming a process model into an executable model, or proposing an approach to transforming a process model into an executable model.

As for the evaluation of BPMSs regarding the BPMN elements implemented by them, the works found do not address the evaluation of the implementation of the elements *per se*. Approaches focused on the evaluation of the BPMN specification also do not verify the elements. Works that propose new model transformation approaches can lead to an even greater difficulty in understanding process models, adding more concepts to be understood beyond BPMN. On the other hand, the BPMN element evaluation proposed herein can be seen as a way to provide a set of elements that can be useful to users if their goal is to automate their process models.

## 3 RESEARCH PROTOCOL

This section presents the research protocol of our evaluation of the implementation of BPMN elements in BPMSs. Our goal is to present the steps we fol-

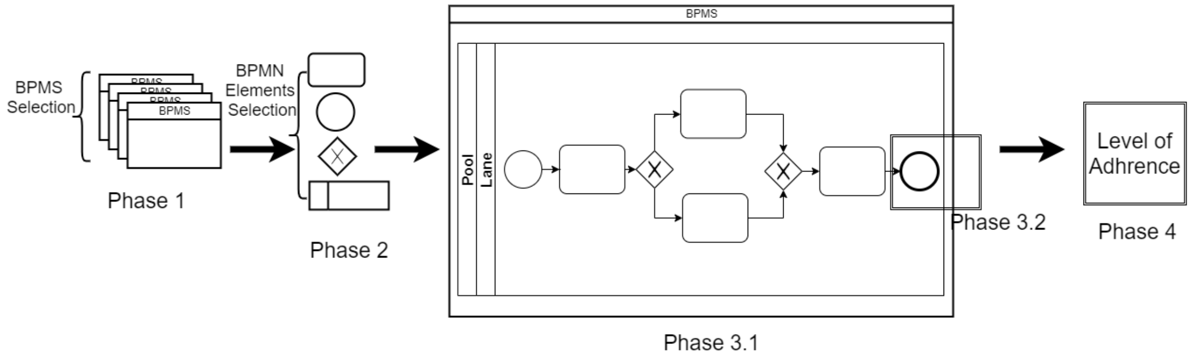


Figure 1: Phases of the Evaluation of BPMN Elements in BPMSs.

lowed as an approach so that they can be followed by those looking to evaluate other BPMN elements or BPMSs in the future.

Briefly, our approach comprised three phases:

1. Selecting the set of BPMSs to be evaluated.
2. Selecting the BPMN elements for evaluation.
3. Evaluating the implementation of BPMN elements in the selected BPMSs (verifying whether the BPMN elements are implemented in the BPMSs and verifying whether the implemented BPMN elements are correctly implemented).
4. Scoring the evaluated BPMSs according to which (and how) BPMN elements are implemented.

Figure 1 depicts the conducted phases. First, we selected a set of BPMSs to be evaluated (cf. "Phase 1"), considering only free BPMSs. Then, we selected a set of BPMN elements (cf. "Phase 2"), for which we considered only elements used for *collaboration diagrams* as defined in the BPMN specification (ISO, 2013). Based on the set of selected BPMN elements, we evaluated one by one for each selected BPMS to verify: whether each BPMN element is implemented in the BPMS (cf. "Phase 3.1"); and whether the implemented BPMN elements are implemented according to the BPMN specification (cf. "Phase 3.2"). Results are counted with a score that defines the level of adherence of each BPMS to the BPMN specification (cf. "Phase 4"). Finally, we analyzed the results.

### 3.1 Selecting the BPMSs

The first phase of the evaluation is the selection of BPMSs. To this end, we considered the following criteria: *(i)* the BPMS must be listed in the BPMN official website[3], present in the "Implementers" list; *(ii)* the BPMS must be free software; and *(iii)* the BPMS must be able to implement and execute processes modeled in BPMN 2.0.

We used only free BPMSs following approaches found in related work (Meidan et al., 2017; Corradini et al., 2018; Kluza et al., 2015). The use of free BPMSs improves the reproducibility of the evaluation presented in this paper. As for the followed procedure, we verified the BPMN website for information about the tool suppliers. If a free version of the BPMS was available, we downloaded it. This approach can also be applied for the commercial versions, making our study applicable for a full set of BPMSs.

In addition, we considered only BPMSs that allow implementing and executing processes modeled in BPMN 2.0. By execution, we mean that the BPMS must be able to carry out the process, by instantiating and controlling its performance. We selected the BPMSs on March 2018.

Finally, Table 2 depicts the four BPMSs selected. Considering all defined criteria, only four BPMSs justify a closer evaluation regarding the degree to which they actually implement the BPMN specification.

Table 2: Final Selected BPMSs.

| BPMS | Supplier | Version | Release date |
|---|---|---|---|
| Bonita BPM | Bonitasoft | 7.6.2 | Jan, 2018 |
| Camunda BPM | Camunda | 7.9.0 | May, 2018 |
| jBPM | KIE Group | 7.7 | Mar, 2018 |
| Web Ratio | WebRatio | 8.8.1 | Not found |

### 3.2 Selecting the BPMN Elements

In the second phase, we selected the set of BPMN 2.0 elements to be used in the evaluation. To this end, we considered only the BPMN elements used for *collaboration diagrams* as defined in the BPMN specification. Thus, we did not consider the elements used specifically for *conversation diagrams* and *choreog-*

[3]http://bpmn.org

*raphy diagrams*. Collaboration diagrams are those more commonly used by BPMN practitioners, which justifies our decision. The set of elements chosen allows the reader of a process model in BPMN to easily recognize the types of elements used and understand the diagram (ISO, 2013). In total, there are 83 elements, considering both the basic elements and their extended versions, which are grouped in five categories – flow objects, artifacts, data, swimlanes and connecting objects.

# 4 EVALUATING THE IMPLEMENTATION OF BPMN ELEMENTS IN BPMS

Once the BPMS and BPMN elements to be evaluated were selected, we proceeded with the third phase of the research protocol (cf. Section 3) and conducted the evaluation itself. To this end, we evaluated the BPMN implementations in two aspects:

1. Verifying whether the BPMN elements are implemented in the BPMSs.
2. Verifying whether the implemented BPMN elements are correctly implemented.

## 4.1 Identifying Implemented BPMN Elements

For this first evaluation aspect, we created a process model on each BPMS under evaluation and tried to add to it each one of the BPMN elements selected for evaluation (cf. Section 3.1). Thus, we seek to identify elements not implemented in BPMSs, i.e., those that would not be possible to be added to the process model as they were not available in the BPMS.

To select the elements, first, we verified the editor of each BPMS. If the BPMN element was available for use in the BPMS editor, we concluded that the corresponding BPMS implements it. We also tried to create extended versions of elements through the element properties, for example by right-clicking on the element. This is often available in some tools for changing the type of some elements, e.g., changing an abstract task to a script task or a standard event to a message event. If there was no reference to the evaluated BPMN element in the BPMS editor or in a base element properties, we concluded that the corresponding BPMS does not implement it. To control the conducted evaluation, we used a spreadsheet where we marked an "X" for each implemented BPMN element. We organized this spreadsheet by BPMN element groups.

## 4.2 Evaluating Implemented BPMN Elements

For the second evaluation aspect, we modeled different processes by incrementally using all the BPMN elements implemented in each BPMS, as identified according to Section 4.1.

After modeled the process, we followed a procedure for obtaining a quantitative assessment for the following groups of elements: activities, events, gateways, data, connection objects, swimlanes and artifacts. This procedure allowed a systematic evaluation of the implementation of the BPMN elements and a homogeneous comparison between the BPMSs.

The procedure for evaluating the implemented BPMN elements consists of: *Element score*: we assigned 0, 1 or 2 points to each BPMN element, considering that *0 points* means that the BPMS does not implement the BPMN element, *1 point* means that the BPMN element is partially implemented considering the BPMN specification and *2 points* means that the BPMN element is fully implemented considering the BPMN specification. For example, considering the part of the definition of the *XOR split-gateway* element that states "The first condition that evaluates to true determines the sequence the token is sent to and no more conditions are henceforth evaluated": if the BPMS does not implement this part of the definition, it receives a maximum of 1 point; but if the BPMS implements this part of the definition, it can receive 2 points, depending on whether or not the rest of the definition of this element is implemented.

*Element group score*: for each group of BPMN elements, we calculated (cf. Equation 1) a Group Score (*GS*) by summing the individual scores of all BPMN elements for the group and dividing the result by the number of elements in the group. In Equation 1, $n$ represents the number of elements in the group and $S$ represents the individual score of the element.

$$GS = \frac{\sum_{i=1}^{n} S_i}{n} \tag{1}$$

*Normalized group score*: for all element group scores previously calculated (cf. Equation 1), we calculated (cf. Equation 2) a Normalized Group Score ($GS_{norm}$) in the 0-10 range, dividing each group score by the highest score obtained by all element groups, and then multiplying the result by 10. In Equation 2, $n$ represents the number of element groups, $GS$ represents the group score and $max(GS)$ represents the highest among the group scores.

Table 3: Summary of BPMS Evaluation.

| BPMN element group | Bonita BPM | Web Ratio BPM | Camunda BPM | jBPM |
|---|---|---|---|---|
| Activities | 8.33 | 5.00 | 8.89 | 7.78 |
| Events | 7.04 | 7.41 | 6.67 | 6.48 |
| Gateways | 7.00 | 9.00 | 7.00 | 7.00 |
| Connection objects | 8.00 | 4.00 | 6.00 | 4.00 |
| Artifacts | 10.00 | 7.50 | 10.00 | 5.00 |
| Swimlanes | 10.00 | 10.00 | 10.00 | 10.00 |
| $BPMS_{score}$ | 8.40 | 7.15 | 8.09 | 6.71 |

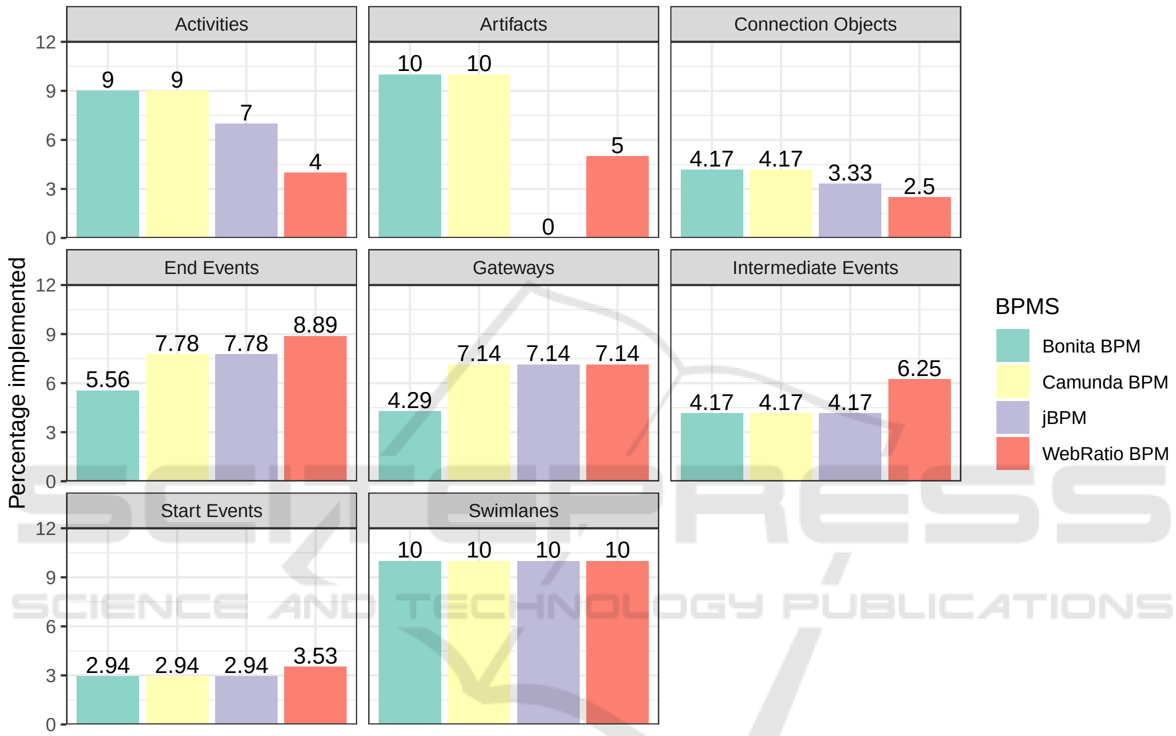


Figure 2: Percentage of Implemented BPMN Elements by Element Group and by BPMS.



$$GS_{norm_i} = \frac{GS_i}{max(GS)} x10, \forall i \in 1..n \quad (2)$$

*BPMS score*: for each BPMS, we calculated (cf. Equation 3) the $BPMS_{score}$ by summing all the group scores for the BPMS and dividing the result by the number of element groups. In Equation 3, $n$ represents the number of BPMN element groups and $GS_{norm}$ represents the normalized group score.

$$BPMS_{score} = \frac{\sum_{i=1}^{n} GS_{norm_i}}{n} \quad (3)$$

Table 3 summarizes the evaluation results. Each row represents a group score (cf. Equation 1), for the evaluations of the BPMN elements of the corresponding group. For example, row "Activities" depicts the scores for this group of BPMN elements, for each BPMS evaluated. The $BPMS_{score}$ (cf. Equation 3) row depicts the final scores obtained by BPMS, considering the six groups above.

## 5 RESULT ANALYSIS

Our study found that support for BPMN elements in BPMSs is limited considering the four BPMSs evaluated. Of the 79 elements evaluated, only 27 are implemented by at least one of the BPMSs (corresponding to 34.18%). We also identified that different BPMSs usually implement the same set of BPMN elements. For example, in terms of *gateways*, BPMSs usually implement AND, XOR and OR gateways while the event-based gateway is not implemented. One hypothesis is that only the BPMN elements most often used by practitioners are implemented in BPMSs.

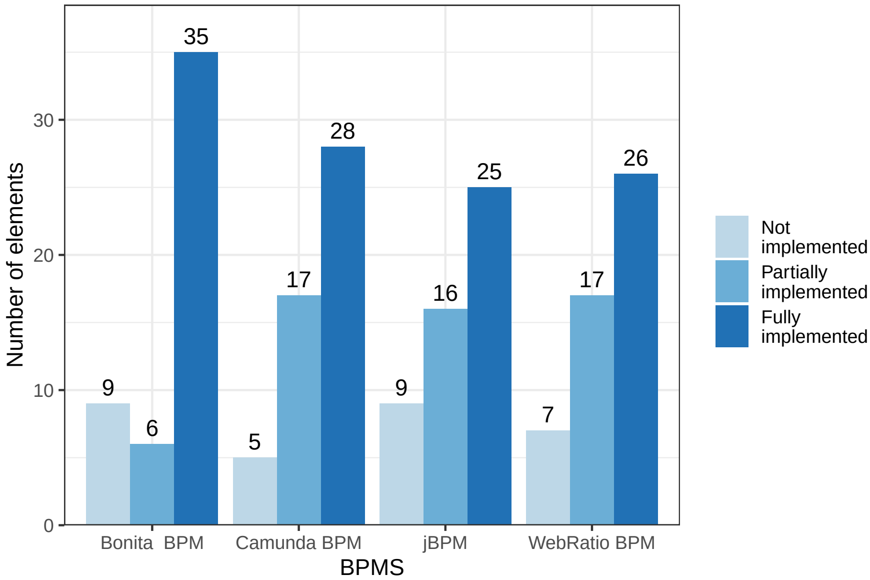


Figure 3: Number of Implemented BPMN Elements by BPMS.

Fig. 2 presents more information about the results obtained by each group of BPMN elements. We split *events* into three groups to allow a more precise result analysis: *start events*, *intermediate events* and *end events*. As a result, one can observe that "start events" is the element group with the fewest elements implemented (average of 30.88%). In contrast, "swimlanes" is the element group with the most elements implemented (average of 100%) followed by "artifacts" (average of 62.5%). One hypothesis for the higher level of implementation of these two groups is their possible ease of implementation as both groups represent simpler elements in terms of behavior rules according to the BPMN specification.

Fig. 3 depicts the results grouped by BPMS, considering the elements that there are an implementation in the BPMS. We present an analysis showing the number of elements fully implemented, partially implemented and not implemented according to the BPMN specification. From this data, one can conclude that Bonita BPM is the BPMS with the largest number of BPMN elements fully implemented, considering the BPMN specification. From 50 elements, 35 are fully implemented (42.17%), 6 are partially implemented (7.23%) and 9 not implemented (10.84%).

Regarding the scope of implementation, we can conclude that although the BPMS implements few elements, not all BPMS perform as defined in the BPMN specification. While Bonita BPM obtained the highest score (about 8.40), jBPM obtained the lowest score (6.71). This means that Bonita BPM offers higher support to BPMN, compared to another tool.

About the evaluated groups, we can observe that BPMSs in general support swimlanes (all scored 10) and artifacts (Bonita and Camunda scored 10; WebRatio 7.5 and jBPM 5). In opposition, activities, events and gateways are weakly supported.

## 6 CONCLUSION

In this paper, we presented an evaluation of the current state of the support of the BPMN elements in available BPMS. To achieve this aim, we focused on a protocol that allows us to investigate the BPMS. With the information about the implementation, we can identify the elements that need to be implemented and hence investigate how to do it.

According to our evaluation of related works, the available investigations are focused on verifying resource aspects of the BPMS. Our research can help to determine the limits of support, considering the BPMS. With this, one can develop strategies to implement the remained unimplemented elements. With these elements implemented, it is possible to increment the number of element offers.

Our main contribution is an approach to evaluate the support of elements on a set of selected BPMS. Following the steps proposed in this paper, it is possible to identify limitations on a BPMS in development and plan a way to implement the remaining elements that there is no existent implementation. Applying the protocol, we obtained a set of 27 elements that there is at least one BPMS that implements it. With the 27 elements, we verified the implementation, revealing that there are elements that the implementation does not follow the BPMN specification.

We identified that the BPMS evaluated in this research does not cover all BPMN elements. For example, this protocol may be applied to *Choreographies* and *Conversations*. About the selection of BPMSs, considering many BPMSs are a not-free license, this condition reduced our number of BPMS evaluated. We cannot assure if these BPMSs provide support to the elements that there is no implementation.

As future work, we suggest: *(i)* analyze the lack of semantics with the elements that there is no implementation; *(ii)* analyze if the missing elements will increase the difficulty to understand the process model, *(iii)* analyze the relation about modeling errors with the lack of elements and *(iv)* propose pseudo-algorithm to the missing elements, aiming to facilitate the development of these elements.

## ACKNOWLEDGEMENTS

José Palazzo Moreira de Oliveira receive support from CNPq by grants 301425/2018-3 and 400954/2016-8; Carlos Francisco Habekost dos Santos is a scholarship holder from CNPq; this study was financed in part by the CAPES - Brazil - Finance Code 001.

## REFERENCES

Arevalo, C., Escalona, M. J., Ramos, I., and Domínguez-Muñoz, M. (2016). A metamodel to integrate business processes time perspective in BPMN 2.0. *Information and Software Technology*, 77:17–33.

Bocciarelli, P., D'Ambrogio, A., Giglio, A., Paglia, E., and Gianni, D. (2014). Empowering business process simulation through automated model transformations. In *Proceedings of the Symposium on Theory of Modeling & Simulation - DEVS Integrative*, DEVS '14, pages 1–9.

Börger, E. (2012). Approaches to modeling business processes: A critical analysis of bpmn, workflow patterns and yawl. *Software and Systems Modeling*, 11(3):305–318.

Cetinkaya, D., Verbraeck, A., and Seck, M. D. (2012). Model transformation from bpmn to devs in the mdd4ms framework. In *Proceedings of the 2012 Symposium on Theory of Modeling and Simulation - DEVS Integrative M&S Symposium*, TMS/DEVS '12, pages 1–6.

Corradini, F., Ferrari, A., Fornari, F., Gnesi, S., Polini, A., Re, B., and Spagnolo, G. O. (2018). A guidelines framework for understandable bpmn models. *Data & Knowledge Engineering*, 113:129–154.

Cortes-Cornax, M., Matei, A., Dupuy-Chessa, S., Rieu, D., Mandran, N., and Letier, E. (2015). Using intentional fragments to bridge the gap between organizational and intentional levels. *Information and Software Technology*, 58:1–19.

Dumas, M., Rosa, M. L., Mendling, J., and Reijers, H. A. (2018). *Fundamentals of Business Process Management, Second Edition*. Springer.

Gassen, J. B., Mendling, J., Thom, L. H., and Palazzo Moreira de Oliveira, J. (2015). Towards guiding process modelers depending upon their expertise levels. In *Advanced Information Systems Engineering Workshops - CAiSE 2015 International Workshops, Stockholm, Sweden, June 8-9, 2015, Proceedings*, pages 69–80.

Geiger, M., Wirtz, G., and der Weberei, A. (2013). Bpmn 2.0 serialization-standard compliance issues and evaluation of modeling tools. In *International Workshop on Enterprise Modeling and Information Systems Architectures*, pages 177–190. Citeseer.

ISO (2013). ISO/IEC 19510:2013: Information technology – Object Management Group Business Process Model and Notation, V2.0.2. Standard, International Organization for Standardization, Geneva.

Kluza, K., Kaczor, K., Nalepa, G. J., and Ślażyński, M. (2015). Opportunities for business process semantization in open-source process execution environments. *2015 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 1307–1314.

Meidan, A., Garca-Garca, J., Escalona, M., and Ramos, I. (2017). A survey on business processes management suites. *Computer Standards & Interfaces*, 51(C):71–86.

OMG, O. M. G. (2013). Business process model and notation (bpmn) version 2.0. Standard.

Peralta, M., Salgado, C., Baigorria, L., Riesco, D., Montejano, G., Debnath, N., and Hu, J. (2014). Workflow models: Management and quality of process in the cloud. In *International Conference on Information Technology: New Generations*, pages 91–96.

Salles, G. B. M., Fantinato, M., Barros, V. A., and Albuquerque, J. P. (2018). Evaluation of the StrAli-BPM approach: Strategic alignment with BPM using agreements in different levels. *International Journal of Business Information Systems*, 27(4):433–465.

Salles, G. B. M., Fantinato, M., Nishijima, M., and Albuquerque, J. P. (2013). A contribution to organizational and operational strategic alignment: Incorporating business level agreements into business process modeling. In *Proc. of the 10th Int. Conf. on Services Computing*, SCC 2013, pages 17–24. IEEE.

Santos, C. F. H. D., Thom, L. H., Cota, É., and Fantinato, M. (2019). Supporting bpmn tool developers through meta-algorithms. *International Journal of Business Information Systems*, 32(4):460–488.

Weske, M. (2014). *Business Process Management*. Springer, Berlin Heidelberg, 2 edition.