UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

CARLOS FRANCISCO HABEKOST DOS SANTOS

Incrementing the Implementation Perspective of the Business Process Model and Notation: Analysis, Survey and Pseudo-algorithms

Thesis presented in partial fulfillment of the requirements for the degree of Doctor of Computer Science

Advisor: Prof. Dr. Lucineia Heloisa Thom

Porto Alegre
2021

CIP — CATALOGING-IN-PUBLICATION

Santos, Carlos Francisco Habekost dos

Incrementing the Implementation Perspective of the Business Process Model and Notation: Analysis, Survey and Pseudo-algorithms / Carlos Francisco Habekost dos Santos. – Porto Alegre: PPGC da UFRGS, 2021.

98 f.: il.

Thesis (Ph.D.) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2021. Advisor: Lucineia Heloisa Thom.

1. Notational elements. 2. Automation. 3. Implementation. 4. BPMN. 5. BPMS. I. Thom, Lucineia Heloisa. II. Título.

“Nunca deixe que lhe digam que não vale a pena
Acreditar no sonho que se tem
Ou que seus planos nunca vão, dar certo
Ou que você nunca vai ser alguém”
— Renato Russo

ACKNOWLEDGMENTS

Chegamos ao fim de um ciclo. Um ciclo que teve os seus alto e baixos, mas após tudo isso, sinto-me com o dever cumprido. Foi uma jornada de 5 anos, da qual não me arrependo de ter passado. Foi uma fase de grande crescimento profissional e pessoal. E durante essa caminhada, várias pessoas que passaram o meu caminho e que contribuíram, seja de forma direta ou indireta para conclusão desse ciclo:

Aos meus pais, João Francisco e Sueli Teresinha, por toda a força e apoio dado, principalmente por darem sempre força para estudar e apoiar minhas escolhas. Com toda a certeza que sem eles não teria chegado onde cheguei.

À minha orientadora, Lucinéia Thom, por ter me aceito e dado dicas e contribuído muito para meu crescimento. Evolui muito e sinto-me mais completo profissionalmente ao final deste ciclo.

Aos meus amigos de laboratório 213: Ana Klock, Guilherme Medeiros, Brenda Salenave, Gabriel Machado, que durante o tempo de bolsa e laboratório (antes da pandemia) foram sempre gentis, colaboradores e o mais importante... Amigos!

Aos meus amigos de apartamento e de vida, Willan de Nardi e Douglas Ostruca, pelas conversas, ajudas, momentos de descontração e principalmente, apoio nos momentos difíceis. Com certeza, vocês fizeram a diferença para eu chegar aqui.

Um agradecimento especial ao Nicolas Bohnenberger pela colaboração no desenvolvimento do analisador. Sua participação foi de grande valia!

E um agradecimento ao pessoal do Instituto de Informática da UFRGS, por sempre manter a estrutura do instituto funcionando bem, mesmo em momentos de crise. Vida longa a universidade pública!

ABSTRACT

Business Process Management (BPM) is a discipline that aims to document, standardize business processes and increase efficiency and quality in the execution of processes. In BPM, a business process can be represented graphically through the Business Process Model and Notation (BPMN), an ISO standardization for process modeling. With modeled processes, organizations may demand the need for automation. Automation requires information technology to perform a process model using a Business Process Management System (BPMS). However, transforming a business-level process to an automated-level process requires implementation details that are not simple to identify. For example, a process model at the business level does not need to specify what computational resources are necessary to perform or specify the data consumed by a task. Besides knowing the details for transforming the process model, it is necessary to understand the limitations of BPMN and BPMS related to the notational elements, understanding the implemented elements in BPMS, and if these elements follow the BPMN specification. In this thesis, we analyzed the implementation of notational elements in BPMS, performing a protocol to identify implemented elements. Furthermore, we evaluate whether the implementation follows the BPMN specification. This analysis would help those who want to transform a process model, in order to identify notational elements that can be automated. Also, we analyzed the frequencies in which elements appear in the process models. In this case, we evaluated 186 accurate models obtained from a Brazilian public university, a processes database, and a master study research. And along with the analysis of implemented elements, we could identify the preferences of users. We performed a third analysis to obtain feedback from process analysts (people responsible for process modeling) and BPMS developers related to BPMN. We discovered that the preference of users is more focused in improving the implementation of notational elements, than in the insertion of new elements. Our last analysis focused on proposed pseudo-algorithms (structured algorithms) for already implemented elements in the BPMS, with a suggestion to improve the BPMN elements. All analysis performed in this thesis guides the process analysis to identify the notational elements to focus on process automation, besides identifying the limits of BPMS.

Keywords: Notational elements. Automation. Implementation. BPMN. BPMS.

Incrementando a Perspectiva de Implementação da Notação e Modelo de Processo de Negócios: Análise, *Survey* e Pseudo-algoritmos

RESUMO

Gerenciamento de Processos de Negócio (BPM) é uma disciplina que visa a documentação e a padronização dos processos de negócio, sendo que um processo de negócio pode ser representado graficamente, através da Notação e Modelo de Processo de Negócio (BPMN - *Business Process Model and Notation*). A partir desses processos modelados, as organizações podem demandar a necessidade de automação desses processos. A automação requer o uso de tecnologia da informação para que modelos de processos possam ser executados através do Sistema de Gerenciamento de Processos de Negócio (BPMS - *Business Process Management System*). No contexto da automação de processos, tem-se com problema, diferentes BPMS que implementam diferentes elementos da BPMN. O objetivo desta tese foi analisar a implementação dos elementos da BPMN de diferentes formas: *(i)* análise de quais elementos estão implementados em BPMS; *(ii)* análise de modelos de processo, em BPMN, identificando quais elementos são comumente utilizados na modelagem de processos; *(iii)* aplicação de um *survey*, identificando o *feedback* dos usuários, no contexto de analista de processos e no contexto de desenvolvedor de BPMS. Como resultados obtidos, observa-se que nem todos os elementos da BPMN estão implementados em BPMS. Além disso, são utilizados os mesmos elementos para modelagem dos processos (como por exemplo: tarefa, desvio exclusivo, dentre outros). Como resultado do *survey*, nós identificamos por exemplo, no contexto dos analistas de processo, que a notação oferece muitos elementos, que possuem as mesmas funcionalidades. No contexto dos desenvolvedores, foi identificado que a notação é normalmente adaptada, de acordo com a necessidade da organização. A última etapa da tese consistiu na elaboração de pseudo-algoritmos (algoritmos estruturados) para elementos já implementados no BPMS, adicionando funcionalidades complementares (por exemplo um elemento de desvio poder iniciar uma instância de processo), como uma sugestão de incremento da BPMN. Como conclusões obtidas nessa tese, destaca-se que as análises realizadas permitiram identificar a realidade da BPMN, em termos de implementação em BPMS, bem como desenvolvido um possível caminho de incremento da notação, através de pseudo-algoritmos.

Palavras-chave: Elementos notacionais. Automação. Implementação. BPMN. BPMS.

LIST OF ABBREVIATIONS AND ACRONYMS

BPM	Business Process Management
BPMN	Business Process Management Notation
BPMS	Business process Management System
CFG	Control-flow Graph
ISO	International Organization for Standardization
OMG	Object Management Group
PD	Process Diagram
WfMC	Workflow Management Coalition
W3C	World Wide Web Consortium

LIST OF FIGURES

LIST OF TABLES

CONTENTS

1 INTRODUCTION

The activities of an organization (such as perform an order, hire a service) are related to business processes. Every organization, regardless of their type, has to manager a number of processes. A business process is a set of activities performed in a particular partial order to fulfill a specific business objective of the organization (WESKE, 2014).

In this context, 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; WFMC, 1999; DUMAS et al., 2018). Business processes reflect the operation of organizations and allow controlling the development of services and products that need to be delivered to a client or another organization. Thus, through BPM, organizations can acquire a competitive edge (NASCIMENTO et al., 2013; PRIEGO-ROCHE et al., 2012; STEMBERGER et al., 2018).

BPM life cycle phases include process *identification*, *discovery*, *analysis*, *redesign*, *implementation* and *process monitoring and controlling* (DUMAS et al., 2018). During the modeling phase (discovery phase), we can model a process in the Business Process Model and Notation (BPMN) (OMG, 2013). In this thesis, we used BPMN 2.0, named as only BPMN. BPMN provides elements that allow to capture the reality of the organization through data objects, signals, parallelism, among others aspects. The process modeled in BPMN can then be implemented, i.e., be further detailed at a granular level that allows its automated execution. These activities can be supported by a Business Process Management System (BPMS).

In recent years, several notations to model processes have been proposed, such as Event Process Chain (EPC) (DUMAS et al., 2018), Activity Diagrams (WESKE, 2014) and Petri nets (AALST et al., 2003). Nonetheless, the BPMN has become a standard for process modeling, disseminated by the Object Management Group (OMG). Since 2013, the BPMN is an ISO standard (ISO, 2013).

BPMN is an effective way to describe the real-world process to be supported by BPMN tools (CORTES-CORNAX et al., 2015). It provides a broad set of notational elements such as activities, gateways and events (OMG, 2013). In this thesis, the word *element* is used to refer to the expression of *notational elements* used in this thesis. BPMN helps to provide better documentation and standardization of process as well as to improve efficiency and quality of execution (THOM; REICHERT; IOCHPE, 2009). For business

professionals, BPMN allows not only internal communication of business procedures but also business- IT alignment and collaboration among business partners (AREVALO et al., 2016).

Process model is the name of a process modeled in the BPMN notation and considers a basic set of elements: *(i)* flow objects, which are the main elements used to define the behavior of a process (there are three flow objects – events, activities and gateways); *(ii)* artifacts, which are focused on providing additional information about the process (BPMN offers two standard artifacts – group and annotation); *(iii)* data, which represents the data consumed or generated by the process (subdivided into data object and database); *(iv)* connection objects, which are focused on connecting two flow objects, artifacts or data (represented by sequence flow, message flow and association); and *(v)* divisions, which are used to group other elements (they are pool, lane and sub-lane).

1.1 Motivation

Succeeding the phases of the BPM life cycle, it is possible to turn the process model in a granularity level that allows implementing and executing the model in a BPMS. The purpose of a BPMS is to coordinate an automated business process model in such a way that all work is done at the right time by the right resource (DUMAS et al., 2018; WESKE, 2014).

Automating a BPMN process model requires an approach to process modeling with implementation details, which differs from a business-oriented process model that is not necessarily detailed and thus may contain ambiguities. These implementation details are necessary and need to be exact for a process model to be interpreted by a BPMS (DUMAS et al., 2018; SANTOS; THOM; FANTINATO, 2015; GASSEN et al., 2015). For example, considering a process model with abstract tasks, we need to specify the attributes, documents, resources and everything that is required to implement this element.

The challenge to transform a process model into an executable version is the lack of implementation aspects in BPMN specification regarding its elements. As a result, BPMS vendors used their proprietary format to map and transform a process model into an *executable model* (GEIGER; WIRTZ; WEBEREI, 2013; SANTOS et al., 2019). A different proprietary format of implementing the elements may cause a lack of interoperability of models between BPMS, i.e., one BPMS may not understand a process modeled in another BPMS. Thus, it is necessary to have a way to measure how adherent the BPMS

is, considering the elements. Figure 1.1 illustrates an example of different proprietary formats. Several proprietaries (*BPMS 1, BPMS 2, BPMS 3, BPMS 4*) implement different elements, leading to a situation where it is not possible to identify the implemented elements and the not implemented elements.

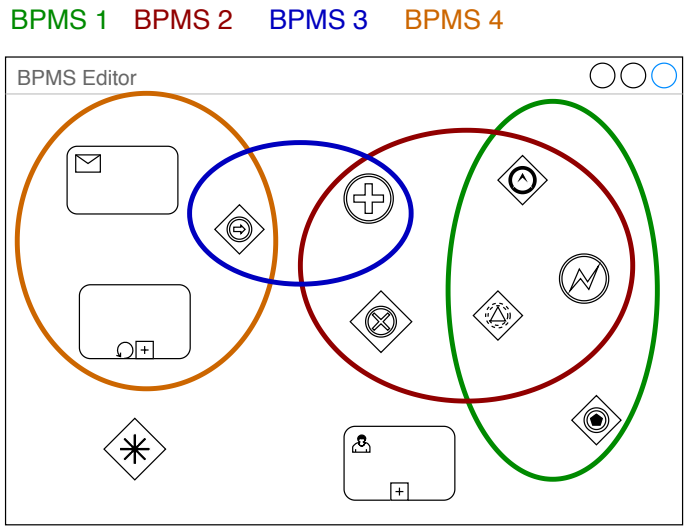


Figure 1.1 – Example of problem.
Source: The author.

In this context, it is necessary to provide an analysis aiming to identify the reality of BPMN elements implementation in existent BPMSs. Such analysis should allow identifying the missing elements, besides the set of preferred elements, by BPMS developers. Also, considering the elements that have an implementation, we can verify if such implementation follows the BPMN specification.

Some works proposed studies of BPMN behavior (BöRGER, 2012), serialization of the notation (GEIGER; WIRTZ; WEBEREI, 2013) and the transformation of a business process model from business level to simulation level (CETINKAYA; VERBRAECK; SECK, 2012). In common, the works presented the limitations of the BPMN implementation, highlighting: *(i)* not all elements are implemented; *(ii)* for the elements that have an implementation, such implementation does not follow the BPMN specification.

However, the works analyzed in this thesis do not identify the set of elements selected in BPMS development. Also, the researches do not verify whether the implementation of the elements follow the BPMN specification. Considering, for example, that users use BPMN notation for learning and training on process modeling, we expect an alignment among tools and process models with the specification. Otherwise, if there is no alignment, this would increase the possibility of errors in the modeling.

Therefore, when we are thinking of improving the BPMN implementation code,

we realize that it is necessary to analyze the implementation of BPMN in a BPMS. This analysis can help to define strategies to facilitate the implementation of the BPMN elements. As a strategy, we proposed pseudo-algorithms that represent the behavior of a given element, in a way that facilitates the implementation by the BPMS developers, with a basic structure of implementation represented by a pseudo-algorithm.

1.2 Research Goals

Considering that the implementation of the elements in BPMS must correspond to the BPMN specification and it is necessary to know the limits of implementation of BPMN elements (in terms of what we can or cannot implement in a BPMS), this thesis proposes *an approach to evaluate the limits of implementation of BPMN elements* according to the following objectives:

- Check the BPMN elements in order to identify the reality of their implementation in BPMS. Also, analyze if the implementation follows the BPMN specification.
- Check process models, in BPMN format, in order to identify the elements used by users during the process modeling.
- Verify the users' feedback about the use of BPMN.
- Define pseudo-algorithms to represent the behavior of the BPMN elements and so increase its readability.

To achieve the first objective, we defined a literature analysis regarding existent BPMS. In this context, we considered tools with free licenses that implement BPMN. The analysis started by verifying if the BPMS implements the analyzed elements. Afterwards, we verified if the implementation follows the BPMN specification.

For the second objective, we performed an analysis of 186 process models, modeled in BPMN. To perform this analysis, we developed a module. As the input of this module, we considered a process model in *.bpmn* format. As output, we obtain a *.csv* file with the name of the element and the respective frequency in the process model. This analysis allows us to identify the preferred elements during process modeling.

In the third objective, we applied a survey to verify the use of BPMN in two contexts: *(i)* process analyst and *(ii)* BPMS developer. In the analyst context, the survey allows us to identify the perception of the BPMN by users. In the developer's context, we intend to verify the limitation for the implementation of BPMN on the BPMS.

Finally, in order to achieve the fourth objective, we developed pseudo-algorithms for the elements whose implementation is missing. We considered the methodology presented in Santos, Thom and Fantinato (2015): Using the BPMN specification, we extracted the algorithm that represents the element's behavior.

1.3 Methodology

The development of this work began with a study of the literature regarding BPMN and BPMS. We analyzed the strengths and weaknesses of each study, aiming to identify how to facilitate the implementation of BPMN for developers in BPMS.

Through this work, we observed the combination of the BPMN analysis in the context of implementation in the BPMS. Besides, we concluded that the usage by the users would allow us to identify how the implementation reality is, being able to outline ways to improve the BPMN notation. Figure 1.2 depicts the steps of the methodology applied in the development of this work.

- *Step 01*: in this step, the BPMN analysis was carried out, considering its implementation in the BPMS. For this, we analyzed the BPMS with a free license and available for download.
- *Step 02*: we evaluated the use of BPMN in the context of process modeling. In order to do so, we considered a set of 58 process models. With this set, we identified the common elements used in process modeling and the elements less used in process models.
- *Step 03*: we applied two surveys, directly with BPM specialists considering the two possible contexts of BPMN use: process analysts, which consider the application with the user and the regard of the developers, in which they feel the purpose for development.
- *Step 04*: we proposed developing a set of pseudo-algorithms focused on joining more behavior on the existing elements instead of implementing elements absent on BPMS. For example, instead of representing a gateway XOR and another to describe a process instance's instant, we join both features in a unique element. We considered the elements obtained in previous steps (steps 1, 2, and 3).

We compared the methodology proposed in this thesis with researches related to the analysis of BPMS and BPMN (GEIGER et al., 2016; SILVA et al., 2014). There-

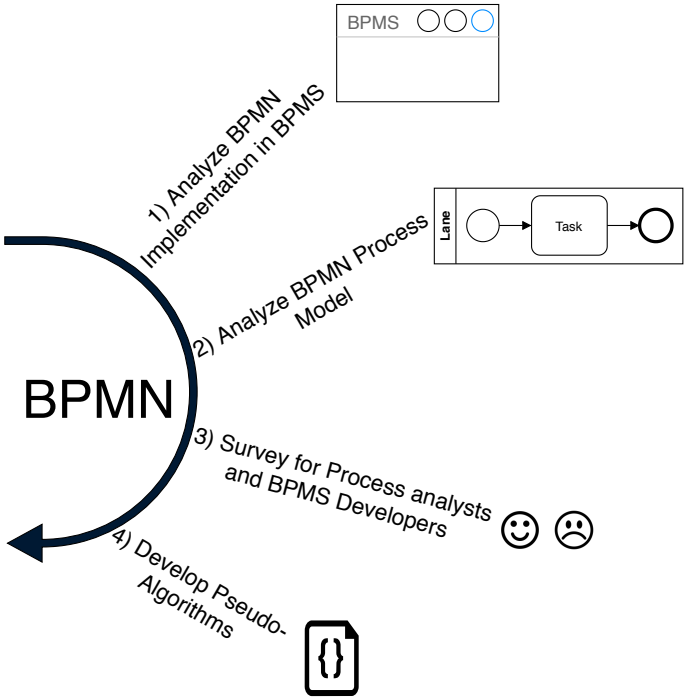


Figure 1.2 – Steps performed in this thesis.
Source: The author.

fore, we demonstrate the effectiveness of the proposed method through the analysis obtained and through the increment of the BPMN elements implementation, using pseudo-algorithms.

1.4 Contributions

In summary, the main contributions of this thesis are:

- A theoretical study regarding the BPMN implementation. With this study, we can provide a state-of-art in BPMS development, including: *(i)* the elements, focused on the implementation; *(ii)* the limitation of the implementation and; (iii) the elements used on the implementation.
- A set of pseudo-algorithms that offers the same behavior with fewer elements.

The contribution of this thesis is to promote a comprehensive study of BPMN and offer pseudo-algorithms expressing the elements. Considering the use of pseudo- algorithms, developers should implement these. However, if the elements do not follow the BPMN specification, the developer can adjust these elements. For example, considering a specific BPMN element, the vendor can verify if the BPMS implements it.

With a missing element, the developer can choose to implement it. Besides, if

there is an implementation of this element, it is possible to verify the element's behavior. When the implementation does not follow the BPMN specification, the developer must adjust it to conform to BPMN.

1.5 Organization of the Text

The outline of this thesis is organized as follows:

- Chapter 2 presents a background about BPM, BPMN and BPMS. Besides, it presents the related works;
- Chapter 3 details the analysis of the implementation of BPMN in BPMS. The protocol used, the BPMS and selected elements are presented, in addition to the results of the analysis of implementation;
- Chapter 4 depicts the analysis of the frequency of use of notational elements in business process models. The implementation performed to extract the elements, the number of elements obtained and an analysis of the results obtained in this stage are presented;
- Chapter 5 presents the questionnaire applied to specialists in the field of BPM. The type of questionnaire presented, the form that was applied, results obtained and analysis of the results are presented;
- Chapter 6 presents the pseudo-algorithms developed for the missing elements. The way of development, verification of the algorithms and discussion on the results obtained;
- Chapter 7 summarizes the approach, discusses its limitations, and concludes the thesis.

2 BACKGROUND AND RELATED WORK

In this chapter, we present the background that supports the approach presented in this thesis. We introduce BPM concepts and discuss works related to the evaluation of BPMN implementations in BPMSs.

2.1 Business Process Management

According to Dumas et al. (2013), BPM aims to organize the performance of work in organizations in order to ensure consistent results and take advantage of opportunities for improvement. BPM is concerned with the management of events, activities, decisions, and invocations of business processes, seeking to add value to organizations and customers.

The application of BPM can make it possible to reduce costs, in addition to contribute to the management of changes in the organization (La Rosa et al., 2011). Based on BPM, it is possible to achieve the strategic alignment between the business and Information Technology (IT) areas, managing technological solutions based on processes that add value to organizations. BPM can be considered a competitive advantage for organizations (FANTINATO; GIMENES; TOLEDO, 2010). BPM has a life cycle. Figure 2.1 presents the life cycle that includes the following phases (DUMAS et al., 2013):

Process identification: In this phase, a business problem is identified (e.g. create a product), processes relevant to the addressed problem are identified, delimited and related to each other. The result of this identification is a new or updated process architecture, which provides an overview of the processes in an organization and their relationships.

Process discovery (also called as-is process modeling): The current state of each of the relevant process is documented, in the form of one or more *as-is* process models. An as-is process model depicts the current state of the business process in the organization. A process model captures the work performed in the organization and how its objectives are achieved (EID-SABBAGH et al., 2012).

Process analysis: Issues associated with the process are identified, documented and quantified, using performance measures. As a result, there is a set of structured questions, which are organized according to their impact, and sometimes according to the effort required to perform a certain action. These questions are used to guide the improvement of business processes.

Process redesign (also called process improvement): In this phase, the goal is to identify changes to the process that would help to solve the issues identified in the process analysis phase. Also, it allows the organization to know the performance goals related to the process. Thus, several options for changes are analyzed and compared considering performance measures.

Process implementation: The changes required to transform the as-is process to the *to-be* process are prepared and performed. It covers two aspects: organizational change management and process automation. Organizational change management refers to the set of activities required to change the way of work of all participants involved in the process. Process automation refers to the use of Information Technology to support the to-be process, such as a BPMS.

Process monitoring and controlling: Once redesigned, and executing the process, relevant data are collected and analyzed to determine whether the process is being performed according to performance measures. Relevant data include bottlenecks, recurring errors, or deviations in behavior; which are identified and corrective measures are applied. Improvement points can be identified, implying that the cycle is repeated continuously.

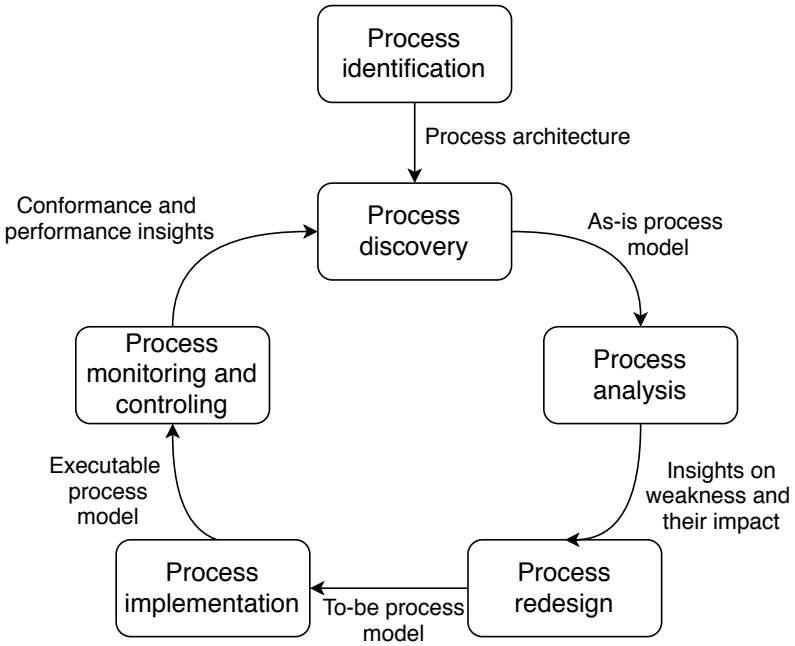


Figure 2.1 – BPM life cycle.
Source: Dumas et al. (2018)

Finally, the BPMS is an extension of classic systems and process management approaches, including *workflows* (WESKE, 2014). Workflows are solutions less expressive than BPMS, focused on a proprietary workflow format. However, the BPMS has a broader reach, from automation and process analysis to process management and organizational

work (AALST, 2011).

2.1.1 Business Process Model and Notation

For the representation of business process models through *as-is* and *to-be* models, it is necessary a notation that allows expressing the goals of the organization. BPMN fulfills its role, allowing the representation of activities, gateways and events. *The Object Management Group* (OMG) is responsible by standardization BPMN (OMG, 2013).

A process model in BPMN is called *process diagram* (PD), and two or more processes interacting with each other form a *collaboration diagram*. Collaboration diagrams consist in a collection of participants and their interaction following set of elements (see Figure 2.2) (ISO, 2013):

- *Flow objects*, used to define the process behavior (they are *activities*, *events* and *gateways*). *(i)* Activities represent the work that a company performs in a process. An Activity can be atomic (activity that could be broken down to a finer level of process detail) or non-atomic (activity compound). The types of Activities that are a part of a Process Model are: Sub-Process and Task, which are rounded rectangles. *(ii)* An Event is something that happens during the execution of a process. These Events affect the flow of the model and usually have a cause (trigger) or an impact (result). There are three types of Events, based on when they affect the flow: *Start*, *Intermediate*, and *End*. *(iii)* A Gateway controls the divergence and convergence of Sequence Flows in a Process. It will determine branching, forking, merging, and joining of paths. Internal markers will indicate the type of behavior control.
- *Data*, used to represent the data consumed or generated by the process (they are *data objects* and *stores/databases*). Data provide information about what Activities require to be performed or what they produce, Data Objects can represent a singular object or a collection of objects. Data Input and Data Output provide the same information for Processes.
- *Connection objects*, used to connect two other elements (they are *sequence flows*). A Sequence Flow is used to show the order in which the Activities will be performed in a Process.
- *Swimlanes*, used to group other elements (they are *pools* and *lanes*). A *Pool* is the graphical representation of a Participant in a Process Diagram. It also acts as a

"swimlane" and a graphical container for partitioning a set of Activities from other Pools, usually in the context of organizations. A Pool may have internal details, in the form of the Process that will be executed. Or a Pool may have no internal details, i.e., it can be a "black box". A *Lane* is a sub-partition within a Process in a Pool, and will extend the entire length of the Process, either vertically or horizontally. A common use of Lanes is to organize and categorize Activities.

- *Message flows*, (*associations* and *data associations*). A Message Flow is used to show the flow of Messages between two Participants that are prepared to send and receive them. In BPMN, two separate Pools in a Collaboration Diagram will represent the two Participants (e.g., Partner Entities and/or Partner Roles).
- *Artifacts*, used to provide additional information about the process. BPMS developers are free to add as many Artifacts as needed (they are *groups* and *annotation*). A *Group* is a graphical representation of elements that are within the same category. The group does not affect the Sequence Flows in a process model. The Category name appears on the diagram as the group label. Categories can be used for documentation or analysis purposes. Groups are one way in which Categories of objects can be visually displayed on the diagram. *Text Annotations* are a mechanism for a modeler to provide additional text information for the reader of a PD.

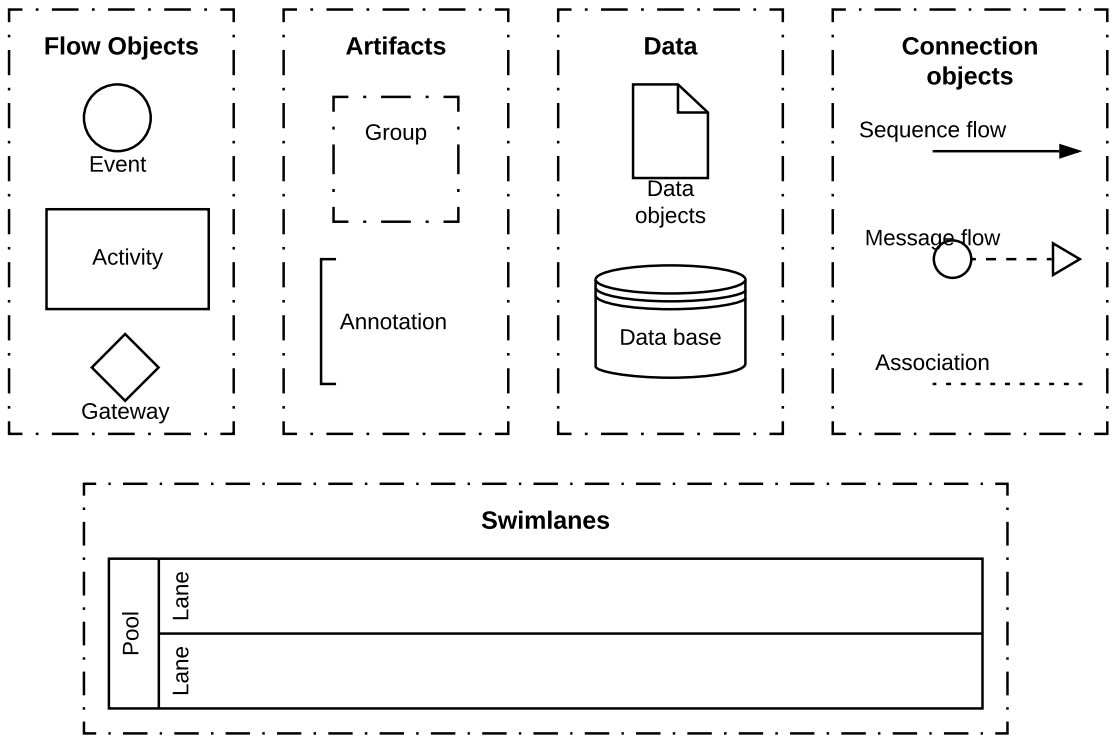
In addition to these elements, we can use extended elements, which are variations of some of the basic elements listed above with specific markers that denote additional or specialized features (e.g., message start event, manual task activity, event-based exclusive gateway, among others).

BPMN also offers conversation diagrams and choreography diagrams. *Conversations* are diagrams centered on the communication of participants, identifying who the participants are and the logical relation of *message exchanges*. Participants are represented by pools and the process flow is not modeled, known as a "black box" (ISO, 2013). *Choreographies* comprise the definition in the expected behavior, basically a procedural contract between process participants (people who participate in the process). While a normal process is modeled within a *pool* (an element used to represent the participant), a *choreography* is based on message exchanges, which involve two or more participants.

As an example of a business process modeled on BPMN, Figure 2.3 depicts a process model for ordered product, adapted from Weske (2014):

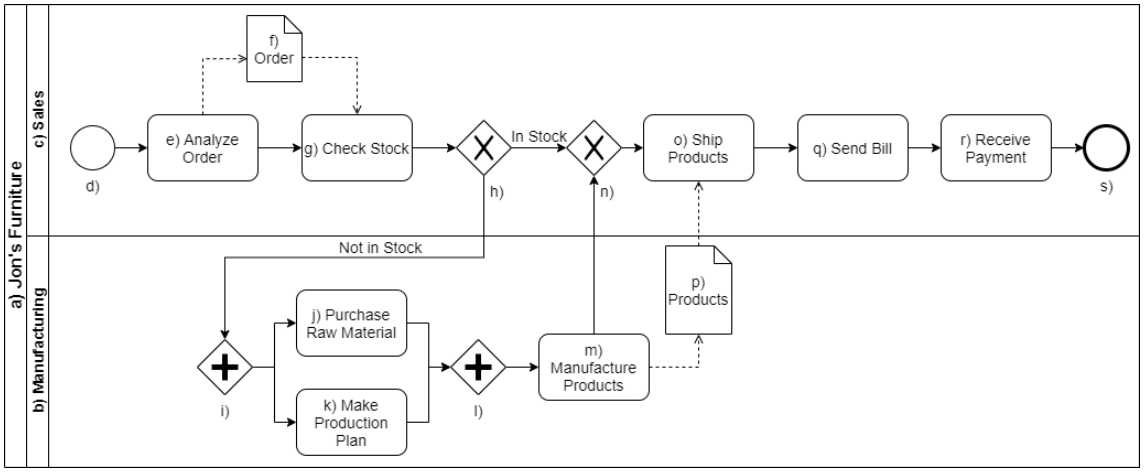
The example presents the main elements of BPMN: events, activities, gateways, and sequence flow. The process model starts with an event *(d)*. A sequence of activities to

Figure 2.2 – Basic set of BPMN elements.



Source: OMG (2013).

Figure 2.3 – Example of BPMN process model.



Source: Weske (2014)

analyze the order *(e)* and check the stock *(g)* before an exclusive getaway *(h)*, representing a deviate. A gateway with the respective marker represents the latter. If the ordered products are in stock, then the lower branch is selected. Otherwise, it is necessary to manufacture the product first; consequently, we need to choose the lower branch, with the label “Not in stock”. To express conditions, we use plain text (“In stock”, “Not in Stock”)

so that humans can easily understand the conditions. The manufacturing part is performed paralleling, represented by a *split* gateway *(i)* and *join* gateway *(l)*. All activities between these elements perform paralleling; in this case, purchase raw material *(j)* and make production plan *(k)*. The process flow waits for all branches to join the gateway and continue the flow. Finally, the branches converge in the exclusive join gateway *(n)* before shipping the products *(o)*, send the bill *(q)*, and receive the payment *(r)* when finishing the process, with end event *(s)*.

Data in processes play an increasingly important role. The example represents the order processed as a data object *(f)*. Data objects can be associated with flow elements, indicating a relationship. In Figure 2.3, there is a data object Order, which is associated to activities analyze order *(e)* and check stock *(g)*. The orientation of the association edge indicates the type of relationship. In our process model, Analyze Order writes the data object, while Check Stock reads it. The same happened in Products *(p)*, where Manufacture Products writes the data object, which is read by Ship Products *(o)*.

Figure 2.3 also represents the two departments of the company modeled, manufacturing *(b)* and sales *(c)*. This representation occurs through the element lane. Receiving and analyzing the order, checking the stock, and deciding about manufacturing the products is also decided by the sales department.

Data in processes plays an increasingly important role. The example represents the order processed as a data object *(f)*. Data objects can be associated with flow elements, indicating a relationship. In the Figure 2.3, there is a data object Order, which is associated to activities Analyze Order *(e)* and Check Stock *(g)*. The orientation of the association edge indicates the type of relationship. In our process model, Analyze Order writes the data object, while Check Stock reads it. The same happened in Products *(p)*, where Manufacture Products writes the data object, which is read by Ship Products *(o)*.

Figure 2.3 also represents the two departments of the company modeled, Manufacturing *(b)* and Sales *(c)*. This representation occurs through the element lane. Receiving and analyzing the order and checking the stock, and deciding about manufacturing the products is also decided by the sales department.

2.1.2 Business Process Management Suite

By representing a process in a BPMS through a modeling notation, e.g., BPMN, one can implement and automate such process. Automation refers to the intention to

automate any part of a process that could be implemented for future execution. As part of the process to be implemented, we consider simple activities as well as the coordination of the whole automated process (KARAGIANNIS, 1995).

Process automation explores knowledge about how different process activities relate to each other. In other words, the types of information systems we consider are sensitive to the process. The process-aware information systems are called Business Process Management Systems (BPMSs). There are other systems with the process, such as Customer Relationship Management (CRM) and Enterprise Resource Planning Systems (ERP). Still, the feature of BPMS exploits an explicit description of a business process to coordinate that process in the form of a process model.

A BPMS is an information system used to implement a process. In this context, an automated process is one that is fully or in part controlled by a software system, controlling the logical and temporal dependencies, as defined in the process model (DUMAS et al., 2013).

To understand how a BPMS works, we can see that a BPMS is somehow similar to a Database Management System (DBMS) (DUMAS et al., 2013). With a DBMS, it is possible to capture company-specific data in a structured way without considering how the exact recovery and storage of the data involved occurs. Standard system installations perform these tasks. Of course, it is necessary to configure the DBMS and fill it with data. It may also be required to adapt the system and its content to actual demands periodically.

2.1.2.1 Automating Business Process

Process automation is a subject addressed in several ways. It generally refers to the intention to automate any conceivable part of the procedural work contained in a business process, from simple operations that are part of a single process activity to the automated coordination of entire and complex procedures (DUMAS et al., 2013).

Mapping processes to the automation level requires modeling details that are notably different from the detail need for communication or analysis. Business-oriented process models are not necessarily accurate and may contain ambiguities. On the other hand, models of executable processes must be precise specifications to be interpreted by a BPMS. There are five steps to transform a business process at the automation level (DUMAS et al., 2013):

- *Identify the boundaries of automation:* we need to identify which parts of the pro-

cess the BPMS can coordinate and which parts it cannot. Commonly, in a process model there are automated, manual, and user tasks. Automated tasks are performed by the BPMS or by an external service, while manual tasks need the intervention of the participants without any software. A user task is between an automated task and a manual task. It is a task performed by a participant with the help of the BPMS worklist (a list with activities to perform, like an email inbox) manager or an external task list manager.

- *Review the manual tasks:* when identifying each task type, we need to check if we can link the manual tasks to the BPMS to maximize the value obtained by the BPMS. Alternatively, we need to isolate these tasks and automate the rest of our process. There are two ways to link a manual task to a BPMS: either implementing it through a user task or through an automated task. Complete the process model: after identifying the process automation boundaries and reviewing the manual tasks, we need to verify that the process model is complete. Often, business-oriented process models skip certain information because modelers feel they are not relevant to the specific modeling purpose. They assume it is common knowledge, or they are not aware of it. However, information that is not relevant in a business-oriented model can be highly pertinent to execute a process model.
- *Bring the process model to an appropriate level of granularity:* there is not necessarily a one-to-one mapping between tasks in a business-oriented model and the corresponding executable model. The BPMS intends to coordinate and manage transfers of work between various resources (human or non-human). Consequently, two or more consecutive tasks assigned to the same resource are candidates for aggregation. If that were the case, BPMS would not add value between these two tasks because it would not manage any transfers.
- *Specify the execution properties:* in the last step, we need to specify how BPMS implements each element. For this purpose, we selected items needed for the execution of elements, such as *(i)* process variables, messages, signals, and errors; *(ii)* task and event variables and their mappings to process variables; *(iii)* service details for service, send and receive tasks, and for message and signal events; *(iv)* code snippets for script tasks *(v)* participant assignment rules and user interface structure for user tasks; *(vii)* task, event, and sequence flow expressions and *(viii)* BPMS-specific properties.

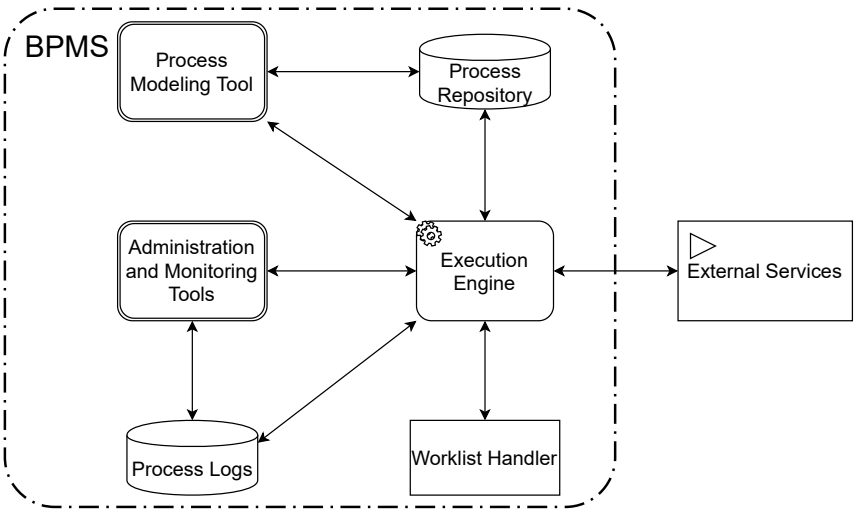
2.1.2.2 Architecture of a BPMS

The architecture determines the main components of a BPMS: the execution mechanism, the process modeling tool, the worklist manipulator, and the administration and monitoring tools. The enforcement mechanism can interact with external services (DUMAS et al., 2013):

- *Execution Engine:* is the central part of a BPMS. The engine provides different functionalities, including: *(i)* the ability to create executable process instances (also named cases), *(ii)* the ability to distribute work to process participants to perform a business process instance from start to end, *(iii)* the ability to automatically retrieve and save data required to perform the process and delegate (automated) activities to software applications across the organization.
- *Process modeling tool:* corresponds to features such as: *(i*) allowing users to create and modify process models. *(ii)* allowing annotating process models with additional data, referring to inputs and outputs, participants, business rules associated with activities, or performance measures related to a process or activity. *(iii)* the ability to save, share and retrieve process models from a process model repository. A process model can be implemented in the engine, allowing their execution. One way to perform the execution is directly from the modeling tool or from the repository.
- *Worklist handler:* this is the component of a BPMS through which process participants *(i)* are offered work items and *(ii)* assigned to them. The execution mechanism controls which work items are past due and makes them available through the worklist handlers of individual process participants. We can imagine a worklist manager, in a BPMS, like an inbox, similar to an e-mail client.
- *External services:* are external applications in the execution of a business process. In many business processes, some activities did not execute entirely in manual mode. Some of these activities can be performed entirely automatically, such as that the execution engine can call an external application, for example, to assess a client's creditworthiness. The external application has to expose a service interface with which the engine can interact.
- *Administration and monitoring tools:* there are tools necessary for administering all operational matters of a BPMS. Consider, as an example, the availability of participants. If someone is not available to work due to illness or vacation, the BPMS should be informed of this fact to avoid allocating work items for this person. Ad-

ministration tools are essential to handle exceptional situations, such as removing outdated work items from the system. Also, the administration tools work to monitor the process. We can use the monitoring tools to monitor running business processes, particularly concerning the progress of individual instances. These tools can aggregate data from different instances, such as average case cycle times or the fraction of delivered too late. The BPMS records the execution of a process model step by step. The data about the actions and the progress of instances can be stored and exported in execution logs. Figure 2.4 depicts how the parts of the architecture interact with each other.

Figure 2.4 – BPMS Architecture.



Source: The author.

2.2 Related Work

We analyzed studies related about to model transformation targeting at execution or simulation models. We sought studies that conducted evaluations of BPMSs with respect to BPMN elements in order to understand the criteria adopted by them during such evaluations as well as the results obtained with the evaluations *per se*. We also sought to identify the main approaches used to obtain a model with more execution details from a process model.

Table 2.1 summarizes the related work discussed in this section. In summary, we found four types of approaches: *(a)* BPMS limitations, i.e., studies whose goal is to eval-

uate overall BPMS limitations; *(b)* model transformation limitations, i.e., studies whose goal is to evaluate limitations on the model transformation; *(c)* model transformation proposals, i.e., studies whose goal is to propose new approaches for model transformation; and *(d)* BPMN specification evaluation, i.e., studies whose goal is to evaluate the BPMN specification.

Table 2.1 – Comparison of related work.

Authors & year	*(a)*	*(b)*	*(c)*	*(d)*
Börger (2012)	X			
Cetinkaya, Verbraeck and Seck (2012)			X	
Peralta et al. (2014)			X	
Geiger, Wirtz and Weberei (2013)		X		
Silva et al. (2014)	X			
Bocciarelli et al. (2014)			X	
Kluza et al. (2015)			X	
Meidan et al. (2017)	X			
Corradini et al. (2018)				X

Source: The author.

Börger (2012) evaluated the BPMN 2.0 specification and found many behaviorally 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 specified 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 notational elements called standard and still are often only partially compatible with each other.

Cetinkaya, Verbraeck and Seck (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 the DEVS are called *atomic models*. These atomic models are defined by 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 the BPMN, there is a corresponding element target in the 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: the number of activities, the total number of precedence dependencies between activities, the connectivity level between activities, the number of split nodes of parallel, the number of XOR-split, the number of OR-split, the number of AND-join, the number of XOR-join and the 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.

Geiger, Wirtz and Weberei (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.

Silva et al. (2014) evaluated a set of BPMSs to identify the most appropriate for organizational characteristics. To this end, these authors applied Multicriteria Decision Aid (MDA) (LAZARTE et al., 2011), an artificial intelligence technique, to help select BPMSs considering the needs for an organization. First, a set of BPMSs were defined, considering all available. Second, in order to identify the goals of the organization, the authors defined a set of criteria, such as simulation, integration, technical support, dynamic form, templates for preparing processes, report, performance, potential tool, flexibility and usability. With these criteria, the MDA technique evaluated the BPMSs, selecting that one best matching the goals.

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, it obtained 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 (Activiti, 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 the translation of a process model into an execution model.

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 showing the BPMS closest to the aims of the BPM life cycle when compared to other BPMSs.

Corradini et al. (2018) proposed 50 guidelines to help modelers improve the understandability of their BPMN process models, facilitating automation. They derived these guidelines on a thorough literature review, which allowed the identification of around 100 guidelines through successive synthesis and homogenization activities. To validate their proposal, they implemented a free and open-source tool, aimed to check the adherence of a model to the guidelines.

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 transform a process model into an executable model, or proposing an approach to transform 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 the 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.

2.3 Final comments

In this chapter, we presented the necessary fundamentals to the understanding of our work. We described the most important concepts: BPM, BPMN and BPMS. Besides, we introduced the related work. Both BPMN and BPMS permeated the entire work, and the related works allow us to identify the state of the art in the BPMN implementation. We ended the chapter with a brief review of the works most relevant to ours.

Comparing the types of the analysis presented in Table 2.1 with this thesis, we can identify that the present work addresses the study of the limitation of BPMS (explained in Chapter 3) and the evaluation of BPMN (explained in Chapter 4). We also found that the focus of related works is on an individual study of either BPMS or BPMN, separately. Furthermore, there was no evidence on the related works of any analysis involving both together.

3 ANALYZING THE IMPLEMENTATION OF BPMN ELEMENTS IN BPMS

This chapter presents the analysis of the implementation of BPMN elements in the BPMS. To perform this analysis, we defined a research protocol, and then we applied this protocol in a group of free BPMS. As a result, we depicted an analysis of the BPMN coverage.

3.1 Research Protocol

This section presents the research protocol we defined for this research. For the elaboration of the protocol, we considered the necessary steps for modeling processes as well as the related works (SILVA et al., 2014; MEIDAN et al., 2017). Our research protocol includes four phases that are explained in this section.

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 BPMN elements are correctly implemented).
4. Scoring the evaluated BPMSs according to which (and how) BPMN elements are implemented.

Figure 3.1 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.

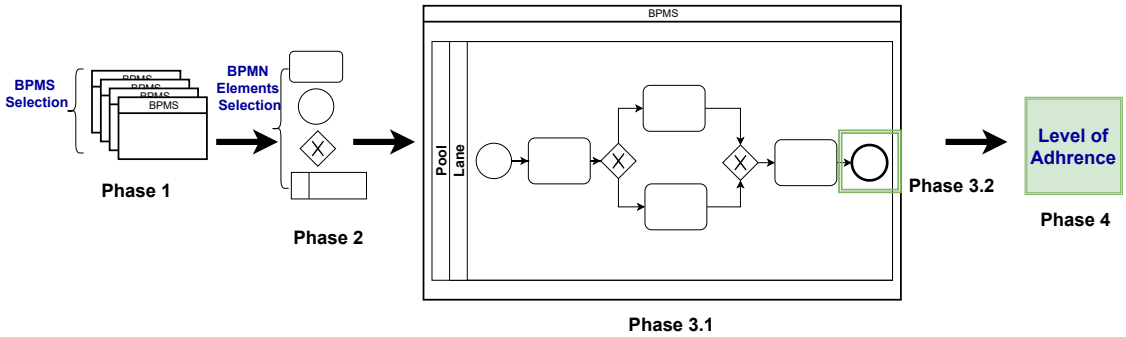


Figure 3.1 – Phases of the evaluation of BPMN elements in BPMSs.
Source: The author.

3.1.1 Selecting the BPMSs

The first phase of the evaluation is the selection of BPMSs. To this end, we considered the available online, and the BPMS listed in the BPMN official website[1], present in the "Implementers" list; *(i)* the BPMS must be free software; and *(ii)* the BPMS must be able to implement and execute processes modeled in BPMN.

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 thesis. 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 the 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. Table 3.1 depicts the candidate BPMSs evaluated. We present the BPMS supplier or developer, the BPMS name, license type and whether the BPMS supports process implementation and execution.

Table 3.1 – Candidate BPMSs evaluated.

N.	Supplier/ developer	Name	License type	Implementation and execution support
1	Kaisha	Active.Net Workflow Engine	Commercial	Yes

Continued on next page

[1] http://bpmn.org

Table 3.1 – continued from previous page

N.	Supplier/ developer	Name	License type	Implementation and execution support
2	Kaisha	ActiveModeler	Commercial	No
3	Informatica	ActiveVOS	Commercial	Yes
4	BOC Group	Adonis	Commercial	Yes
5	Alfresco	Alfresco Activiti BPM	Commercial	Yes
6	Altova	Altova UMovel	Commercial	Not informed
7	Barium	Barium Process Modeler	Commercial	No
8	BP1	BeeBPM	Commercial	No
9	Bizagi	Bizagi Modeler	Free	No
10	Bizagi	Bizagi Studio	Commercial	Yes
11	BonitaSoft	Bonita BPM	Free	Yes
12	Borland	Borland Together	Commercial	Not informed
13	SAP	BPM & Integration Solutions	Commercial	Not informed
14	TransWare	BPM-X	Commercial	Yes
15	Appian	Business Process Management Suite	Commercial	Yes
16	No Magic	Cameo Business Modeler	Commercial	Yes
17	Camunda	Camunda BPM	Free	Yes
18	Semture GmbH	Cubetto	Commercial	No
19	Aurea	CX Process	Commercial	Yes
20	Interfacing	Digital Enterprise Management System	Commercial	Yes
21	Soyatec	eBPMN	Free	No (BPMN 1.0)
22	Sparx Systems	Enterprise Architect	Commercial	Yes
23	IBM	IBM Process Designer	Commercial	Yes
24	Software AG	iBPMS	Commercial	Yes
25	iGrafx	IGrafx	Commercial	Yes

Continued on next page

Table 3.1 – continued from previous page

N.	Supplier/ developer	Name	License type	Implementation and execution support
26	MID GMbH	Innovator for Buiness Analysts	Commercial	Yes
27	Intalio	Intalio BPMS	Free	Not informed
28	Intellior	Intellior BPM	Commercial	Not informed
29	Fujitsu	Interstage Business Process Management	Commercial	Not informed
30	Intellivate	IYOPRO	Commercial	Yes
31	RedHat	jBPM	Free	Yes
32	KnowGravity	KnowEnterprise	Commercial	Yes
33	No Magic	Magic Draw	Commercial	Yes
34	ModelFoundry	ModelFoundry	Free	No
35	OpenText	OpenText Process Suite	Commercial	Yes
36	Oracle	Oracle BPM	Commercial	Yes
37	Oryx	Oryx Editor	Commercial	Yes
38	Pectra	Pectra BPM Sabvia	Not informed	Yes
39	SAP	Process Orchestration	Commercial	Yes
30	QPR Software	QPR ProcessDesinger	Commercial	Yes
41	Rapilabs	Rigrr BPMN Editor	Free	No
42	Rocket	Rocket API	Commercial	Yes
43	Santeon	Santeon BPM	Commercial	Not informed
44	Signavio	Signavio Process Editor	Commercial	No
45	Wondeware	Skelta BPM	Commercial	Yes
46	TIBCO Software	TIBCO ActiveMatrix	Commercial	No
47	Trisotech	Trisotech	Commercial	Yes
48	Microsoft	Visio 2013	Commercial	Yes
49	Visual Paradigm	Visual Paradigm Modeler	Commercial	No

Continued on next page

Table 3.1 – continued from previous page

N.	Supplier/ developer	Name	License type	Implementation and execution support
50	ITP Commerce	Vizi BPM Suite	Commercial	No
51	W4 Software	W4 Web Modeler	Commercial	Yes
52	Web Ratio	WebRatio BPM Plantaform	Free	Yes
53	Lanner	Witness System Simulation	Commercial	Yes
54	Work token	Work Token	Commercial	Not informed

Figure 3.2 summarizes the selection of BPMSs following the criteria application. From the total list of 54 BPMSs, only nine are free (i.e., 16.6% of the total); and from these, only four support process implementation and execution (i.e., 7.40% of the total). To archive this amount, we analyzed each BPMS website, downloaded and analyzed what the necessary licenses for execution were.

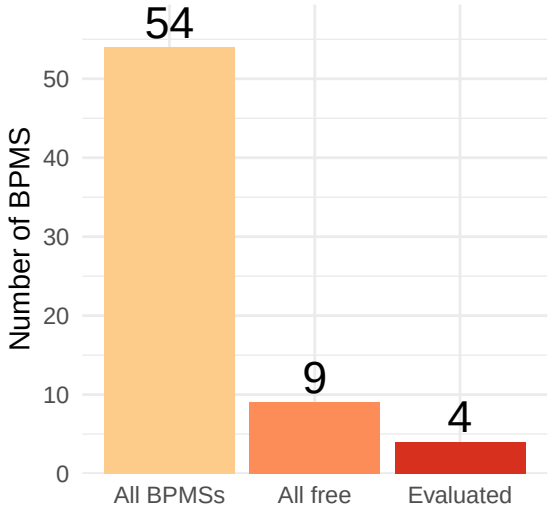


Figure 3.2 – Results of BPMS selection.
Source: The author.

Finally, Table 3.2 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.

Table 3.2 – 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

Source: The author.

3.1.2 Selecting the BPMN Elements

In the second phase, we selected the set of BPMN 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. Collaboration diagrams are those more commonly used by BPMN practitioners (OMG, 2013), 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, we selected 83 elements, from BPMN specification, considering both the basic elements and their extended versions. The list of all selected BPMN elements are presented as follows:

- *Flow objects*:
  - *Activities*: abstract atomic task, service task, send task, receive task, user task, manual task, business rule task. script task, sub-process, transaction.
  - *Events*:
    - *Start – standard*: abstract, message, timer, conditional, signal, multiple and parallel multiple.
    - *Start – event sub-process interrupting*: message, timer, escalation, conditional, error, compensation, signal, multiple and parallel multiple.
    - *Start – event sub-process non-interrupting*: message, timer, escalation, signal, multiple and parallel multiple.
    - *Intermediate – catching – standard*: message, timer, conditional, link, signal, multiple and parallel multiple.
    - *Intermediate – catching – boundary interrupting*: message, timer, escalation, conditional, error, cancel, signal, compensation, multiple and parallel multiple.
    - *Intermediate – catching – boundary non-interrupting*: message, timer,

escalation, conditional, signal, multiple and parallel multiple.

- *Intermediate – throwing*: abstract, message, escalation, compensation, link, signal and multiple.
- *End events*: abstract, message, escalation, error, cancel, compensation, signal, multiple and terminate.

- *Gateways*: exclusive, parallel, inclusive, event-based, complex, exclusive event-based (instantiate) and parallel event-based (instantiate).

- *Data*: object, collection, input, output and store (database).
- *Connection objects*:
  - *Sequence flows*: normal/uncontrolled, conditional and default.
  - *Message flows*: standard, initiating message and non-initiating message.
  - *Associations*: non-directional, directional, bi-directional and compensation.
  - *Data associations*: standard and directed data association.
- *Swimlanes*: pools and lanes.
- *Artifacts*: groups and text annotations.

3.2 Performing the Analysis

Once the BPMS and BPMN elements to be evaluated were selected, we proceeded with the third phase of the research protocol (cf. subSection 3.1) and conducted the evaluation itself. To this end, we evaluated the BPMN implementations in two aspects:

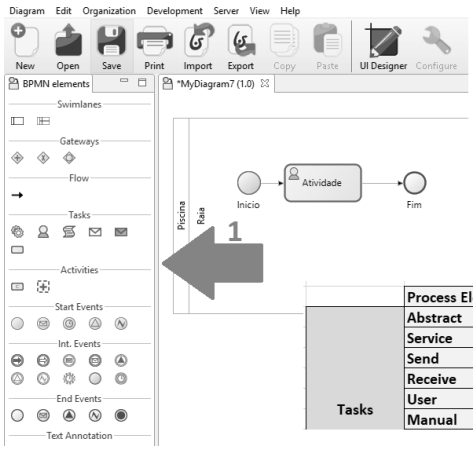
1. Verifying whether the BPMN elements are implemented in the BPMSs.
2. Verifying whether the implemented BPMN elements are correctly implemented, comparing their implementations with BPMN specification.

3.2.1 Identifying Implemented BPMN Elements

For this first evaluation aspect, we created a process model on each BPMS under evaluation and tried to add to each the BPMN elements selected for evaluation (cf. Section 3.1.1). Thus, we sought 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.

Figure 3.3 illustrates the procedure for identifying the implemented BPMN elements. We verified the editor of each BPMS (cf. “Step 1”). 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 into a script task or a standard event into a message event. If there was no reference to the evaluated BPMN element in the BPMS editor or in the base element property, we concluded that the corresponding BPMS does not implement it (cf. “Step 2”). 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. Appendix B present the spreadsheet with full analysis.



	Process Element/ Tool	Bonita BPM	WebRatio BPM	Camunda BPM	jBPM
Tasks	Abstract	X		X	X
	Service	X	X	X	X
	Send	X		X	X
	Receive	X		X	X
	User	X	X	X	X
	Manual		X	X	X

Figure 3.3 – Example of how identify a element implementation.
Source: The author.

3.2.2 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 3.2.1.

Figure 3.4 depicts an example of a process model used in the evaluation, considering the *XOR split-gateway element*. For this element, the BPMN specification states that “Each token arriving at any incoming sequence flow activates the gateway and is routed to exactly one of the outgoing sequence flow. To determine the outgoing sequence flow

that receives the token, the conditions are evaluated in order. The first condition that evaluates to true, determines the sequence flow the token is sent to. No more conditions are henceforth evaluated."

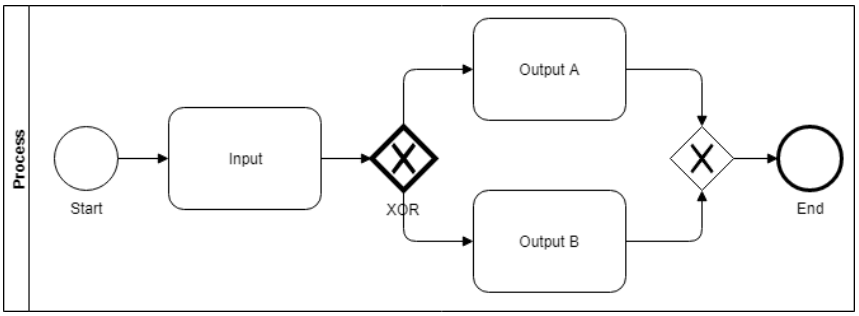


Figure 3.4 – Example of a process model used in the evaluation.
Source: The author.

We added several elements needed to evaluate the XOR element in the process model: the task "Input" defines the attributes used in the conditions to be evaluated for the XOR element while "Output A" and "Output B" define possible outgoing flows that the token can be sent to. To evaluate the behavior of the XOR element, we defined different conditions to evaluate if the token is correctly addressed.

3.2.3 Scoring Evaluated BPMSs

We defined a plan for specifying a score to the following groups of elements: activities, events, gateways, data, connection objects, swimlanes and artifacts. This procedure allowed a systematic evaluation of the implementation of these BPMN elements and a homogeneous comparison between the BPMSs. With this score, we were able to define how close the element is to the specification, with 0 showing that the implemented element is different from the implementation and 10 is the same as defined in the specification. 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 condi-

tions 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 3.1) 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 3.1, n represents the number of elements in the group and S represents the individual score of the element.

$$GS = \frac{\sum_{i=1}^{n} S_i}{n} \tag{3.1}$$

- *Normalized group score:* for all element group scores previously calculated (cf. Equation 3.1), we calculated (cf. Equation 3.2) 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 3.2, n represents the number of element groups, GS represents the group score and $max(GS)$ represents the highest among the group scores.

$$GS_{norm_i} = \frac{GS_i}{max(GS)} x10, \forall i \in 1..n \tag{3.2}$$

- *BPMS score:* for each BPMS, we calculated (cf. Equation 3.3) the $BPMS_{score}$ by summing all the group scores of the BPMS and dividing the result by the number of element groups. In Equation 3.3, n represents the number of BPMN element groups and GS_{norm} represents the normalized group score.

$$BPMS_{score} = \frac{\sum_{i=1}^{n} GS_{norm_i}}{n} \tag{3.3}$$

Table 3.3 summarizes the evaluation results. Each row represents a group score (cf. Equation 3.1) 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 3.3) row depicts the final scores obtained by the BPMS, considering the six groups above.

Table 3.3 – 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**

Source: The author.

3.3 Result Analysis

Our study found that support for BPMN elements in BPMSs is limited considering the four BPMSs evaluated. From the 83 elements evaluated, only 27 are implemented by at least one of the BPMSs (corresponding to 34.18%). In addition, we 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 3.5 presents more information about the results obtained by each group of BPMN elements. We split events elements 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.

Figure 3.6 depicts the results grouped by the BPMS, considering the elements that have an implementation in the BPMS. We present an analysis showing the number of elements fully implemented, partially implemented and not implemented according to 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 83 elements, 35 are fully implemented (42.17%), 6 are partially implemented (7.23%) and 9 are not implemented (10.84%).

Regarding the scope of implementation, we can conclude that although the BPMS

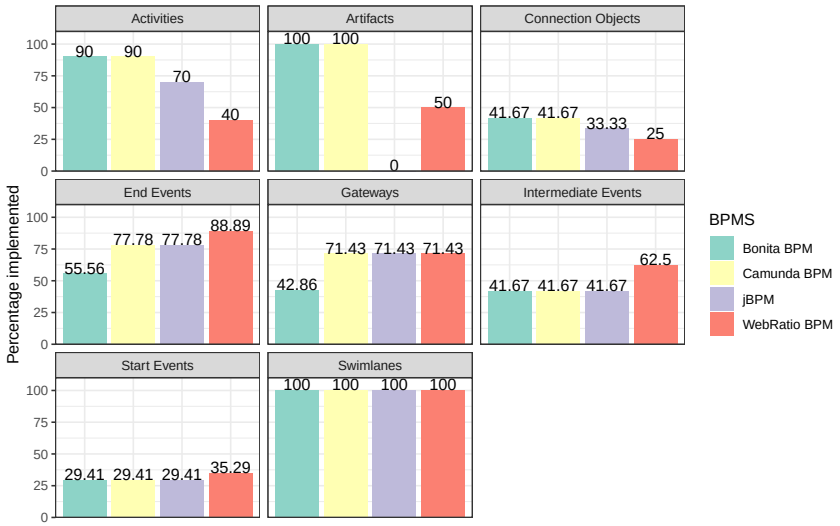


Figure 3.5 – Percentage of implemented BPMN elements by element group and by BPMS.
Source: The author.

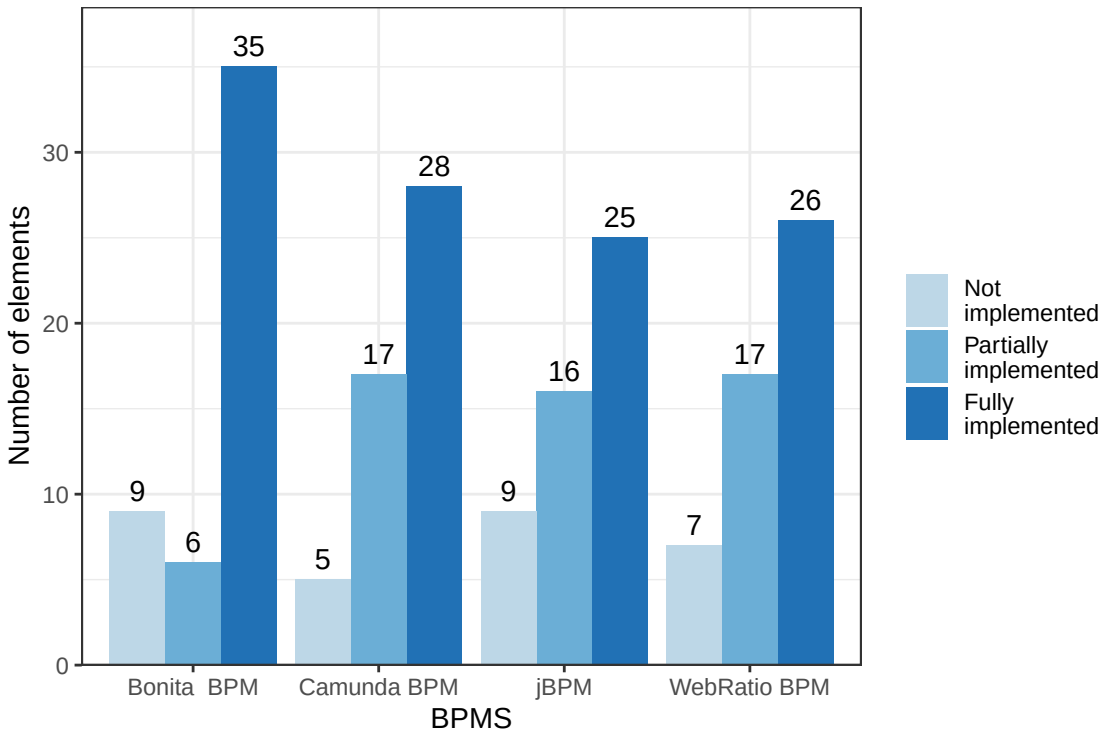


Figure 3.6 – Number of implemented BPMN elements by BPMS.
Source: The author.

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, 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 in terms of implementation.

3.4 Final Comments

We presented an analysis that allows the identification of the reality of implementation of elements in BPMSs. We realized that the developers focus on the implementation of the same notational elements. Thereby, we can infer that their focus is on implementing elements that are fully functional than elements that do not offer all the necessary resources for execution. However, it is essential to verify the use of these elements in process models, identifying some aspects, such as *(i)* frequency of elements used in models; *(ii)* if there is, what elements are avoided. We present this analysis in chapter 4.

4 ANALYZING THE FREQUENCY OF BPMN ELEMENTS IN PROCESS MODELS

Considering the need to understand the BPMN implementation of elements, it is necessary to establish means to analyze the distribution of elements in process models. This chapter aims to describe the analysis of 186 processes modeled in BPMN, obtained from 3 different sources. With this analysis, it is possible to evaluate and identify a set of notational elements most used in process modeling. This analysis is necessary to show the focus of the elements used. We defined criteria to select the models based on our need to identify the BPMN specification models to perform the analysis. Then, we develop an analyzer in Python language to generate data in *cvs* format with the number of elements used in process models. With cvs obtained, we performed an analysis with the achieved results.

The main stages of our analysis are (Figure 4.1): define criteria to select the BPMN models to be analyzed; select the models; develop the analysis, automate the way to extract the elements from models; and finally, analyze the achieved results.

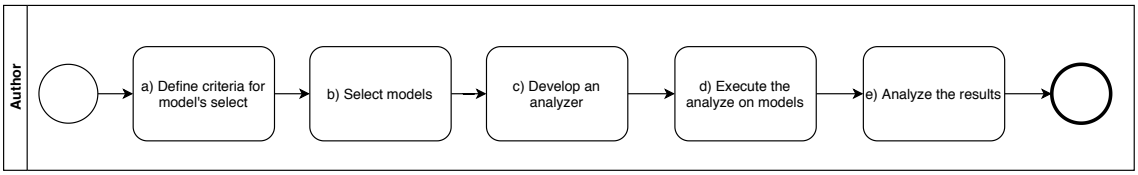


Figure 4.1 – BPMN models analyze stages.
Source: The author.

4.1 Selecting the BPMN models

As the first step of our analysis, we selected the models to be analyzed. We search for models in the database of articles, such as ACM digital library, IEEE Explore, google scholar and research gate.

The criteria for selecting the process models were as follows: *(i)* process modeled in BPMN 2.0, including *(ii)* real models, identifying the elements that are used in real-world; *(iii)* the model should be in the .bpmn extension, a format defined in BPMN specification; *(iv)* and it should allow downloading the models, so that they can be analyzed.

We searched for “business process models” on the database of articles on the in-

ternet and looked for any information about the process models' download. As a result, we found process models to download in three sources: First, in the research gate, we obtained a link to a website, named as *RePROSitory*, that provides process models to download[1]. We applied the criteria for selecting, and we obtained 62 process models. Second, we found a master thesis that turns available a database of processes modeled (FRIEDRICH, 2010). Focused on generating business process models, from Natural Language input, we could obtain 66 models in .bpmn format. Third, we obtained 58 models from a project to automate the business process from a Federal University of Brazil. As a result, Figure 4.2 summarizes the process models used in our analysis, organized by repository.

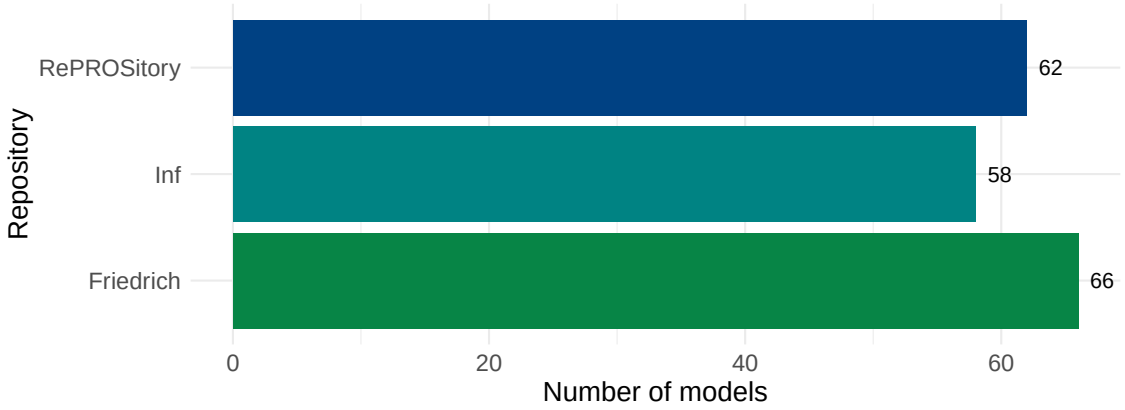


Figure 4.2 – Repositories of process models used in our analysis.
Source: The author.

4.2 BPMN models Analyzer

The next step in our analysis is to obtain the models to identify the elements used in the modeling. We developed a process model analyzer, which was given a process model in *.bpmn* as input. As output, we obtained a *cvs* file with the list of elements and respective amounts used to model the input process.

As first step to develop the analyzer, we need to understand the format of the .bpmn file. The .bpmn extension represents a file on XML format, and the BPMN specification defines its structure. There is a definition for names of elements in this specification, in addition to the XML structure terminology. In addition to the .bpmn extension, there are other formats, such as *json* and proprietary formats, defined by each vendor. However, in these specific formats (*json*, proprietary), there is no guarantee that they implement

[1] Available in: <https://pros.unicam.it:4200/home>

BPMN under the BPMN standard. Therefore, we decided to choose the *bpmn* files to follow the specification.

BPMN specification defines the XML of elements through an XML Schema Definition (XSD), a recommendation of the *World Wide Web Consortium* (W3C)[2]. The aim of the XSD is to define the legal building blocks of an XML document, describing: the XML elements and attributes that should appear in a file, the number and order of child elements, data types for elements and attributes , default and fixed values of elements and attributes (WORLD WIDE WEB CONSORTIUM, 2012). Listing 4.1 depicts the XSD for the element Sequence Flow (OMG, 2013).

```
1  <xsd:element name="sequenceFlow" type="tSequenceFlow" substitutionGroup
       ="flowElement"/>
2  <xsd:complexType name="tSequenceFlow">
3    <xsd:complexContent>
4      <xsd:extension base="tFlowElement">
5        <xsd:sequence>
6          <xsd:element name="conditionExpression" type="tExpression"
             minOccurs="0" maxOccurs="1"/>
7        </xsd:sequence>
8        <xsd:attribute name="sourceRef" type="xsd:IDREF" use="required"/>
9        <xsd:attribute name="targetRef" type="xsd:IDREF" use="required"/>
10       <xsd:attribute name="isImmediate" type="xsd:boolean" use="
           optional"/>
11     </xsd:extension>
12   </xsd:complexContent>
13 </xsd:complexType>
```

Listing 4.1 – XSD for element Sequence Flow (OMG, 2013)

In the example presented in Listing 4.1, there are items used to define the XML. For example, the element “name” (see line 1) defines the name the XML component has; type defines the class that the element belongs to; the attributes (lines 8, 9, and 10) defines the child of this XML element. In this case, two attributes are required (that will be named as “sourceRef” and “targetRef”, respectively), and one is optional (that will be named as “isImediate”). When generating the XML based on this XSD, this structure will always have two attributes to define the source and the target of sequence flow. Listing 4.1 shows an example of XML generated.

```
1        <bpmn:sequenceFlow id="SequenceFlow_1vyljg2" sourceRef="
```

[2]Available in <https://www.w3.org/TR/xmlschema11-1/>

```
Task_0bamfxb" targetRef="EndEvent_11oaveo" />
```

Listing 4.2 – Example of XML generate based on XSD

In Listing 4.1, we obtained an XML that represents a sequence flow. “Id” defines a value of internal control, “sourceRef” and “targetRef” refer to the IDs of the elements that are connected to this sequence flow. Listing 4.3 depicts a complete XML file about a process model.

```
<?xml version="1.0" encoding="UTF-8"?>
<bpmn:definitions
   xmlns:bpmn="http://www.omg.org/spec/BPMN/20100524/MODEL"
   xmlns:bpmndi="http://www.omg.org/spec/BPMN/20100524/DI"
   xmlns:di="http://www.omg.org/spec/DD/20100524/DI"
   xmlns:dc="http://www.omg.org/spec/DD/20100524/DC" id="
      Definitions_1djmi2u" targetNamespace="http://bpmn.io/schema/bpmn
      " exporter="Camunda Modeler" exporterVersion="2.2.4">
   <bpmn:collaboration id="Collaboration_0leorr3">
      <bpmn:participant id="Participant_0q99x44" name="BET" processRef=
         "Process_1" />
   </bpmn:collaboration>
   <bpmn:process id="Process_1" isExecutable="true">
      <bpmn:startEvent id="StartEvent_1agtejh">
         <bpmn:outgoing>SequenceFlow_0xusxma</bpmn:outgoing>
      </bpmn:startEvent>
      <bpmn:task id="Task_1ddn959" name="Select tweet from repository">
         <bpmn:incoming>SequenceFlow_0xusxma</bpmn:incoming>
         <bpmn:outgoing>SequenceFlow_0n7qppw</bpmn:outgoing>
      </bpmn:task>
      <bpmn:task id="Task_0bamfxb" name="Send tweet">
         <bpmn:incoming>SequenceFlow_0n7qppw</bpmn:incoming>
         <bpmn:outgoing>SequenceFlow_1vyljg2</bpmn:outgoing>
      </bpmn:task>
      <bpmn:endEvent id="EndEvent_11oaveo">
         <bpmn:incoming>SequenceFlow_1vyljg2</bpmn:incoming>
      </bpmn:endEvent>
      <bpmn:sequenceFlow id="SequenceFlow_1vyljg2" sourceRef="
         Task_0bamfxb" targetRef="EndEvent_11oaveo" />
      <bpmn:sequenceFlow id="SequenceFlow_0n7qppw" sourceRef="
         Task_1ddn959" targetRef="Task_0bamfxb" />
      <bpmn:sequenceFlow id="SequenceFlow_0xusxma" sourceRef="
         StartEvent_1agtejh" targetRef="Task_1ddn959" />
```

```
28    </bpmn:process>
29    <bpmndi:BPMNDiagram id="BPMNDiagram_1">
30       <bpmndi:BPMNPlane id="BPMNPlane_1" bpmnElement="
             Collaboration_0leorr3">
31          <bpmndi:BPMNShape id="Participant_0q99x44_di" bpmnElement="
                Participant_0q99x44">
32             <dc:Bounds x="227" y="57" width="601" height="274" />
33          </bpmndi:BPMNShape>
34          <bpmndi:BPMNShape id="StartEvent_1agtejh_di" bpmnElement="
                StartEvent_1agtejh">
35             <dc:Bounds x="315" y="187" width="36" height="36" />
36          </bpmndi:BPMNShape>
37          <bpmndi:BPMNShape id="Task_1ddn959_di" bpmnElement="
                Task_1ddn959">
38             <dc:Bounds x="401" y="165" width="100" height="80" />
39          </bpmndi:BPMNShape>
40          <bpmndi:BPMNEdge id="SequenceFlow_0xusxma_di" bpmnElement="
                SequenceFlow_0xusxma">
41             <di:waypoint x="351" y="205" />
42             <di:waypoint x="401" y="205" />
43          </bpmndi:BPMNEdge>
44          <bpmndi:BPMNShape id="Task_0bamfxb_di" bpmnElement="
                Task_0bamfxb">
45             <dc:Bounds x="551" y="165" width="100" height="80" />
46          </bpmndi:BPMNShape>
47          <bpmndi:BPMNEdge id="SequenceFlow_0n7qppw_di" bpmnElement="
                SequenceFlow_0n7qppw">
48             <di:waypoint x="501" y="205" />
49             <di:waypoint x="551" y="205" />
50          </bpmndi:BPMNEdge>
51          <bpmndi:BPMNShape id="EndEvent_11oaveo_di" bpmnElement="
                EndEvent_11oaveo">
52             <dc:Bounds x="701" y="187" width="36" height="36" />
53          </bpmndi:BPMNShape>
54          <bpmndi:BPMNEdge id="SequenceFlow_1vyljg2_di" bpmnElement="
                SequenceFlow_1vyljg2">
55             <di:waypoint x="651" y="205" />
56             <di:waypoint x="701" y="205" />
57          </bpmndi:BPMNEdge>
58       </bpmndi:BPMNPlane>
59    </bpmndi:BPMNDiagram>
```

```
60 </bpmn:definitions>
```

Listing 4.3 – Example of a BPMN file. Source: Author.

In Listing 4.3, lines 2 - 6 bring the definition of model, such as the BPMS of origin, version and links for BPMN specification, for example. Line 8 presents the first element of BPMN: the lane, named as participant (see tag ““bpmn:participant”). Lines 10 - 28 (controlled by tag “<bpmn:process>”) bring the definition of all elements in the process models. We can identify this definitions trough the tags identifying the elements, such as “bpmn:task”, “bpmn:endEvent” or “bpmn:sequenceFlow”. Lines 29 - 59 (controlled by tag “<bpmndi:BPMNDiagram>”) show the data about visual information such as width, height, position on the screen, among other information.

Understanding the structure of XML is essential to identify how the process model file works to develop the analyzer. From this understanding, we could find that to know if an element is present on the model, we need to see all the tags child present in the “<bpmn:process>” tag. We then decided that with an XML file from a business process, we could generate a csv file, listing the elements presents in a model. With this csv, we can perform analyses about the process database.

To implement the analyzer, we used the Python programming language. The code is available on GitHub[3]. It has dependencies on the following libraries: *pandas*, *bs4*, *lxml* and *PySimpleGUI*. Pandas is used to store intermediate information and deliver the output file in csv format; bs4 and lxml allow us to navigate the .bpmn files with legible code; and PySimpleGUI is used to provide a more user-friendly interface, again with legible code. The analyzer provides an interface for interaction, in which it was possible to select a model or files from a directory. Then, by clicking on the “ok” button, the csv file is generated with the obtained data. Figure 4.3 shows the extractor interface.

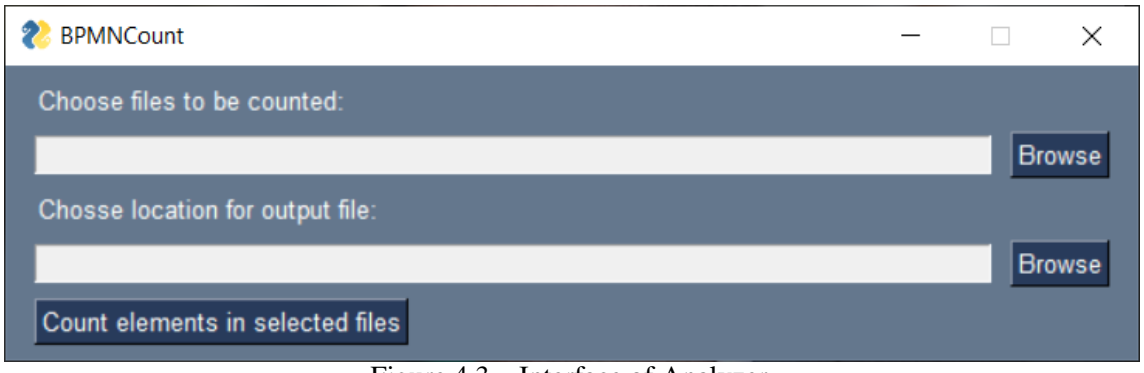


Figure 4.3 – Interface of Analyzer.
Source: The author.

[3]Available on github: <https://github.com/carloshabekost/BPMNAnalysis>

To generate the csv used in our analysis, we processed the data about models in two steps: *(i)* we separated each repository (Friedrich, Inf, and RePROSitory) in different directories[4]. Then, we executed the analyzer, selecting the models in each directory, and generating the csv; *(ii)* afterward, we unified the results obtained in a single csv, aimed to be used in our analysis.

Figure 4.4 shows an example[5] about how the data is organized, in a csv file (Database of processes used as an example: RePROSitory). Each line describes the element that appeared in the models analyzed. Each column describes the marker that the element can have. Markers are events that could be attached in an element. When an element performs its action, the event attached performs the action as well. The numbers describe the occurrence of each element in the analyzed model. For example, in Figure 4.4, "StartEvent" with value "49" indicates that we find this element in 49 analyzed models.

element/marker	regular	messageEventDefinition	timerEventDefinition	linkEventDefinition
startEvent	49	8	1	0
intermediateCatchEvent	0	6	6	9
intermediateThrowEvent	0	7	0	9
endEvent	51	7	0	0
exclusiveGateway	40	0	0	0
parallelGateway	21	0	0	0
inclusiveGateway	6	0	0	0
task	61	0	0	0
sendTask	1	0	0	0
receiveTask	0	0	0	0
userTask	2	0	0	0

Figure 4.4 – Example of data obtained in repository "RePROSitory".
Source: The author.

After finding the csv file for each repository analyzed, we refined the results in step *(ii)*. In this case, we grouped the results obtained for each element and sum the counts obtained. Considering Figure 4.4 as an example, we summed all occurrence of element "StartEvent", resulting "58" (we could see this result in Table 4.1, column "RePROSitory" and line "StartEvent"). We applied this adjustment in all csv reached for the three repositories (Friedrich, Inf and RePROSitory). Table 4.1 depicts the final csv obtained.

[4] Available in <https://github.com/carloshabekost/BPMNAnalysis/tree/master/modelos>
[5] Full version available in:
<https://carloshabekost.github.io/BPMNAnalysis/data/RePROSitory.html>
<https://carloshabekost.github.io/BPMNAnalysis/data/inf.html>
<https://carloshabekost.github.io/BPMNAnalysis/data/friedrich.html>

Finally, once we had obtained the final csv, we performed some statistics to understand the distribution of elements in the analyzed models and to know if the repositories could have a similarity in the choice of elements.

Table 4.1 – Final CSV with data from repository.

group	Friedrich	Inf	RePROSitory
startEvent	76	71	58
intermediateCatchEvent	39	33	21
intermediateThrowEvent	16	22	16
endEvent	72	62	60
exclusiveGateway	47	54	40
parallelGateway	13	33	21
eventBasedGateway	11	4	0
inclusiveGateway	10	7	6
task	61	61	70
sendTask	15	23	5
receiveTask	4	19	4
userTask	3	9	2
businessRuleTask	0	0	1
manualTask	1	17	0
serviceTask	5	5	2
scriptTask	1	1	1
callActivity	3	2	0
subProcess	14	19	13
dataObject	9	9	19
dataStoreReference	1	5	2
boundaryEvent	12	8	5
textAnnotation	6	12	14
laneSet	21	37	14
sequenceFlow	66	58	62
participant	43	57	28
messageFlow	31	31	14
lane	21	37	14
dataInputAssociation	8	5	17
dataOutputAssociation	8	14	14

Source: The author.

4.3 Analyzing the data found

In this section, we present an analysis of the variance in the three repositories analyzed. However, in our research, we need to consider a lot of elements to see the similarity. In this case, we are interested in techniques that allow identifying such similarity. Aiming to see the resemblance, we considered the ANOVA technique (BOX; HUNTER;

HUNTER, 2005) because we have many BPMN elements to consider.

ANOVA is a statistical test used in experimental data. Typically it is applied on samples with many factors to be analyzed. A test is considered statistically significant if deemed unlikely to have occurred by chance, assuming the null hypothesis is truth. A statistically significant result, when a probability (p-value) is less than a pre-specified threshold (significance level), justifies the rejection of the null hypothesis, but only if the prior probability for the null hypothesis is not high (BOX; HUNTER; HUNTER, 2005; MONTGOMERY, 2007).

Analyzing the results obtained on Table 4.2, we can see the variance of elements (seeing in the group level) in the repositories explored. Considering the p-value that we defined as limited (0.05), we can conclude that the elements are similar (connection objects and events have a higher value than 0.05). This shows that there is a tendency to use the same elements, regardless of the model analyzed. In other words, regardless of the BPMS used for process modeling, there is a tendency to use the same elements.

Table 4.2 – Anova Test for element groups.

BPMN element group	Diff	Lower	Upper	p-Value
Activities	-57.33	-104.45	-10.20	0.011
Events	-17.66	-64.79	29.45	0.286
Gateways	-40.33	-87.45	6.79	0.023
Connection objects	-23.00	-70.12	24.12	0.493
Artifacts	-56.00	-103.12	-8.87	0.014
Swimlanes	-36.66	-103.12	-8.87	0.014

Source: The author.

4.4 Final Comments

As presented in this chapter, it is possible to conclude that, even with the variations in specific elements, such as Send Task, Receive Task, exclusive gateway, observe that there is a tendency for users to use the same notational elements. It may be evidence that these elements are already sufficient to express a process model.

Therefore, we verified that in addition to analyzing the models presented in this chapter, it is necessary to see other aspects to answer which are the elements used in process design. In this case, it is essential to know the user's acceptance and understand his preferences and difficulties in choosing the notational elements.

5 A SURVEY TO IDENTIFY THE USE OF BPMN BY USERS

This chapter presents a survey about the user's perspective, considering process analysts and BPMS developers. We decided to focus only on this audience because these are experts in BPMN and know the limitations and aspects that must be improved regarding the notation.

Applying a survey in this thesis is essential to avoid the risk of not solving a real problem if we do not consider the notation user target. Finally, users do not always know what they want until they are asked directly (FØLSTAD; LAW; HORNBÆK, 2012; MAYRING, 2004).

We decided to apply a form with specific questions considering the feedback of users regarding BPMN to perform our survey. To spread the survey, we used google forms — a questionnaire to identify the perspective of using BPMN.

To identify the questions to be applied in the survey, we first analyzed which information we needed to identify from users. As the audience of our analysis, we focused on people with solid expertise in BPMN.

To develop the surveys, we applied google forms. The propagation and sharing of the surveys occurred between October 2019 and February 2020. To reach the survey audience, we contact BPMS developers and the process modeling consulting organization through email.

The main stages of our analysis are (Figure 5.1): *(a)* define target to receive our survey; *(b)* define questions for the survey; *(c)* elaborate a structure to apply the survey; *(d and e)* apply the survey in English and Portuguese versions; and finally, *(f)* analyze the results obtained.

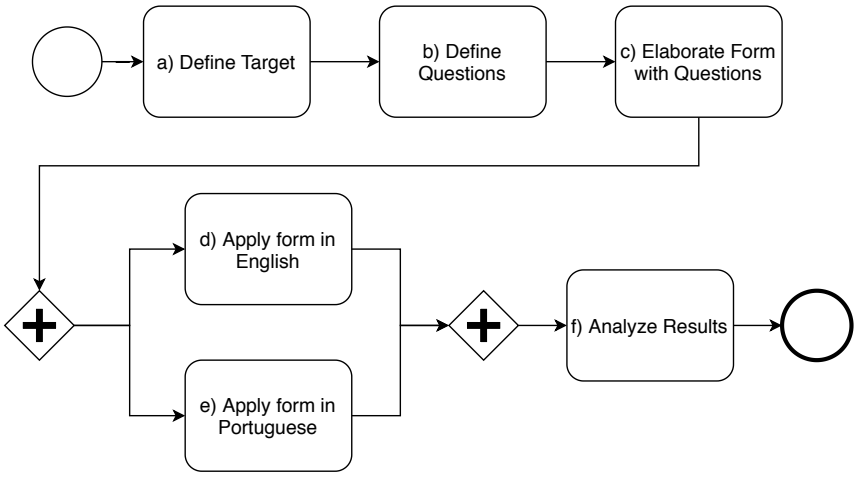


Figure 5.1 – Survey stages.
Source: The author.

5.1 Elaborating the Survey

Aiming to discover the feedback of users, related to the notation, we developed some questions about BPMN. However, before defining the questions, we needed to define the strategy to elaborate the survey. Table 5.1 presents the steps we performed to develop the survey.

To develop the forms, we sought to identify questions that could identify the social context of the participants. The questions focused on identifying the reality where BPMN is applied, considering the context of process analysts and developers.

Table 5.1 – Steps to develop the Survey.

Step	Action
Plan what will be measured	Evidence the survey objectives. Define the research subject in the survey. Obtain additional information on the research subject from source. Determine what will be asked about the research subject.
Develop the survey	For each subject, determine the content of each question. Define the format of each question. Determine how the questions will be written. Assess each question in terms of its ease of understanding, required knowledge and skills need for the participants.
Sequencing and Appearance Decisions	Arrange the questions in an appropriate order. Group all questions from each subtopic to get the best out of them.
Pre-Test and Correction Problems	Read the entire survey to verify if makes sense. Check for possible issues in the survey and fix them.

Author: Adapted from Henkel (2017).

Considering the context of process analysts allows us to identify how the notation is accepted by users. Besides, the survey allows identifying how limited the notation is, where it is not clear to the user, among others. In the context of the developers, we aim to identify the complexity of implementing a BPMS with the notation.

From the two perspectives analyzed through the survey, we describe ways to present improvements in the notation to align with the needs raised in the questionnaires.

We developed the survey using the same questions in both English[1] and Por-

[1] **Analyst**: https://forms.gle/EqiJghaHwRFTcXfW7;
Developers: https://forms.gle/JbuLg98xnREYVaYK7

tuguese[2]. We decided to choose these languages to make it possible to reach more participants.

We split the survey into two steps: *(i)* aimed to get to know the profile of the interviewee through general questions; *(ii)* aimed to understand the user perception concerning notation.

5.1.1 BPMN from the Process Analysts' Perspective

In this survey, we analyzed the opinion of process analysts concerning BPMN notation. For this, we considered the difficulties and challenges that users usually face with the use of the notation. Appendix C depict the form used to apply the survey. Considering the questions, we applied the following:

(i) What is the business of the organization? In this question, we aimed to identify the business of the organization from the participant. This question aims to identify the area in which the user is inserted.

(ii) For how long has the organization managed, performed or executed their process? We aimed to identify the time of use of the BPMN notation. This usage time allows identifying the maturity of the process analyst concerning the notation.

(iii) What are the factors that motivated the organization to choose Business Process Management? In this question, we aimed to identify why the organization decided to choose BPMN as a notation for modeling processes.

After the questions to get to know the context of the organization, the survey moved on to the second step, aiming to identify the process analyst's view and considerations about BPMN.

(i) Through BPMN, can the processes of the organization be fully mapped? Are there particularities that BPMN cannot express? We aimed to identify whether BPMN satisfies the needs of the organization and identify what it cannot express in terms of process design. This question is crucial because we can identify the limits of BPMN, what remains for it to go beyond what it currently offers, and what can be improved.

(ii) What are the benefits identified by the organization while using BPMN? In this question, we focused on identifying the benefits of using notation. From identifying these aspects, it is possible to verify the characteristics that a notation must have to bring

[2]**Analyst**: https://forms.gle/ru7vhFZjSWMUsGG77;
Developers: https://forms.gle/h9r28nL4H2uGWCSv9

benefits to the organization. These aspects are considered when developing a notation.

(iii) How difficult is it to use BPMN? Here, we aimed to investigate how difficult the notation is in the user's perspective. The question below allows us to guide where the notation should be more practical for the user to use it.

(iv) What are the limitations identified with the use of the BPMN? This question aimed to identify where the notation is limited and exercise ways of improving it.

(v) Are the notational elements provided by the BPMN sufficient to represent the process models of the organization? An important question, which can show the need to develop the implementation of the missing notational elements.

(vi) Considering the notational elements of the BPMN available in the BPMS adopted by the organization, is there any unused element for process modeling? Is there any particular reason for not considering this element (or these elements)? In this question, we aimed to find out what are the notational elements which are usually not used by the organization. From identifying these elements, we can have initial insides of which elements to focus on for a possible improvement for the user.

(vii) Would you have suggestions for the improvement of the BPMN in terms of expressiveness? If yes, which ones? Finally, this question aimed to identify what the process analyst could suggest for improvements. Such suggestions can give us a way to identify which aspects to focus on in improving the BPMN rating.

5.1.2 BPMN from the Developers' Perspective

In this survey, we analyzed the view of BPMS developers. To build the survey, we considered the same aspects of the survey for process analysts. Appendix C depict the form used to apply the survey. We applied the following:

Part I – *(i) What is the business of the organization?* In this question, we aimed to identify the business of the organization, whether it works with BPMN development only or other niches.

(ii) For how long has the organization adopted business process management? We aimed to identify for how long the organization has adopted BPM.

(iii) For how long has the organization used BPMN? In this question, we aimed to identify for how long the organization has adopted BPMN.

(iv) Which factors motivated the organization to develop a proper BPMS? This question aimed to understand the motivations that led the organization to focus on a proper

BPMS development.

Part II – *(i) During the BPMS development, has the organization considered some BPMN specifications? Which specifications?* In this question, we focused on understanding whether the organization used any document to develop the BPMS. If the answer is "yes", we asked about the document used. This question is essential to understand developers' adherence to BPMN specification.

(ii) What were the defined criteria for choosing the notational elements of the BPMN to be implemented in the BPMS? We defined this issue in order to better understand the motivation in choosing the elements.

(iii) Which were the challenges faced during the development of the notational elements? In this question, we aimed to understand the difficulties in BPMS development. This understanding allows us to plan ways to be able to plan how to facilitate the implementation of elements.

(iv) Did the organization adapt the notational elements during the BPMS development? and (v) *If the answer in the previous question was "Yes", which were the factors considered by the organization in choosing these elements?* In (iv), we asked if the participant had made some adaptation in the notation. If yes, then the participant answered in (v) the reasons for adopting the notation.

(vi) Would you have suggestions for the improvement of the BPMN? Which ones? As the last question, we asked the participant if they have any suggestions to make regarding the BPMN. It was essential to better understand the needs of the participants, so that we could address them in our work.

5.2 Results Obtained

As a technical-methodological procedure to analyze the answers to our survey, we applied content analysis of the answers. Content analysis is an interpretative analysis of texts through the decomposition of discourse and the rational reconstruction of a central idea with logical rules regarding the origin of these messages to create categories (MAYRING, 2004).

In the context of this thesis, we analyzed the meanings of words, phrases, signs, and symbols, and the decoder unconsciously incorporated them into the mind itself to create categories. The semantic and content analyses applied in the answer for the open questions allowed reaching the first stage, analyzing the organization and textual system-

atization.

When we consider the analysis of the answers, we need to consider dichotomous, trichotomous, multiple, and polytomous. These aspects influence the categories describing the nature and the variation associated with the answers in terms of categorical levels.

It is possible to establish an equal number of categories and different answers obtained in the case of an open question in a survey. Also, we need to differentiate qualitatively or mutually exclude each other (VAUS; VAUS, 2013). Besides, as a result, the attributes of a category must internally show homogeneity (category internal consistency) or a categorical system (KRIPPENDORFF, 2018). Qualitative criteria must be accuracy, completeness, and exclusivity (HAYES; KRIPPENDORFF, 2007). Also, we need to consider that we cannot categorize some responses because it expresses aspects dimensioned both physically and mentally at the same time.

5.2.1 Results obtained in the survey applied to the Process Analysts

In this section, we present the results obtained in both version of the survey (Portuguese and English). To avoid having any bias in interpreting the answers obtained, we performed all the analysis and interpretation of the answers in the language in which the answer was given. We obtained 7 Portuguese answers and 3 English answers. Regardless of the number obtained, we are focused on receiving feedback from users who use BPMN.

(i) What is the business of the organization? In our analysis, we realized that we covered different kinds of organizations, as presented in 5.2:

Table 5.2 – Kinds of organizations covered by survey.

Sequence	Answer
Answer 1	Insurance and pension
Answer 2	Process education
Answer 3	Financial
Answer 4	Research, software development with a focus on corporate management
Answer 5	Process Consulting
Answer 6	Government
Answer 7	BPMS vendor
Answer 8	Education
Answer 9	University - research and teaching
Answer 10	Credits

Source: The author.

(ii) For how long has the organization managed, performed or executed their process? Figure 5.2 presents the result obtained. We found that the minimum obtained was 2 years and the maximum is 20 years.

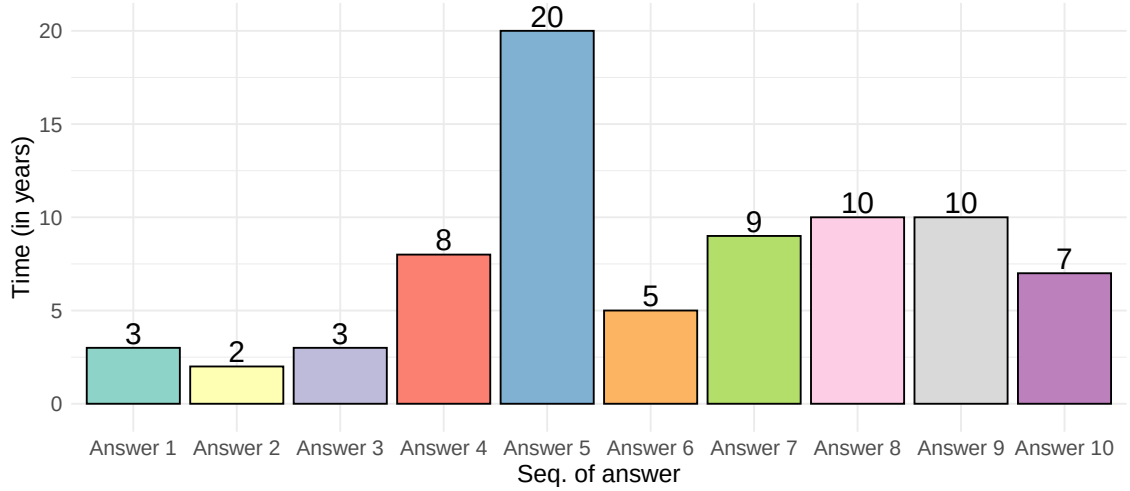


Figure 5.2 – How long the organizations know it processes.
Source: The author.

(iii) Through BPMN, can the processes of the organization be fully mapped? Are there particularities that BPMN cannot express? In this question, Table 5.3 describes the full answers of the participants.

To analyze the responses obtained, we considered the following aspects: whether the participants agreed with the statement of the question, if they disagreed, we analyzed the suggestion given by the participant. Thus, we identified the following aspects: *(i)* 5 agree (Answers 1, 2, 3, 4, and 6); *(ii)* 5 disagree (Answers 5, 7, 8, 9, and 10).

Analyzing the responses from those who disagreed, we can observe that the suggestions raised were: difficulty in implementing the elements and the weak semantics of the elements. With these suggestions, we can have insides to increment the notation. However, the answers given need more details (for example, answer 5 points that the notation does not express everything that the organization needs, but it does not detail what is missing).

(iv) How difficult is it to use BPMN? Figure 5.3 describes the level of difficulty reported by the participants.

We could comprehend that the answers trend is on a medium to a hard level to understand the notation. The results obtained may indicate a need to establish ways to facilitate the use of the notation by users.

(v) What are the limitations identified with the use of BPMN? Table 5.4 depicts the results obtained.

Table 5.3 – Participants answers about BPMN Expressiveness.

Sequence	Answer
Answer 1	Sim, através da notação BPMN os processos podem ser mapeados por completo.
Answer 2	Em geral podem ser mapeados. Em alguns casos, como na utilização de subprocessos eventuais, realizamos uma flexibilização do uso do elemento.
Answer 3	A notação e modelo BPMN é específica para processos previsíveis. Deve-se utilizar CMNN para casos dinâmicos e DMN para decisões com base em regras de negócio.
Answer 4	Não me deparei com problemas pertinentes à BPMN nos processos que trabalhei.
Answer 5	Para a automação dos processos, a notação não expressa tudo o que precisamos.
Answer 6	Os processos poderiam ser mapeados integralmente, mas existem restrições de tempo e custo não permitiram um mapeamento completo na integra.
Answer 7	Como trabalho em uma empresa de Software e no seu leque de produtos consta uma ferramente BPMS para automatização dos processos foi mais fácil a aproximação com a disciplina de gestão. Iniciei em um setor, mas acabei evoluindo internamente na empresa e passei para área de Processos. Principal fator e ganho que eu vi é poder compreender o funcionamento total da empresa, desde uma prospecção de um novo cliente até a entrega do produto. Por isso quando se enxergam dentro do processo se sentem valorizados e importantes. Então, penso que esse seja o ponto principal. Fora também que após conhecer o processo, analisar e conseguir ver oportunidades de melhorias e os retornos tragos para organização são muitos importantes e essenciais na busca da melhoria, que seria a melhoria contínua, sempre podemos melhorar em algo e entregar algo melhor.
Answer 8	Difficult to express: duration of activities, levels of details (it's a limitation of tool, not bpmn)
Answer 9	Maybe some semantic information.
Answer 10	No

Source: The author.

Analyzing the results obtained about the limitations of the BPMN, we could observe that the difficulty in understanding is a limiting factor for notation (answer 2, 6, and 10). Also, participants identified the repeated elements, with extra functions (answer 1), unused elements (answer 4), as well as difficulties in transforming a business model into an executable model (answer 5) or integrating with other standards (answer 3 and 8).

(vi) Are the notational elements provided by the BPMN sufficient to represent the process models of the organization? In this question, we noted that all responses state that the number of current elements is sufficient to express the process of the organization.

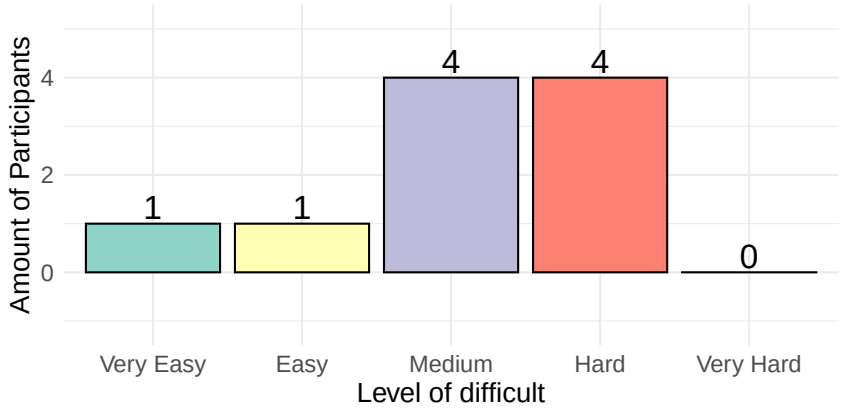


Figure 5.3 – BPMN difficulty levels, according to participants
Source: The author.

Table 5.4 – Participants answers about BPMN limitation.

Sequence	Answer
Answer 1	Tem alguns elementos que por trás tem a mesma ideia de função principal, mas alguns deles tem algumas funções extras e essas sim, para quem não conhece fica difícil interpretar.
Answer 2	Vejo muitas pessoas usarem a notação BPMN como se fosse fluxograma, ou sem conhecer suas regras gramaticais. Isto pode levar a interpretações equivocadas.
Answer 3	Falta melhorar a integração e atualizar os elementos de acordo com os outros standards de BPMN (CMMN e DMN).
Answer 4	Alguns colaboradores das organizações que desconhecem a BPMN tem dificuldades para compreender e ler modelos. Além disso, há muitos elementos na BPMN que são pouco utilizados e por vezes esquecidos para o que servem.
Answer 5	Para a automação do processo mostra-se limitada.
Answer 6	Os especialistas de cada área precisam entender os benefícios da gestão por processos para se engajarem no objetivo de mapear os processos.
Answer 7	Mapear 'coisas' externas ao padrão/processo.
Answer 8	Can't attach document to information flow, can't link process models - subprocesses to expanded processes in different files.
Answer 9	Some basic knowledge in BP is needed.
Answer 10	Complex processes are Hard to model. Versioning is great pain With keeping compatible

Source: The author.

(vii) Considering the notational elements of the BPMN available in the BPMS adopted by the organization, is there any unused element for process modeling? Is there any particular reason for not considering this element (or these elements)? In this question, one could observe several answers as presented in Table 5.5. While 5 participants reported that they did not consider not using any element (answers 1, 2, 5, 7, and 10), the others cited at least one unused element (complex gateway, event-based gateway, inter-

mediate event). We can identify that these cited elements are the most advanced of the notation and, there is an easier equivalent version. For example, as mentioned in answer 3, the participant, instead of using the complex gateway, prefers to use the gateways such as OR and XOR, for example.

Table 5.5 – Participants' answers about unused elements.

Sequence	Answer
Answer 1	Não.
Answer 2	Não, todos são utilizados na organização. Como a BPMS utilizado não tem todo o leque de elementos, se torna mais fácil a utilização de todos eles. Em contrapartida existe alguns novos tipos de elementos que faltam na BPMS utilizada, como por exemplo ligação entre duas piscinas.
Answer 3	Os elementos relacionados a modelagem de transações (subprocesso transacional, eventos de compensação e cancelamento) não utilizamos pois são aplicáveis apenas na modelagem de processos em nível sistêmico muito particulares. Gateway complexo é raramente usado. Gateways de início por evento são evitados.
Answer 4	Conector de página. Este elemento é um "goto" no diagrama que quebra o fluxo contínuo.
Answer 5	Não implementamos na nossa ferramenta todos os elementos. Na verdade, implementamos alguns na medida que são demandados pelos clientes
Answer 6	A maior dos processos pode ser representado com um sub conjunto da bpmn
Answer 7	Não.
Answer 8	We're not using all event types, such as signal.
Answer 9	Maybe complex gateways or so many different events.
Answer 10	No.

Source: The author.

(vii) Would you have suggestions for the improvement of the BPMN in terms of expressiveness? If yes, which ones? Table 5.13 shows that 4 participants did not make suggestions (answers 1, 3, 8, and 10). Considering the participants who made suggestions, two mentioned reducing notational elements (answers 4 and 6). The remaining participants focused on simple suggestions, such as increasing notation details, focusing on the automation of elements (answer 7), or making improvements in sub-processes (answer 5), or, finally, improving the definition of specific elements (answer 2). We can consider this question as very important to obtain strategic aspects for the improvement of the BPMN. We realized that the primary focus of our work should be to improve the existing elements and avoid adding new ones.

Table 5.6 – Participants suggestions about BPMN.

Sequence	Answer
Answer 1	Não.
Answer 2	Mudar a chamada de atividade (call activity) com borda dupla pela tarefa de chamada de processos da CMMN utilizando o ícone de processos vazio para para chamada síncrona e o ícone de processo para chamada assíncrona; Trocar a tarefa de regra de negócios (Business Rule Task) por tarefa de decisão conforme a CMMN; Eliminar a tarefa manual e utilizar o mesmo conceito da CMMN para tarefa de usuário com e sem bloqueio.
Answer 3	Nenhuma.
Answer 4	Pode haver alguns elementos sobrando no leque da notação e até outros que possam dificultar a interpretação, fazendo que as pessoas não entendam perfeitamente o que cada elemento faz de fato. Mas é um detalhe não impactante e que pode ser ou não melhorado.
Answer 5	Permitir subprocesso reusável em subprocessos eventuais, permitindo usar swimlanes para representar papeis e responsabilidades no fluxo eventual. Criar um tipo de tarefa específico para envio de e-mails. Criar um tipo de tarefa específico para usuário robô.
Answer 6	Reduzir a quantidade de elementos notacionais e manter apenas os mais utilizados. Fazer um estudo quantitativo do uso dos elementos notacionais em uma base quantidade suficiente para ser estatísticamente relevante para realizar a retirada dos elementos menos utilizados
Answer 7	Minha sugestão seria para evoluir nos aspectos de especificação de regras de negócio e interfaces que deveriam estar associadas a algumas atividades do modelo. Isso ajudaria no processo de automação.
Answer 8	Can't think of any.
Answer 9	Some attached semantic information (also url or remote data definition, linking with some ontologies or linked data).
Answer 10	No.

Source: The author.

5.2.2 Results obtained in the survey applied to the Developers

In this subsection, we analyze the results obtained in the survey applied to BPMS developers. As in the survey applied to analysts, we applied it in two versions (Portuguese and English). We will present the original answers in both languages, showing the answers first in Portuguese and then in English. The form of analysis of the answers consisted of analyzing the textual structure and identifying the participant's meaning in the answer. We obtained 4 Portuguese answers and 1 English answer.

(i) What is the business of the organization? In this question, we observed that mainly the answers came from participants that are part of organizations focused on developing strategic solutions.

Table 5.7 – Business of the organization of participants.

Sequence	Answer
Answer 1	Desenvolvimento de solução BPMS.
Answer 2	Desenvolver soluções estratégicas para as empresas em forma de software.
Answer 3	Gestão de Empresa.
Answer 4	O foco de negócio da organização se baseia no desenvolvimento de soluções inovadoras para a gestão corporativa.
Answer 5	University.

Source: The author.

About the questions *(ii) For how long has the organization adopted business process management? (iii) For how long has the organization used BPMN? and (iv) Which factors motivated the organization to develop a proper BPMS?*, Table 5.8 depicts the result for these questions. We observed that the target audience of this survey is users who have strong knowledge of BPMN notation, considering the time using BPMN. These participants can add their knowledge in improving the notation.

Table 5.8 – Information about the participants.

Sequence	Time adopting BPM	Time using BPMN	Motivation
Answer 1	15 years	14 years	Necessidade de mercado.
Answer 2	7 years	7 years	Organização, gerenciamento e automatização de processos visando a otimização de processos anteriormente realizados de forma física.
Answer 3	Full time	Desde o principio	Ferramenta do trabalho feita em cima do BPMN, formulários e processos do trabalho todo feito em volta da utilização de BPMN.
Answer 4	6 years	6 years	A gestão de processos é um fator principal para a maioria das organizações, e por isso é essencial o desenvolvimento de uma ferramenta para gerenciar esses processos.
Answer 5	10 years	9 years	Computer science teaching.

Source: The author.

(v) During the BPMS development, has the organization considered some BPMN specifications? Which specifications? Table 5.9 describes that the developers focused on the development of the notation using the official documents. Only participant 5 did not answer that question.

(vi) What were the defined criteria for choosing the notational elements of the

Table 5.9 – Specification considered in BPMS development.

Sequence	Answer
Answer 1	Sim, especificação oficial BPMN da OMG.
Answer 2	Sim. A especificação provida pela BPMN 2.0 relacionada aos elementos do processo. BPMN e XPDL relacionadas a importação e exportação de processos entre BPMS distintos.
Answer 3	Sim, no momento não possuo o links das documentações
Answer 4	Sim, foi utilizado a especificação global BPM CBOK
Answer 5	-

Source: The author.

BPMN to be implemented in the BPMS? According to Table 5.10, we can highlight that the BPMS focus of development is on implementing only the elements necessary for the business processes modeling. Also, the addition of new elements was performed according to customer demand. In this regard, the idea further reinforced is that when it comes to BPMS development, it is not necessary to focus on the development of all notational elements, but only those necessary for modeling processes.

Table 5.10 – Criteria to choose the elements.

Sequence	Answer
Answer 1	Nosso foco é na simplicidade da solução e no usuário de negócio, portanto tinhamos que limitar a, no máximo, 15 elementos. Começamos pelos mais simples e fomos acrescentando com o tempo conforme a necessidade de clientes.
Answer 2	Elementos que pudessem auxiliar na automatização de atividades e que na prática fossem mais utilizados, como atividade de usuário.
Answer 3	Não faço ideia.
Answer 4	Os elementos foram sendo adicionados assim que foram sendo solicitados. Primeiro foram implementados alguns elementos básicos, como atividades de usuário, gateways básicos, para depois ir especificando melhor os elementos.
Answer 5	Use current standards and notation that is used in companies.

Source: The author.

(vii) Which were the challenges faced during the development of notational elements? Table 5.11 shows the obtained results. In this regard, we realized that the most significant challenge of the organization to implement the BPMN is to adapt the elements to execute in BPMS. So, it would be necessary to understand why it leads to adaptation if it is only a question of having a differential compared to other companies or difficulties in understanding the BPMN.

About questions *(viii) Did the organization adapt the notational elements during the BPMS development? and (ix) If the answer in the previous question was "Yes",*

Table 5.11 – Challenges on developing a BPMS.

Sequence	Answer
Answer 1	Nenhum em específico. Alguns elementos da notação passam a ideia errada ao usuário de negócio que desconhece a notação e isso foi um desafio considerando que nosso publico não precisa ser formado em BPMN para operar a ferramenta.
Answer 2	Entender corretamento a sua funcionalidade dentro da notação BPMN 2.0. Qual o papel do elemento dentro da notação e a sua funcionalidade.
Answer 3	Adaptar os elementos
Answer 4	Os desafios na implementação dos elementos foram justamente adaptá-los para o software BPMS, pois tiveram que ser alterados para se ajustar à nossa arquitetura
Answer 5	The initial training.

Source: The author.

which were the factors considered by the organization for choosing these elements?, all participants answered that it is necessary to adapt the BPMN to implement the elements. Considering the factors for this action, Table 5.12 shows that one of the primary needs to adapt the elements is to adapt the notation to the standard of the organization, instead of following it precisely as defined in the specification (answers 2, 3, and 4). One of the answers mentions confusion by users due to how the elements were represented (answer 1). Participant 5 did not answer this question.

Table 5.12 – Factors to adapt the elements.

Sequence	Answer
Answer 1	O evento de troca de mensagem nas primeiras versões para nós era envio de e-mail que tem mais relação com o icone que representa e nossos clientes entendiam melhor. Posteriormente criamos um novo elemento, de "marco", a pedido de nossos clientes.
Answer 2	Adequar os elementos as funcionalidades próprias do software desenvolvido pela empresa.
Answer 3	Adaptar as necessidades da organização.
Answer 4	Foram adaptados pois os elementos não se adaptavam a arquitetura do nosso software.
Answer 5	-

Source: The author.

(x) Would you have suggestions for the improvement of the BPMN? Which ones? In the last question, we aimed to get suggestions given by users. Table 5.13 describes that the participants reinforce the idea that it is necessary to simplify the notation, focusing on improving the semantics of the elements (answer 5), not implementing elements that are not yet included in the BPMS, and making their presentation more straightforward.

Table 5.13 – Suggestions for increase BPMN.

Sequence	Answer
Answer 1	Estamos cansado do BPMN, a especificação tem 14 anos e não evoluiu. A promessa dela não se cumpriu, de ser um elo entre negócio e TI. Ela é muito complexa e a ultima revisão só complicou mais. Para que a gente continue adotando ela no futuro e não troque por outra mais simples, só se oficialmente fosse criada uma versão mais simples. Isso existe pela noção de subset BPMN mas não é oficial. Teria que ter uma versão onde os icones fosse mais genericos.
Answer 2	Simplificar a notação, removendo elementos pouco usual ou alterando as sua funcionalidade, pensando na sua utilização prática e não somente como um elemento gráfico. Existem elementos cuja funcionalidade pode ser executada de outras maneiras. Em suma, diminuir a complexidade de notação em relação a quantidade de elementos.
Answer 3	Exemplo de elementos que poderiam ser removidos/alterados: gateway baseado em eventos paralelos, gateway complexo, evento múltiplo e subprocessos ad hoc.
Answer 4	Não
Answer 5	Add some semantic information (remote data definition as linked data or ontologies)

Source: The author.

5.3 Final Comments

The application of the surveys was essential for the development of this thesis. It was possible to identify a viable path to be taken in the proposal to elaborate pseudo-algorithms. We realized the main focus is on improving the already implemented elements instead of including BPMN elements that have not yet been implemented. Considering the analysis performed in chapter 4, we focus on the implemented elements and make suggestions for improving their implementation with the surveys.

As positive aspects of the application of the questionnaire, we can mention that the target audience rating was covered, that is, people who know the notation and can contribute with their knowledge in improving the notation.

As a limitation of the surveys, we can identify that few users were participating. Even though it was broadcasted on several possible channels, such as e-mail, Twitter, LinkedIn, finding people with knowledge in the area was challenging.

6 PROPOSING PSEUDO-ALGORITHMS FOR BPMN 2.0 ELEMENTS IN BPMS

This chapter describes the steps followed in this thesis to develop pseudo-algorithms based on the definition of the notational elements of BPMN. A limitation of the BPMN specification is the fact that the notation is not fully adhering to the notational elements rules (STROPPI; CHIOTTI; VILLARREAL, 2011; SANTOS et al., 2019). We also identified in chapter 3 the most used elements by the BPMS; in chapter 4 we identified the elements focused on process models; in chapter 5 we discussed user preferences regarding them. In this chapter, we present an approach to increase coding of the BPMN elements, focusing on the elements identified in previous chapters (e.g., OR, sequence flow, task, among others).

In Santos et al. (2019) we used a technique to enable the development of pseudo-algorithms behavioral logic without human intervention. From the textual definition of a notational element, we identify passages that present some behavioral logic, such as deviations. This obtained logic is organized in an algorithm, forming pseudo-algorithms. The technique used was Business Rule. A business rule, according to Karakostas (1990), is a declaration that defines some behavior of the information.

We extracted from the textual definition of the notational XOR element, obtained in BPMN specification, one example of business rule "If and only if none of the conditions evaluates to true, the token is passed on the default Sequence Flow". We considered it as a business rule because it provides an action that should be performed with the *token* when there is no condition evaluated as true in a gateway XOR.

According to Karakostas (1990), when coding an information system, developers use these statements to write the source code for that system. Figure 6.1 presents an example of implementation for the XOR textual definition section.

```
if conditions_true = 0 then
    default_sequence_flow ⟵ token;
```

Figure 6.1 – Example of implementation.
Source: Authors.

In this example, *conditions_true* == *0* verifies that none of the conditions has been evaluated as true. *default_sequence_flow* ← *token* implements the action of passing the token to the default sequence flow. Based on meta-algorithms and business rule, we apply these concepts in this thesis, considering the following steps:

1. Definition of the group of BPMN elements to be studied;
2. Use of an approach to obtain a code fragment, given the definition of a notational element;
3. Generation of pseudo-algorithms based on the obtained fragments and;
4. Verification of the generated pseudo-algorithms.

Through the pseudo-algorithms obtained in Step 3, we intend to add existing features to the elements regarding implementation, extending their purpose. As a group of elements to be studied, we considered the elements focused by BPMS vendors and the frequency of user (see chapter 3 and 4). From this group, we analyzed possible implementation, based on each element definition, in the BPMN specification. With the definition identified, we developed the pseudo code, creating a logical structure based on that textual definition. Finally, we performed verification, identifying whether the pseudo-algorithms correspond to the textual definition.

6.1 Pseudo-algorithms for BPMN elements

The first step of the increment of the notation was to define the set of elements for developing pseudo-algorithms. As a selection criterion, we have chosen one element of each set of elements (considering: task, gateway, sequence flow, events). In this group, we looked for elements that have an implementation in all BPMS analyzed, and these elements are present in the analyzed models in chapter 4.

For the generation of the pseudo-algorithm, we analyzed the textual definition of each element. Afterward, we identified and implemented the business rules, such as deviation identified in specification. We performed the implementation manually. The identification of business rules is a complex task, and documents such as specifications are intended for human beings, typically described in natural languages, such as Portuguese or English (NASCIMENTO, 2014).

We generated pseudo-algorithm with the extracted code. We defined symbols to represent the idea of items that repeatedly appear throughout the pseudo-algorithms to make the logic of the pseudo-algorithms more concise. The abbreviation was taken to create the symbols, for example, condition (c) and exception (e). Special symbols representing an action or state were also considered, such as attribution ($\leftarrow$) and emptiness ($\emptyset$). Table 6.1 illustrates the symbols used in the pseudo-algorithms.

Table 6.1 – Symbols used to develop the pseudo-algorithms

Item	*symbol*
token	tk
input	i
output	o
process	P
sequence flow	sf
default sequence flow	$sf_{default}$
message flow	fm
condition	c
true	T
set of conditions	C
assignment	$\leftarrow$
empty	$\emptyset$
exception	e
equals	$=$
not equal to	$\neq$
greater than	$>$
subset	$\in$

Source: The author.

Pseudo-algorithm 1: *Inclusive Gateway (OR) + Start Process Instance*. We propose to extend the OR functionality with the possibility of starting a process instance.

Definition: "Each token arriving at any incoming Sequence Flows activates the gateway and is routed to exactly one of the outgoing Sequence Flows. In order to determine the outgoing Sequence Flows that receives the token, the conditions are evaluated in order. The first condition that evaluates to true determines the Sequence Flow the token is sent to. No more conditions are henceforth evaluated. If and only if none of the conditions evaluates to true, the token is passed on the default Sequence Flow.

In case all conditions evaluate to false and a default flow has not been specified, an exception is thrown.

Each occurrence of one of these gateway will lead to the creation of a new Process instance, if the flag indicating this action is true.". The pseudo-algorithm obtained is as it follows (see Algorithm 1).

Pseudo-algorithm 2: *Sequence Flow + default path*. We propose to enable the sequence flow to have a default path, in addition to checking the conditions. *Definition:* "A Sequence Flow is used to show the order of Flow Elements in a Process or a Choreography. Each Sequence Flow has only one source and only one target. The source and target must be from the set of the following Flow Elements: Events (Start, Intermedi-

Algorithm 1: Pseudo-algorithm proposed for OR

```
initialization;
for condition in condition.list() do
    if condition = T then
        sf ← tk;
        hasTrue ← T
    end
end
if hasTrue ≠ T then
    throw new exception ("input token is missing");
end
if startProcess = T then
    output ← Proces Instance;
end
```

ate, and End), Activities (Task and Sub-Process; for Processes), Choreography Activities (Choreography Task and Sub-Choreography; for Choreographies), and Gateways. A Sequence Flow can optionally define a condition Expression, indicating that the token will be passed down the Sequence Flow only if the Expression evaluates to true. This Expression is typically used when the source of the Sequence Flow is a Gateway or an Activity.

If the sequence flow has a marker as default, then the token takes the flow". The pseudo-algorithm obtained is as it follows (see Algorithm 2).

Algorithm 2: Pseudo-algorithm proposed for Sequence Flow

```
initialization;
if source = gateway or source = task then
    if C ≠ ∅ then
        verify conditions
        if C  T and  sf_default then
            sf ← sf_default;
        end
    else
        throw new exception ("there is not conditions");
    end
end
```

Pseudo-algorithm 3: *Receive Task + Start Process Instance*. We propose to enabling the element to start a process instance, besides to receive a message. *Definition:* "Upon activation, the Receive task begins waiting for the associate message. When the message arrives, the data in the Data output of the Receive task is assigned from the data in the Message, and Receive Tasks is completed if the instantiate attribute of the Receive

Task is set to true, the Receive Tasks itself can start a new process Instance". The pseudo-algorithm obtained is as it follows (see Algorithm 3).

Algorithm 3: Pseudo-algorithm for proposed Receive Task

```
1 initialization;
2 waitingm;
3 receivem;
4 if instantiate = T then
5 |   start P
6 end
```

6.2 Verification of the Pseudo-algorithms

With the development of meta-algorithms, it is necessary to determine that the work developed is really in conformity with the definitions of the BPMN elements in the BPMN specification. For this, the present work makes use of *software tests* to achieve these objectives.

The software test is a broader element, referred to as verification and validation. *Verification* deals with activities that ensure that software correctly implements the requirements raised. The *validation*, on the other hand, deals with a set of activities that ensure that the software built corresponds to the client's wishes (PRESSMAN, 2006).

Any description of the behavior of a developed system can apply functional tests, from an informal description to an official specification and at any level of granularity. A functional specification is the most important source of information for testing. Deriving test cases from a specification is called a functional test (PEZZÈ; YOUNG, 2008; PATTON, 2006).

In this thesis, the specification of the notational elements of the BPMN can be considered a functional specification. From this, we extracted the behavior for the element. Thus, we performed the verification using *Decision Tables* and *Control Flow Graphs* (CFG), considering the requirements of the notational elements as requirements (PEZZÈ; YOUNG, 2008).

Decision Tables are tables that represent the inputs versus test conditions. The decision table helps to check all possible combinations of conditions for testing, and testers can quickly identify missed conditions. We can indicate the conditions as True (T) and False (F) values (PEZZÈ; YOUNG, 2008).

For example, consider a specification as a construction of a decision table, which determines the value that a variable (named as y) must assume, based on the variable x:

"*According to the value of x, calculate the value of y:*

*- **If** x corresponds to 1, **then** y should receive twice x, **otherwise it should be tested** if x corresponds to 2. **If** true, then y should receive 2 in the power x. **If false, then check** if x corresponds to 3. In this situation, y will receive 2 divided by x.*"

The first step to build a decision table is to list the conditions available in the specification. As conditions, we considered sentences that deal with any input, processing, output, or decision criteria (PEZZÈ; YOUNG, 2008). In the example, the sections dealing with some processing (e.g., structure *if*, *then*) were defined as conditions. In the example specification, the highlighted sentences fit this criterion:

- Is $x = 1$? - Possible values: *True* (T) or *False* (F).
- Is $x = 2$? - Possible values: T or F.
- Is $x = 3$? - Possible values: T or F.

In the example, considering the specification, it is identified if x can correspond to three different values (1, 2, and 3). These identified rules configure the conditions that the decision table uses.

The second step is to calculate the combinations of the possible values. The possible values are Boolean (T or F), then the combinations are given by $2^{\text{number of conditions}}$. In the example, 3 conditions were identified, then $2^3 = 8$ possible combinations.

In the third step, we created the decision table with the combination of values and the respective actions of the combinations. Actions are identified based on combinations of conditions. Figure 6.2 shows the initial decision table.

In the condition area, we presented the possible combinations for the three conditions. Each column has a combination of conditions, with *T* representing True and *F*, False. The shared area has the set of shares raised, with the mark "X" corresponding to the share for that particular combination of conditions. Considering the second column of combinations, reading a combination from the table can be done as follows: "When x corresponds to 1 is false, x corresponds to 2 is false, and x corresponds to 3 is true, so the action y divided by 2 can occur."

The columns highlighted in Figure 6.2 (see 1, 2 and 3) indicate columns in the table that can be simplified. For example, in 1, the columns may have an identical value, except for one value (indicated with the ellipse). Regardless of the value assumed in this

Figure 6.2 – Initial Decision Table

		Possible Combinations							
Conditions	*x* corresponds to 1	F	F	F	F	T	T	T	T
	x corresponds to 2	F	F	T	T	F	F	T	T
	x corresponds to 3	F	T	F	T	F	T	F	T
Actions	*y* twice *x*					X	X	X	X
	y ^ *x*			X	X				
	y is 2 / *x*		X						
				1		2		3	

Source: The author.

condition, it does not affect the result action. Next, join the columns, and add the *don't care* ("-") in the disparate values. Repeat the same procedure in 2 and 3. When ending the first stage, review the columns, and if necessary, repeat the procedure until there are no more combinations between the columns. Minimization is necessary to eliminate redundant columns. At the end of the simplification, Figure 6.3 depicts the final decision table.

Figure 6.3 – Simplified Decision Table

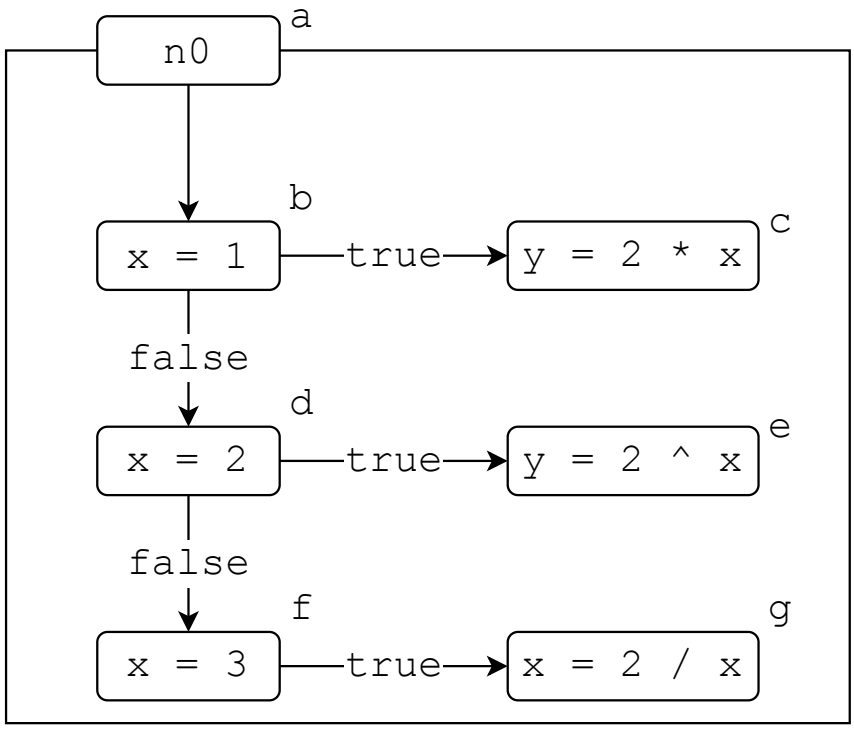
		Possible Combinations			
Conditions	*x* corresponds to 1	F	F	F	T
	x corresponds to 2	F	F	T	-
	x corresponds to 3	F	T	-	-
Actions	*y* twice *x*				X
	y ^ *x*			X	
	y is 2 / *x*		X		

Source: The author.

With a final decision table, we applied a test suite to test an implementation represented in the CFG form. Figure 6.4 represents an example of CFG; The Decision Table of Figure 6.3 represents the specification. Then, it is necessary to go through the CFG

paths with each of the test cases (each column in the *Possible Combinations* area of the Figure 6.3).

Figure 6.4 – Control-Flow Graph example.



Source: The author.

Each test case traversed through the graph and reached a final edge, and we affirm that this case is *verified*. If any test case does not reach a final edge, there is some inconsistency (some non-conformity in the implementation). At the end of the test set, if all cases reach a final edge in the graph, it is possible to affirm, based on this test, that the implemented code *represents* the specification. Based on the set of test cases in the decision table in Figure 6.3, we obtained the following paths:

- Test case t_1: {b = F, d = F, f = F } $\longrightarrow$ [Edges covered p_1: a, b, d, f];
- Test case t_2: {b = F, d = F, f = T } $\longrightarrow$ [Edges covered p_2: a, b, d, f, g];
- Test case t_3: {b = F, d = T, f = - } $\longrightarrow$ [Nodes covered p_3: a, b, d, e];
- Test case t_4: {b = T, d = -, f = - } $\longrightarrow$ [Nodes covered p_4: a, b, c];

Each test case resembles the value (T or F) assumed at a given edge (with a condition). For example, in t_1 a test is presented: $b = F$. The test indicates that the condition present in the node *b* ($x = 1$) will be tested with the value *false* (*F*). The combination of the assumed value in this edge with the others identifies the traveled edges, as depicted in

p_1. In test cases, whose nodes have the value "-" (*don't care*), it is assumed that regardless of the value that it will assume, it will not affect the edges.

In the example of this chapter, all test cases reached the final CFG nodes (*c, e, f, g*). Then, it is possible to conclude that the implementation represents its respective specification based on this test.

Based on the definition of the Decision table and CFG in this thesis, the decision tables represent the textual definitions of the elements, and the CFG represents the pseudo-algorithms.

Pseudo-algorithm 1: *Inclusive Gateway (OR) + Start Process Instance*

Specification: "Each token arriving at any incoming Sequence Flows activates the gateway and is routed to exactly one of the outgoing Sequence Flows. In order to determine the outgoing Sequence Flows that receives the token, the conditions are evaluated in order. **The first condition that evaluates to true** determines the Sequence Flow the token is sent to. No more conditions are henceforth evaluated.

If and only if none of the conditions evaluates to true, the token is passed on the default Sequence Flow.

In case all conditions evaluate to false and a **default flow has not been specified**, an exception is thrown.

Each occurrence of one of these gateway will lead to the **creation of a new Process instance**, **if the flag indicating this action is true**."

The first step is to identify the conditions. The highlighted in the text correspond to the conditions identified:

- *The conditions are evaluated?* - Possible values: *True* (T) or *False* (F).
- *None of the conditions evaluates to true*? - Possible values: T or F.
- *All conditions evaluate to false and default flow has not been specified*? - Possible values: T or F.
- *The flag indicating this action (create a process instance) is true*? - Possible values: T or F.

Based on the conditions identified, the second step was to define the decision table, as presented on Table 6.2

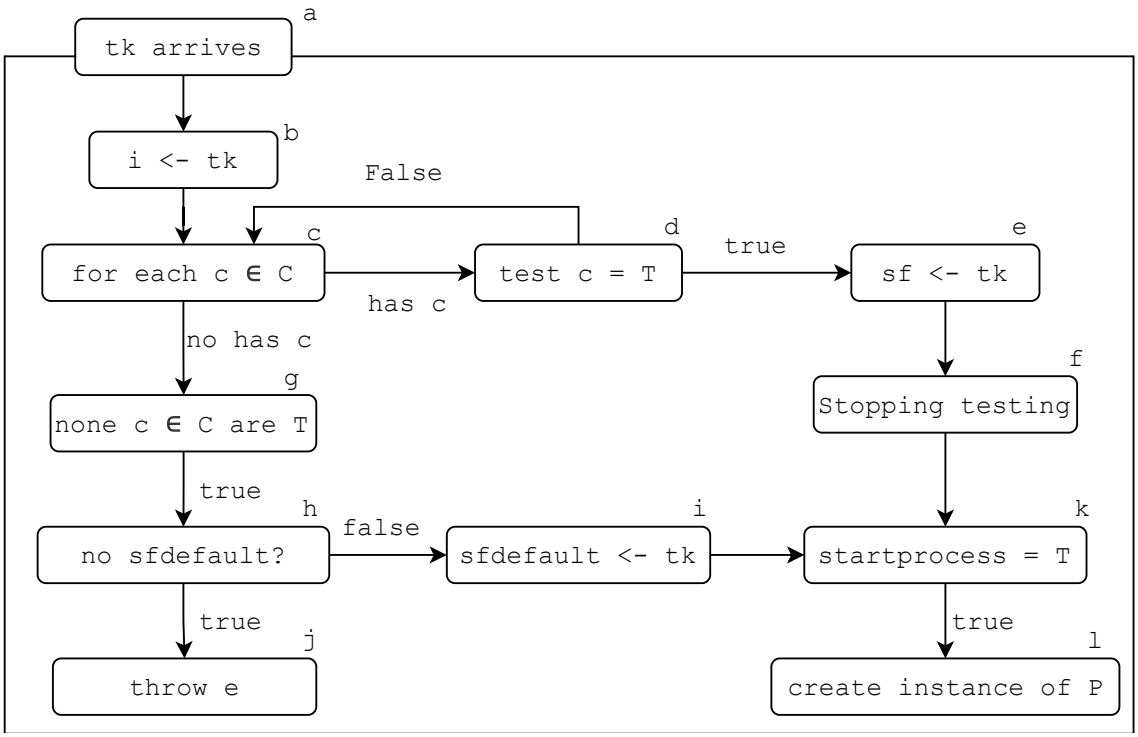
With the Decision Table, we check the suite test, to validate the pseudo-algorithm defined for the element OR. The third step was to elaborate the Control-Flow Graph, presented in Figure 6.5.

Table 6.2 – Decision table for proposed OR element.

		Possible combinations				
Conditions	*The conditions are evaluated?*	F	F	F	T	T
	None of the conditions evaluates to true?	T	T	T	-	-
	All conditions evaluate to false and default flow has not been specified?	F	F	T	-	-
	The flag indicating this action (create a process instance) is true?	T	F	-	F	T
Actions	*The token is sent to sequence flow*				X	X
	The token is passed on the default Sequence Flow	X	X			
	An exception is thrown			X		
	Create a process instance	X				X

Source: The author.

Figure 6.5 – Control-Flow Graph for proposed OR element.



Source: The author.

From the set of test cases in the decision table, the following paths are performed in the CFG:

- Test case t_1: {d=F, g=T, h=F, k=T} $\longrightarrow$ [Visited edges p_1: a, b, c, d, g, h, i, k, l];
- Test case t_2: {d=F, g=T, h=F, k=F} $\longrightarrow$ [Visited edges p_2: a, b, c, d, g, h, i, k];

- Test case t_3: {d=F, g=T, h=T, k=-} $\longrightarrow$ [Visited edges p_3: a, b, c, d, g, h, j];
- Test case t_4: {d=T, g=-, h=-, k=F} $\longrightarrow$ [Visited edges p_4: a, b, c, d, e, f, k];
- Test case t_5: {d=T, g=-, h=-, k=T} $\longrightarrow$ [Visited edges p_5: a, b, c, d, e, f, k, l];

For this test suite, all test cases reached the final edges of the CFG (*j, k, l*). Then, it is possible to conclude that the pseudo-algorithm for the element OR represents the BPMN specification.

- Pseudo-algorithm 2: *Sequence Flow + default path.*

Definition: "A Sequence Flow is used to show the order of Flow Elements in a Process or a Choreography. Each Sequence Flow has only one source and only one target.

The source and target MUST be from the set of the following Flow Elements: Events (Start, Intermediate, and End), Activities (Task and Sub-Process; for Processes), Choreography Activities (Choreography Task and Sub-Choreography; for Choreographies), and Gateways.

A Sequence Flow can optionally **define a condition Expression**, indicating that the token will be passed down the Sequence Flow only if the Expression evaluates to true. This Expression is typically used when the source of **the Sequence Flow is a Gateway or an Activity**.

If the sequence flow has a marker as default, **then the token takes the flow**."

First step is to identify the conditions. The highlighted in the text correspond to the conditions identified from the BPMN specification:

- *Source is gateway or activity?* - possible values: *True* (T) or *False* (F).
- *Is there a expression?* - possible values: T or F.
- *Has a marker as default?* - possible values: T or F.

Based on the conditions identified, the second step was to define the decision table, as presented on Table 6.3.

With the Decision Table, we have the test suite to verify the pseudo-algorithm defined for the sequence flow. With that, the third step was to elaborate the Control Flow Graph, presented in Figure 6.6.

From the set of test cases in the decision table, the following paths are performed in the CFG:

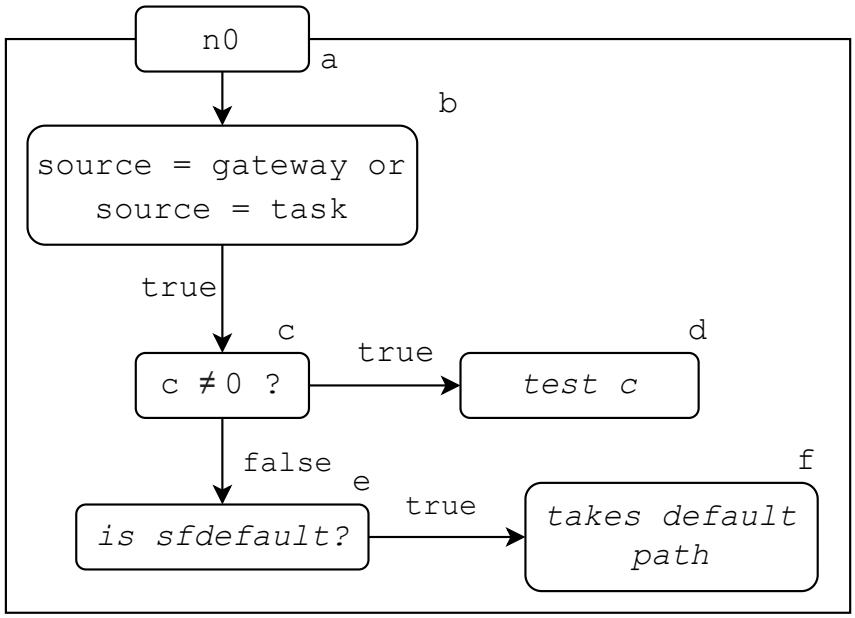
- Test case t_1: {b=F, c=-, e=-} $\longrightarrow$ [Edges visited p_1: a, b];
- Test case t_2: {b=T, c=F, e=F} $\longrightarrow$ [Edges visited p_2: a, b, c, e];

Table 6.3 – Decision Table for Sequence Flow.

		Possible Combinations			
Conditions	*Source is gateway or activity?*	F	T	T	T
	Is there a expression?	-	F	F	T
	Has a marker as default?	-	F	T	-
Actions	*Evaluate the expression*				X
	Token takes the default flow			X	

Source: The author.

Figure 6.6 – Control-Flow for Sequence Flow.



Source: The author.

- Test case t_3: {b=T, c=F, e=T} $\longrightarrow$ [Edges visited p_3: a, b, c, e, f];
- Test case t_4: {b=T, c=T, e=-} $\longrightarrow$ [Edges visited p_4: a, b, c, d];

For this test suite, all test cases reached the final edges of the CFG (*b, d, e, f*). Then, it is possible to affirm that the sequence flow pseudo-algorithm represents the specification.

- Pseudo-algorithm 3: *Receive Task + Start Process Instance*.

Definition: "Upon activation, the Receive Task begins waiting for the associated Message. When the Message arrives, the data in the Data Output of the Receive Task is assigned from the data in the Message, and Receive Task completes. For key-based correlation, only a single receive for a given CorrelationKey can be active, and thus the Message matches at most one Process instance. For predicate-based correlation, the Mes-

sage can be passed to multiple Receive Tasks. **If the Receive Task's** instantiate attribute is set to true, the Receive Task itself can start a new Process instance."

First step is to identify the conditions. The highlighted in the text correspond to the conditions identified:

- *Is the instantiate attribute set to true?* - Possible value: *True* (T) or *False* (F).

Based on the condition identified, the second step was to define the decision table, as presented on Table 6.4.

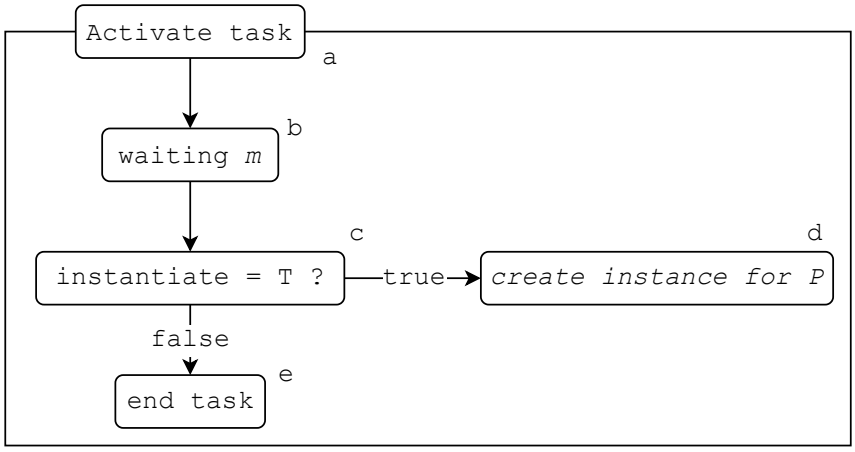
Table 6.4 – Decision Table for Receive Task.

		Possible Combinations	
Condition	*Is the instantiate attribute set to true?*	F	T
Actions	*Start a new Process instance*		X
	Only complete the Receive Task	X	

Source: The author.

With the Decision Table, we have the test suite to verify the pseudo-algorithm defined for the Receive Task. With that, the third step was to elaborate the Control Flow Graph, presented in Figure 6.7.

Figure 6.7 – Control-Flow for receive Task.



Source: The author.

From the set of test cases in the decision table, the following paths are performed in the CFG:

- Test case t_1: {c=F} $\longrightarrow$ [Edges visited p_1: a, b, c, d];
- Test case t_2: {c=T} $\longrightarrow$ [Edges visited p_2: a, b, c, e];

For this test suite, all test cases reached the final edges of the CFG (*d, e*). Then, it is possible to affirm that the pseudo-algorithm of the receiving task element represents the specification.

6.3 Final Comments

In this chapter, we proposed a way to increase the implementation of the elements focused on models, BPMS, and users. We have also applied techniques that could verify and validate the pseudo-algorithm proposed. For that, we applied a verification to demonstrate that the pseudo-algorithms represent the textual definition of the BPMN elements.

With the pseudo-algorithms, we can demonstrate that it is possible to increase the semantics of a process model by increasing the implementation of elements without adding more complexity. It is possible to join the shared resources in the same element.

As limitations of this chapter, we do not develop all pseudo-algorithm possibilities, and considering the verification, we should perform validation with users. However, considering the verification performed (with decision tables and control-flow graphs), we can evidence that the pseudo-algorithms represent the definition of the elements as the BPMN textual specification presents. Then, by performing the steps to extract the pseudo-code for all pseudo-algorithm possibilities and verifying the techniques presented in this thesis, we can obtain the same result as presented in this chapter.

This validation was applied to validate one of the specific objectives of this work, which aimed to define meta-algorithms that adhere to the textual definition available on the BPMN specification.

7 CONCLUSION

In this thesis, we evaluated 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 implementation. With the information on the implementation, we could identify the elements that need to be implemented and explore how to do it.

The analysis of related works identified that the existent contributions are focused on verifying resource aspects (how to handle documents, what reports the BPMS offers, among others) of the BPMS. Our research can help to determine the limits of BPMN implementation, 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 expressiveness of process models.

Our first contribution is an analysis of the BPMS in terms of reality. This analysis allowed us to identify that the BPMS focus is on implementing a set of notational elements instead of all the elements. We can identify that the focus is on implementing few elements but a more adherent way to BPMN. Moreover, the method applied in the analysis (as presented in chapter 3) allows us to replicate the analysis at any time in the future, updating the study with new BPMS.

Our second contribution is in identifying recurrent elements used in a process model. From the 186 diagrams analyzed in our research, we identified the common use of elements such as event-based gateway, send task, receive task in process modeling. Identifying common elements leads to identifying the tendency for different users to use the same elements for process modeling. With the application of the Anova technique, we could verify that the diagrams obtained from different sources use similar elements, reinforcing that regardless of the BPMS used, the elements used tend to be the same.

Our third contribution is to obtain user feedback regarding BPMN. We obtained the feedback through a survey applied to specialists in process modeling and BPMS developers. We found that the focus is not on having a BPMS with many elements implemented but on the amount needed to express the processes of the organization for both participants. The found focus helps us understand that any research performed on BPMN elements must focus on the elements that already have an implementation in the BPMS.

Our fourth contribution is the proposal of pseudo-algorithms to increase the expressiveness of the elements already implemented in the BPMS. In choosing the elements, we considered the criteria defined in the previous chapters: *(i)* elements implemented in

the BPMS, Chapter 3; *(ii)* elements that appear in process models, Chapter 4; *(iii)* based on participant feedback, Chapter 5. Using a technique proposed by Santos et al. (2019), it was possible to develop the pseudo-algorithms and verify them using control-flow graphs and decision tables. The work produced in this step indicates that the users can use the strategy of developing pseudo-algorithms for those who want to extend or develop coding for BPMN elements. It is a technique already used and allows adherence to the definition of the notational element.

As limitations, 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 BPMSs evaluated. We cannot assure if these BPMSs provide support to the elements that have no implementation.

Each of the steps performed on this thesis brings a contribution to the thesis. We can go further and combine all the steps, like a framework for BPMS evaluation, applying in a specific frequency of time (e.g., 2 - 5 years) to measure the use of elements in the BPMS. The analysis steps (BPMS, process models, and survey) would be helpful to measure adherence to the elements in BPMS, and the pseudo-algorithm would help develop a solution for those identified elements to complement these elements or adding new features.

From the development of the pseudo-algorithms, we suggest as future work, the extension of the implemented elements, adding new features that can be useful to process modeling. Related works made efforts in this direction. Stroppi, Chiotti and Villarreal (2011) proposed an extension to BPMN but conceiving new elements. However, the addition of new elements may increase the complexity of the BPMN. Thus, we suggested the extension considering only the elements already implemented in BPMS.

We suggest conducting a study about the complexity of the process models to help to identify aspects of difficulty in understanding the process models. As an example of understanding, we can consider that certain elements should increase or decrease the complexity of the elements. Similar researches are currently underway in this regard. Mendling, Reijers and Aalst (2010) proposed seven guidelines for process modeling, and one talks about the number of elements that a model can have (no more than 50). It helps with complexity, but we can go further and analyze other factors, such as graphical representation, the implementation of the element, among other factors.

REFERENCES

AALST, W. M. P. V. D. et al. Workflow patterns. **Distributed and Parallel Databases**, v. 14, n. 1, p. 5–51, 2003.

AALST, W. V. D. **Process mining: discovery, conformance and enhancement of business processes**. 1. ed. [S.l.]: Springer Science & Business Media, 2011. ISBN 9783642193453.

AREVALO, C. et al. A metamodel to integrate business processes time perspective in BPMN 2.0. In: . [S.l.: s.n.], 2016. v. 77, p. 17–33. ISBN 0950-5849. ISSN 09505849.

BOCCIARELLI, P. et al. Empowering business process simulation through automated model transformations. In: **Proceedings of the Symposium on Theory of Modeling & Simulation - DEVS Integrative**. [S.l.: s.n.], 2014. (DEVS '14), p. 1–9.

BöRGER, E. Approaches to modeling business processes: A critical analysis of bpmn, workflow patterns and yawl. **Software and Systems Modeling**, Springer-Verlag New York, Inc., v. 11, n. 3, p. 305–318, jul. 2012.

BOX, G.; HUNTER, J.; HUNTER, W. **Statistics for Experimenters: Design, Innovation, and Discovery**. [S.l.]: Wiley, 2005. (Wiley Series in Probability and Statistics). ISBN 9780471718130.

CETINKAYA, D.; VERBRAECK, A.; SECK, M. D. 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**. [S.l.: s.n.], 2012. (TMS/DEVS '12), p. 1–6. ISBN 978-1-61839-786-7.

CORRADINI, F. et al. A guidelines framework for understandable bpmn models. **Data & Knowledge Engineering**, v. 113, p. 129–154, 2018. ISSN 0169-023X.

CORTES-CORNAX, M. et al. Using intentional fragments to bridge the gap between organizational and intentional levels. **Information and Software Technology**, v. 58, p. 1–19, 2015. ISSN 0950-5849.

DUMAS, M. et al. **Fundamentals of Business Process Management**. 1. ed. Berlin Heidelberg: Springer, 2013.

DUMAS, M. et al. **Fundamentals of Business Process Management, Second Edition**. [S.l.]: Springer, 2018. ISBN 978-3-662-56508-7.

EID-SABBAGH, R.-H. et al. A Platform for Research on Process Model Collections. In: INT. WORKSHOP ON BUSINESS PROCESS MODEL AND NOTATION. **Proceedings...** Vienna, Austria: Springer, 2012. v. 125, n. 4, p. 8–22.

FANTINATO, M.; GIMENES, I. M. S.; TOLEDO, M. B. F. Product line in the business process management domain. In: **Applied Software Product Line Engineering**. [S.l.]: Auerbach Publications, 2010. chp. 20, p. 497–530.

FØLSTAD, A.; LAW, E.; HORNBÆK, K. Analysis in practical usability evaluation: A survey study. In: **Proceedings of the SIGCHI Conference on Human Factors in Computing Systems**. New York, NY, USA: Association for Computing Machinery, 2012. (CHI '12), p. 2127–2136. ISBN 9781450310154.

FRIEDRICH, F. Automated generation of business process models from natural language input. In: . [S.l.]: Citeseer, 2010.

GASSEN, J. B. et al. 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**. [S.l.: s.n.], 2015. p. 69–80.

GEIGER, M. et al. BPMN 2.0: The state of support and implementation. In: . [S.l.]: Elsevier B.V., 2016. v. 80, p. 250–262. ISSN 0167739X.

GEIGER, M.; WIRTZ, G.; WEBEREI, A. der. Bpmn 2.0 serialization-standard compliance issues and evaluation of modeling tools. In: CITESEER. **International Workshop on Enterprise Modeling and Information Systems Architectures**. [S.l.], 2013. p. 177–190.

HAYES, A. F.; KRIPPENDORFF, K. Answering the call for a standard reliability measure for coding data. **Communication methods and measures**, Taylor & Francis, v. 1, n. 1, p. 77–89, 2007.

HENKEL, K. A categorização e a validação das respostas abertas em surveys políticos. **Opinião**, scielo, v. 23, p. 786 – 808, 12 2017.

ISO. **ISO/IEC 19510:2013: Information technology – Object Management Group Business Process Model and Notation, V2.0.2**. Geneva, 2013. 1–507 p.

KARAGIANNIS, D. BPMS: business process management systems. **CM SIGOIS Bulletin**, v. 16, n. 1, p. 10–13, 1995.

KARAKOSTAS, V. Modelling and maintenance software systems at the teleological level. **Journal of Software Maintenance**, v. 2, n. 1, p. 47–60, 1990.

KLUZA, K. et al. Opportunities for business process semantization in open-source process execution environments. **2015 Federated Conference on Computer Science and Information Systems (FedCSIS)**, p. 1307–1314, Sept 2015.

KRIPPENDORFF, K. **Content analysis: An introduction to its methodology**. [S.l.]: Sage publications, 2018.

La Rosa, M. et al. Managing Process Model Complexity via Concrete Syntax Modifications. **IEEE Transactions on Industrial Informatics**, v. 7, n. 2, p. 255–265, 2011.

LAZARTE, I. M. et al. An mda-based method for designing integration process models in B2B collaborations. In: **ICEIS 2011 - Proceedings of the 13th International Conference on Enterprise Information Systems, Volume 3, Beijing, China, 8-11 June, 2011**. [S.l.: s.n.], 2011. p. 55–65.

MAYRING, P. Qualitative content analysis. **A companion to qualitative research**, v. 1, n. 2004, p. 159–176, 2004.

MEIDAN, A. et al. A survey on business processes management suites. **Computer Standards & Interfaces**, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 51, n. C, p. 71–86, mar. 2017.

MENDLING, J.; REIJERS, H. A.; AALST, W. M. P. van der. Seven process modeling guidelines (7PMG). **Information and Software Technology**, v. 52, n. 2, p. 127–136, 2010.

MONTGOMERY, D. C. **Introduction to statistical quality control**. [S.l.]: John Wiley & Sons, 2007.

NASCIMENTO, G. S. et al. Discovering business processes in legacy systems using business rules and log mining. In: **Proc. of the IEEE 10th Int. Conf. on e-Business Engineering (ICEBE)**. New Jersey, USA: IEEE Computer Society, 2013. p. 207–212.

NASCIMENTO, G. S. d. **Um Método para Descoberta Semi-Automática de Processos de Negócio Codificados em Sistemas Legados**. 109 p. Thesis (PhD) — Universidade Federal do Rio Grande do Sul, 2014.

OMG, O. M. G. **Business Process Model and Notation (BPMN) Version 2.0**. [S.l.], 2013.

PATTON, R. **Software testing**. 2. ed. Indianopolis: Sams Pub., 2006. ISBN 0672327988.

PERALTA, M. et al. Workflow models: Management and quality of process in the cloud. In: **International Conference on Information Technology: New Generations**. [S.l.: s.n.], 2014. p. 91–96.

PEZZÈ, M.; YOUNG, M. **Teste e Análise de Software: Processos, Princípios e Técnicas**. Porto Alegre: Bookman, 2008. ISBN 9788577802623.

PRESSMAN, R. **Engenharia de Software**. 6. ed. São Paulo: McGraw Hill Brasil, 2006. ISBN 8586804576.

PRIEGO-ROCHE, L. M. et al. Business process design from virtual organization intentional models. In: INT. CONF. ON ADVANCED INFORMATION SYSTEMS ENGINEERING. **Proceedings...** Berlin, Heidelberg: Springer-Verlag, 2012. (CAiSE'12), p. 549–564.

SANTOS, C. F. H.; THOM, L. H.; FANTINATO, M. Investigating completeness of coding in business process model and notation. In: **Proc. of the 17th Int. Conf. on Enterprise Information Systems (ICEIS)**. Lisboa, Portugal: SciTePress, 2015. v. 3, p. 328–333.

SANTOS, C. F. H. D. et al. Supporting bpmn tool developers through meta-algorithms. **International Journal of Business Information Systems**, v. 32, n. 4, p. 460–488, 2019.

SILVA, L. C. et al. Selection of a business process management system: An analysis based on a multicriteria problem. In: **International Conference on Systems, Man, and Cybernetics (SMC)**. [S.l.: s.n.], 2014. p. 295–299.

STEMBERGER, M. I. et al. Propositions on the interaction of organizational culture with other factors in the context of BPM adoption. **Business Proc. Manag. Journal**, v. 24, n. 2, p. 425–445, 2018.

STROPPI, L. J. R.; CHIOTTI, O.; VILLARREAL, P. D. Extending BPMN 2.0: (method) and tool support. In: **Proc. of the 3rd Int. Workshop on Business Process Model and Notation (BPMN)**. Berlin Heidelberg, Germany: Springer, 2011. (LNBIP, v. 95), p. 59–73.

THOM, L. H.; REICHERT, M.; IOCHPE, C. On the support of workflow activity patterns in process modeling tools: Purpose and requirements. In: **Proc. of the 3rd Workshop on Business Process Management (WBPM)**. Porto Alegre, Brazil: Brazilian Computing Society, 2009. p. 1–6.

VAUS, D. D.; VAUS, D. de. **Surveys in social research**. [S.l.]: Routledge, 2013.

WESKE, M. **Business Process Management**. 2. ed. Berlin Heidelberg: Springer, 2014.

WFMC, W. M. C. W. S. **WFMC-TC-1011: Workflow Management Coalition: Terminology & Glossary**. [S.l.], 1999. 1–65 p.

WORLD WIDE WEB CONSORTIUM. **W3C XML Schema Definition Language (XSD) 1.1**. [S.l.], 2012.

APPENDIX A — RESUMO ESTENDIDO

Gerenciamento de Processos de Negócio BPM - Business Process Management - é uma disciplina que visa a documentação e a padronização dos processos de negócio, sendo que um processo de negócio pode ser representado graficamente, através da Notação e Modelo de Processo de Negócio (BPMN - *Business Process Model and Notation*). A partir dos modelos representados graficamente, as organizações podem demandar a necessidade de automação desses processos.

A automação requer o uso de tecnologia da informação para que modelos de processos possam ser executados através do Sistema de Gerenciamento de Processos de Negócio (BPMS - *Business Process Management System*).

No contexto da automação de processos, tem-se com problema, diferentes BPMS que implementam diferentes elementos do BPMN. O desafio de transformar um modelo de processo em uma versão executável é o pouco detalhamento da implementação dos elementos, na especificação BPMN. Como resultado, os desenvolvedores de BPMS definem seu formato proprietário para mapear e transformar um modelo de processo em um modelo executável. É possível ter-se diferentes implementações entre os elementos da notação, o que pode causar falta de interoperabilidade de modelos entre diferentes BPMS, isto é, um BPMS pode não interpretar um processo modelado em outro BPMS. Com isso, o objetivo desta tese é propor *uma abordagem para avaliar os limites de implementação dos elementos BPMN*, em um trabalho que foi desenvolvido em 4 etapas.

A primeira etapa consiste na análise de quais elementos do BPMN estão implementados em BPMS. Para realizar essa análise, foi definido um protocolo de busca para identificar os BPMS a serem analisados, em seguida, aplicamos esse protocolo em um grupo de BPMS livre. Como resultado, descrevemos uma análise da cobertura do BPMN.

A segunda etapa trata da análise da frequência de elementos utilizados na modelagem de processos, em BPMN. A análise considerou 186 processos modelados em BPMN, obtidos de 3 fontes diferentes. Com essa análise, é possível avaliar e identificar um conjunto de elementos notacionais mais utilizados na modelagem de processos. São apresentados os critérios para seleção dos modelos, com base na necessidade deste trabalho. Em seguida, é apresentado um analisador que foi desenvolvido para gerar dados no formato *cvs* com a quantidade de elementos utilizados nos modelos de processos. Com os cvs obtidos, foi realizada uma análise dos resultados alcançados.

A terceira etapa apresenta um *survey* aplicado ao público-alvo do BPMN: os de-

senvolvedores de BPMS e analistas de processos. Trata-se de indivíduos que interagem diretamente com a notação, especialistas em BPMN e possuem conhecimento das limitações e aspectos que devem ser melhorados em na notação. A aplicação do *survey* foi através de um formulário com perguntas específicas considerando o *feedback* dos usuários em relação ao BPMN para realizar nossa pesquisa. Este formulário foi desenvolvido utilizando o *Google Forms*. A divulgação e compartilhamento do *survey* ocorreu entre outubro de 2019 e fevereiro de 2020. Para atingir o público da pesquisa, contatamos os desenvolvedores do BPMS e a consultoria de modelagem de processos por e-mail.

A quarta etapa focou no desenvolvimento de pseudo-algoritmos, como um caminho para incremento da notação BPMN. Com base nos resultados das analises obtidas nas etapas anteriores, e em trabalho prévio que tratou da extração da pseudo-algoritmos a partir de descrição textual, foram identificados os elementos para se aplicar a técnica de geração de pseudo-algoritmo. O mesmo foi constatado através de técnica de teste de software, baseado em tabela de decisão e grafo de fluxo de controle, para verificar se o pseudo-algoritmo desenvolvido segue a definição do BPMN o qual foi extraído.

A partir de todas as etapas executadas, foi obtido como resultados obtidos, observa-se que nem todos os elementos do BPMN estão implementados em BPMS. Além disso, são utilizados os mesmos elementos para modelagem dos processos (como por exemplo: tarefa, desvio exclusivo, dentre outros). Como resultado do *survey*, nós identificamos por exemplo, no contexto dos analistas de processo, que a notação oferece muitos elementos, que possuem as mesmas funcionalidades. No contexto dos desenvolvedores, foi identificado que a notação é normalmente adaptada, de acordo com a necessidade da organização. A última etapa da tese consistiu na elaboração de pseudo-algoritmos (algoritmos estruturados) para elementos já implementados no BPMS, adicionando funcionalidades complementares (por exemplo um elemento de desvio poder iniciar uma instância de processo), como uma sugestão de incremento do BPMN.

Como conclusões obtidas nessa tese, destaca-se que as análises realizadas permitiram identificar as limitações da BPMN, em termos de implementação em BPMS. Além disso, o desenvolvimento de pseudo-algoritmos possibilitou identificar um possível caminho de incremento da notação.

APPENDIX B — SPREADSHEET OF BPMS ANALYSIS

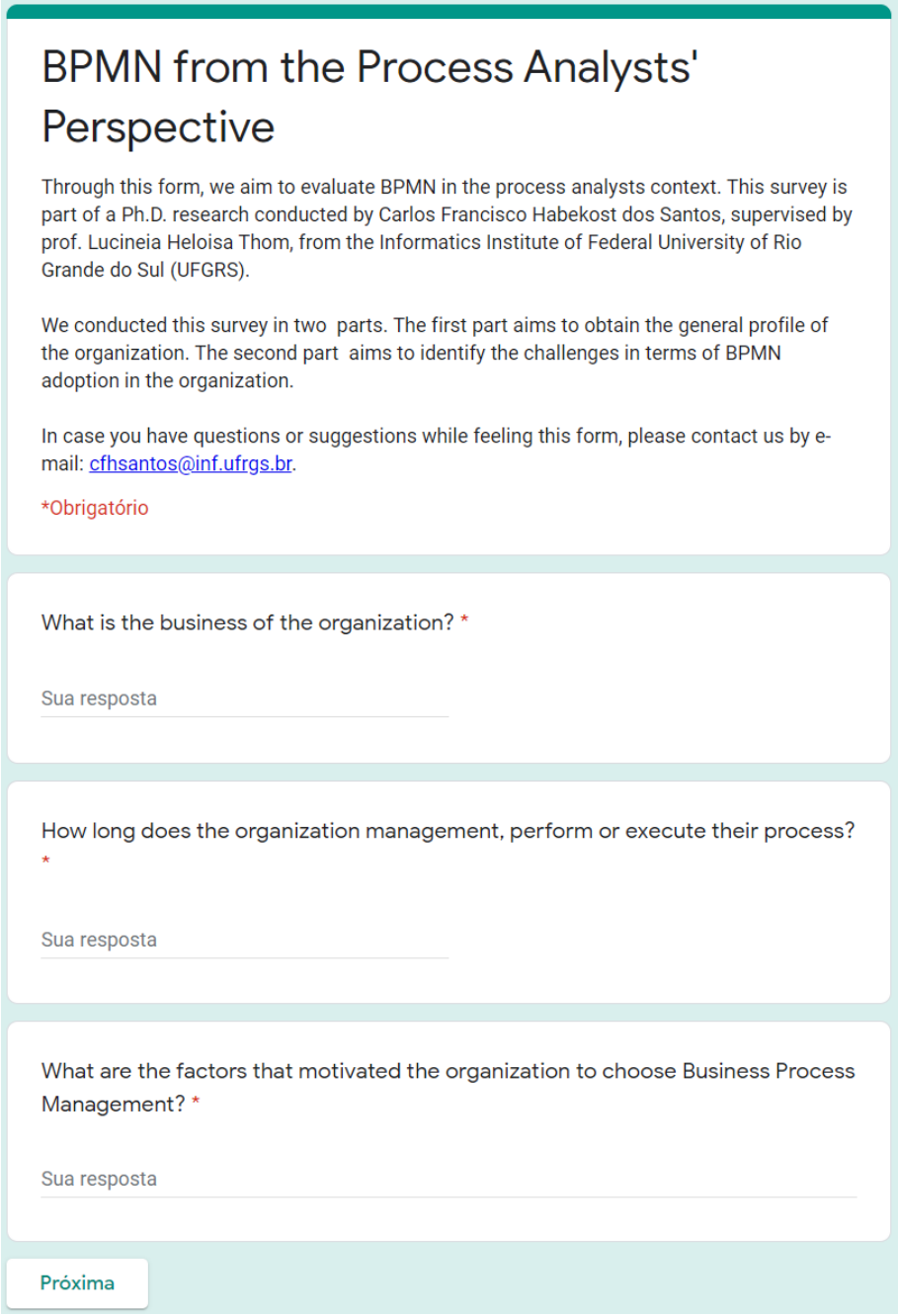
	Process Element/ Tool	Bonita BPM	WebRatio BPM	Camunda BPM	jBPM
Tasks	Abstract	X		X	X
	Service	X	X	X	X
	Send	X		X	X
	Receive	X		X	X
	User	X	X	X	X
	Manual		X	X	X
	Business Rule			X	
	Script	X		X	X
	Sub-process	X	X	X	
	Transaction				
Start Event	None	X	X	X	X
	Interrupting - Message	X	X	X	X
	Non-interrupting - Message				
	Interrupting - Time	X	X	X	X
	Non-interrupting - Time				
	Interrupting - Conditional		X	X	X
	Non-interrupting - Conditional				
	Interrupting - Signal	X	X	X	X
	Non-interrupting - Signal				
	Interrupting - Multiple		X		
	Non-interrupting - Multiple				
	Interrupting - Parallel Multiple				
	Non-interrupting - Parallel Multiple				
	Interrupting - Escalation				
	Non-interrupting - Escalation				
	Interrupting - Error	X			
	Interrupting - Compensation				
Intermediate Event	None	X	X	X	X
	Non-interrupting - Catch Message	X	X	X	X
	Throw Message	X	X	X	X
	Timer	X	X	X	X
	Non-Interrupting Timer	X			
	Conditional		X		
	Non-interrupting - Conditional				
	Catch Signal	X	X	X	X
	Non-interrupting - Catch Signal				
	Throw Signal	X	X	X	X
	Catch-multiple		X		
	Non-interrupting - Catch Multiple				
	Throw Multiple		X		
	Catch Parallel Multiple				
	Non-interrupting - Parallel Multiple				
	Catch Escalation				
	Non-interrupting Catch-escalation				
	Throw Escalation			X	X
	Catch Error	X	X		
	Catch Compensation		X		
	Throw Compensation		X	X	X
	Catch Link	X	X	X	X
	Throw Link	X	X	X	X
	Catch Cancel		X		

Figure B.1 – Spreadsheet with identified elements. Source: Author.

	Process Element/ Tool	Bonita BPM	WebRatio BPM	Camunda BPM	jBPM
End Event	None	X	X	X	X
	Message	X	X	X	X
	Signal	X	X	X	X
	Multiple		X		
	Escalation			X	X
	Error	X	X	X	X
	Compensation		X	X	X
	Cancel		X		
	Terminate	X	X	X	X
Gateways	Exclusive	X	X	X	X
	Inclusive	X	X	X	X
	Parallel	X	X	X	X
	Complex		X	X	X
	Event-based		X	X	X
	Event-based to Start a Process				
	Parallel event-based to Start a Process				
Artifacts	Group		X		
	Text Annotation	X	X	X	X
Swimlanes	Pool	X	X	X	X
	Lanes	X	X	X	X
Connection Objects	Sequence Flow	X	X	X	X
	Conditioal Sequence Flow	X	X	X	X
	Default Sequence Flow	X	X	X	X
	Message Flow			X	
	Initialing Message Flow				
	Non-initialing Message Flow				
	Association	X		X	X
	Directional Association				
	Bi-directional Association				
	Data Association				
	Directed Data Association				
	Conversation Link				

Figure B.2 – Spreadsheet with identified elements. Source: Author.

APPENDIX C — SURVEY FORMS



BPMN from the Process Analysts' Perspective

Through this form, we aim to evaluate BPMN in the process analysts context. This survey is part of a Ph.D. research conducted by Carlos Francisco Habekost dos Santos, supervised by prof. Lucineia Heloisa Thom, from the Informatics Institute of Federal University of Rio Grande do Sul (UFGRS).

We conducted this survey in two parts. The first part aims to obtain the general profile of the organization. The second part aims to identify the challenges in terms of BPMN adoption in the organization.

In case you have questions or suggestions while feeling this form, please contact us by e-mail: cfhsantos@inf.ufrgs.br.

*Obrigatório

What is the business of the organization? *

Sua resposta

How long does the organization management, perform or execute their process? *

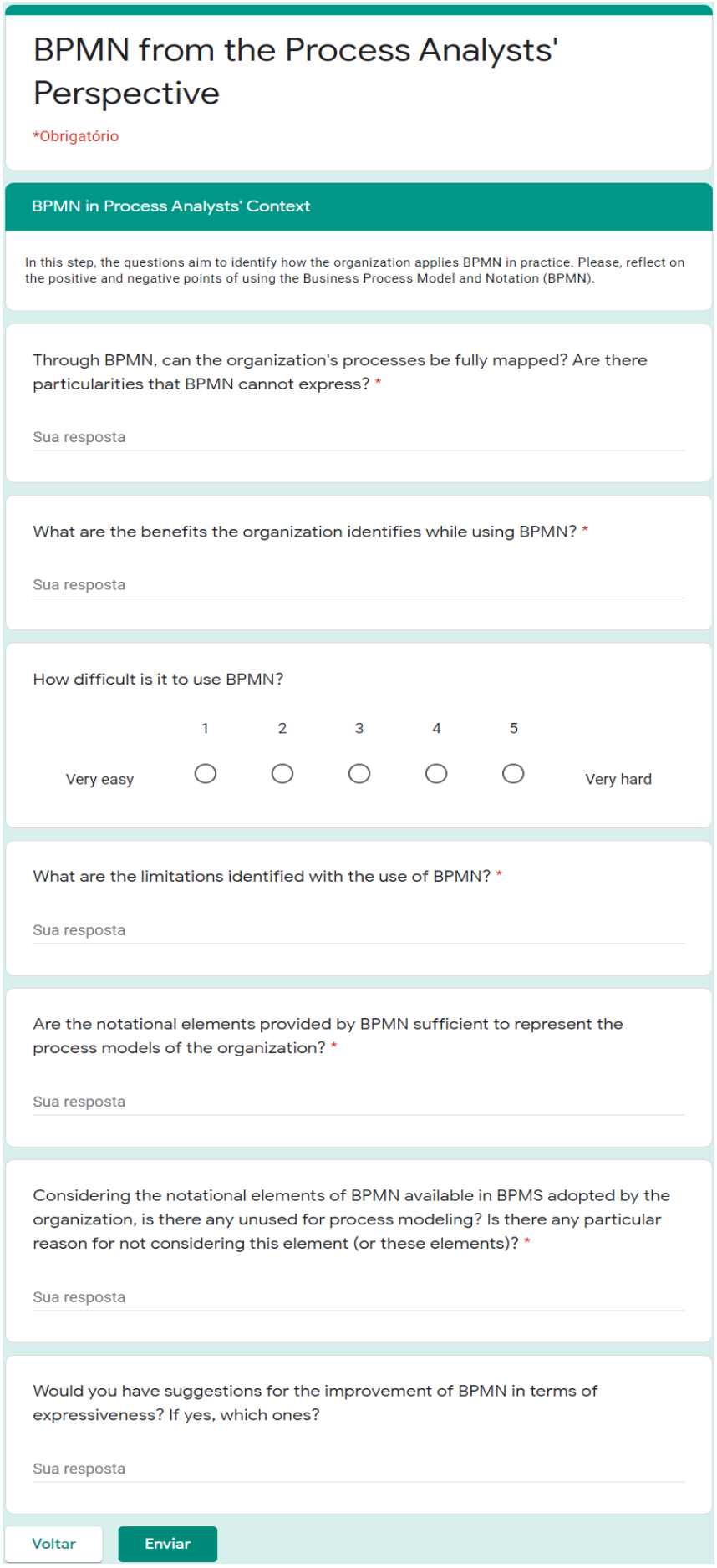
Sua resposta

What are the factors that motivated the organization to choose Business Process Management? *

Sua resposta

Próxima

Figure C.1 – Survey for Analysts - Part I.



BPMN from the Process Analysts' Perspective

*Obrigatório

BPMN in Process Analysts' Context

In this step, the questions aim to identify how the organization applies BPMN in practice. Please, reflect on the positive and negative points of using the Business Process Model and Notation (BPMN).

Through BPMN, can the organization's processes be fully mapped? Are there particularities that BPMN cannot express? *

Sua resposta

What are the benefits the organization identifies while using BPMN? *

Sua resposta

How difficult is it to use BPMN?

	1	2	3	4	5	
Very easy	○	○	○	○	○	Very hard

What are the limitations identified with the use of BPMN? *

Sua resposta

Are the notational elements provided by BPMN sufficient to represent the process models of the organization? *

Sua resposta

Considering the notational elements of BPMN available in BPMS adopted by the organization, is there any unused for process modeling? Is there any particular reason for not considering this element (or these elements)? *

Sua resposta

Would you have suggestions for the improvement of BPMN in terms of expressiveness? If yes, which ones?

Sua resposta

Voltar Enviar

Figure C.2 – Survey for Analysts - Part II.

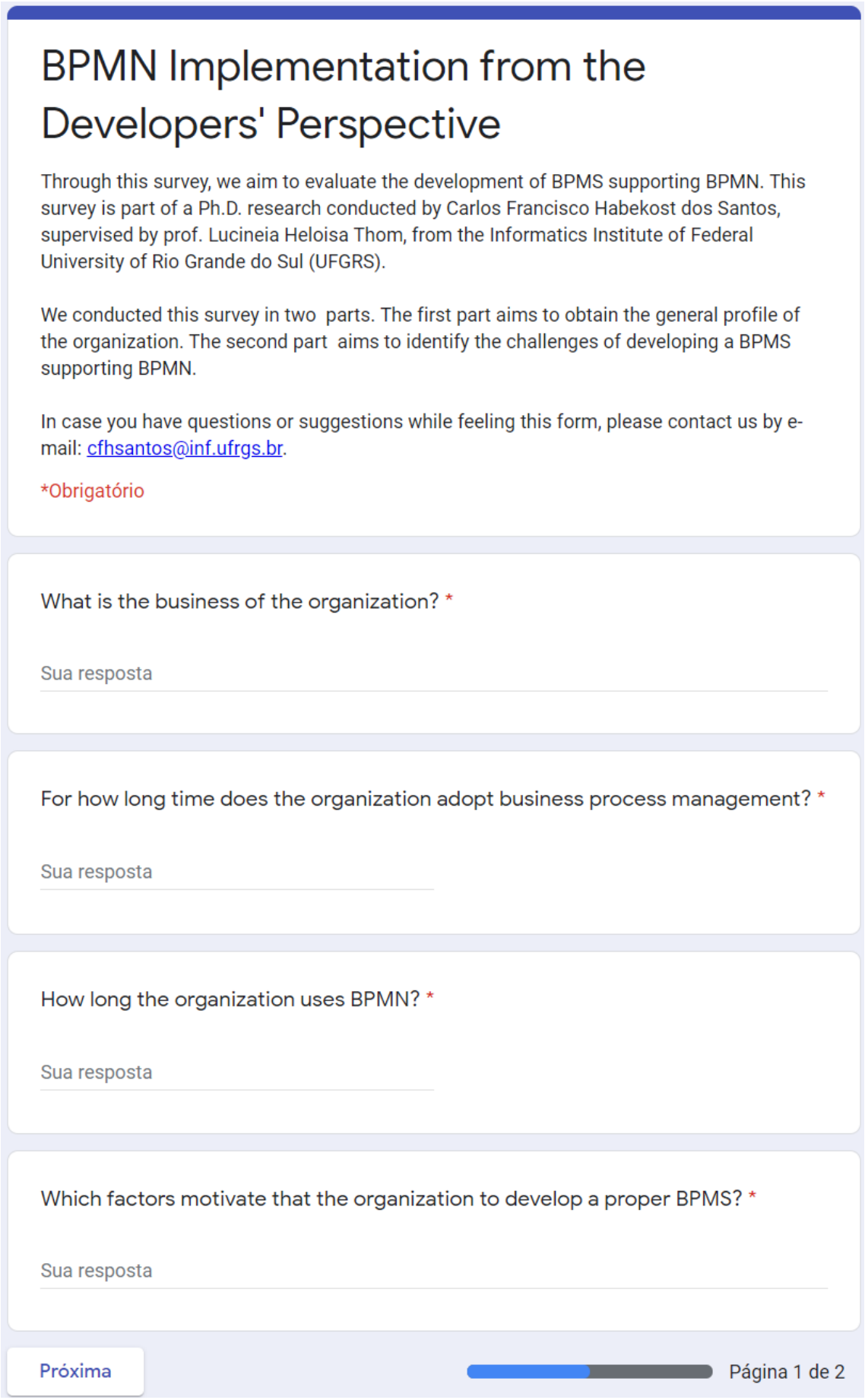


Figure C.3 – Survey for developers - Part I.

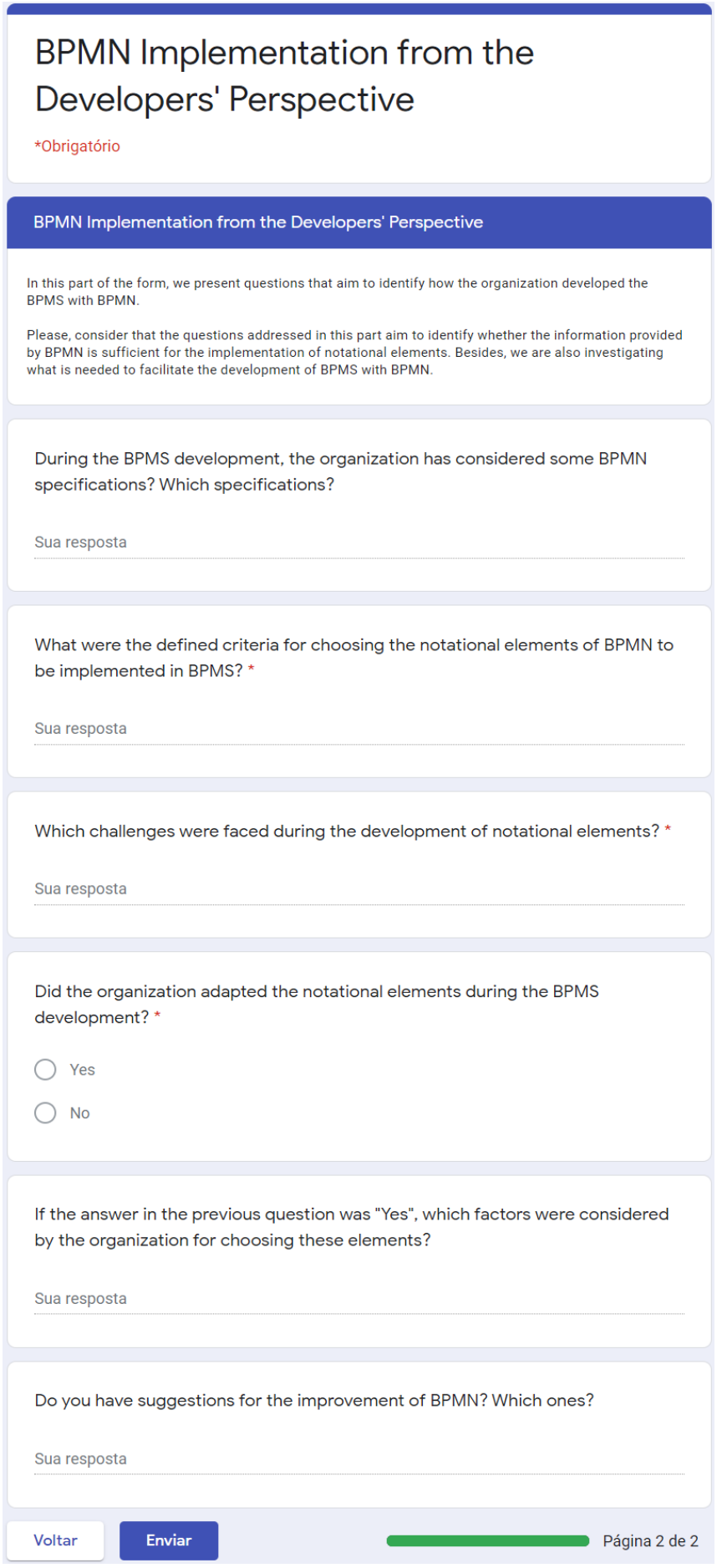


Figure C.4 – Survey for developers - Part II.

APPENDIX D — SCIENTIFIC CONTRIBUTIONS

This Appendix presents the publications of the author during his studies towards the PhD Degree in Computing at PPGC-UFRGS. There are two papers related to this thesis:

- Carlos Francisco Habekost dos Santos, Marcelo Fantinato, Érika Cota, Lucinéia Heloisa Thom. **Supporting BPMN tool developers through meta-algorithms**. (2019). International Journal of Business Information Systems (IJBIS).
  Status: Published. Qualis: B1;
- Carlos Francisco Habekost dos Santos, Marcelo Fantinato, José Palazzo Moreira de Oliveira, Lucinéia Heloisa Thom. **Analyzing Support for implementing BPMN 2.0 elements in Business Process Management Systems**. (2020). 22nd International Conference on Enterprise Information Systems (ICEIS 2020).
  Status: Published. Qualis: B2.

Also, the author has publication with the research group, as follow:

- Miller Biazus, Carlos Francisco Habekost dos Santos, Larissa Narumi Takeda, José Palazzo Moreira de Oliveira, Marcelo Fantinato, Jan Mendling, Lucinéia Heloisa Thom. **Software Resource Recommendation for Process Execution Based on the Organization's Profile**. (2019). International Conference on Database and Expert Systems Applications (DEXA).
  Status: Published. Qualis: B1.