UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

JOÃO VITOR DE CAMARGO

A Complementary Analysis of BPMN 2.0-Based Tools Behavior Regarding Process Modeling Problems

Work presented in partial fulfillment
of the requirements for the degree of
Bachelor in Computer Science

Advisor: Prof. Dr. Lucineia Heloisa Thom
Coadvisor: Msc. Vinicius Stein Dani

Porto Alegre
November 2021

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Reitor: Prof. Carlos André Bulhões
Vice-Reitora: Profª. Patrícia Helena Lucas Pranke
Pró-Reitora de Graduação: Profª. Cíntia Inês Boll
Diretora do Instituto de Informática: Profª. Carla Maria Dal Sasso Freitas
Coordenador do Curso de Ciência de Computação: Prof. Rodrigo Machado
Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

"Continuous effort - not strength or intelligence -
is the key to unlocking our potential."

— Sir Winston Churchill

"And I knew exactly what to do.
But in a much more real sense, I had no idea what to do."

— Michael Scott (Steve Carell)

AGRADECIMENTOS

Visto que esse trabalho representa o fim de uma etapa, é justo iniciar por reconhecer aqueles que estiveram presentes em todas as minhas etapas até aqui. Portanto, pelo suporte ao longo dos anos, sou grato aos meus pais, Mirtes e Oracilde, à minha irmã Sheila e à minha madrinha Lurdes. Agradeço também à minha parceira Vitória pela companhia e incentivo nos últimos 5 anos.

Sou igualmente grato aos professores e demais mestres que me guiaram durante os anos. Em especial, agradeço aos meus orientadores, Lucineia e Vinicius, pela dedicação ao nosso trabalho. Por fim, gostaria de agradecer a todos os colegas e amigos que nos últimos anos contribuíram para minha formação e desenvolvimento.

ABSTRACT

A business process model can be an essential asset for an organization, as it enables participants to understand the business processes in which they are involved. These models are mainly designed using automated process modeling tools and comply with the Business Process Model and Notation (BPMN) 2.0, an ISO standard widely accepted by the community. However, problems in process models may generate inconsistent interpretations and lead to the implementation of incorrect solutions. Therefore, process modeling tools that support BPMN 2.0 should be able to detect these problems. Still, the literature shows that modeling tools behave differently when facing the same problems. Hence, this study analyzes how BPMN 2.0-based process modeling tools are currently reacting and providing feedback about problems in business process models. As a study case, process modeling anti-patterns are used. These anti-patterns compose a class of commonly recreated modeling bad practices, diverse in type, cause and impact. This study starts by reviewing and complementing experiments learned from the literature to understand the current state of problem detection by modeling tools. In total, each of ten anti-patterns is modeled in ten modeling tools. With the results, the difference between paid and free modeling tools regarding feedback about problems is evaluated, together with an analysis of which types of problems are more often detected. Furthermore, problem feedback should be displayed understandably. Therefore, problematic models are created and the visual feedback about the problems within them is presented according to recommendations from the literature. Then, the ten modeling tools are exposed to these troubled models to analyze the visual aspects of their reaction to problems. Finally, the behavior presented by the tools is compared with the literature to evaluate which recommendations are followed by the tools and the current gaps on visual feedback presented by modeling tools.

Keywords: Business process management. BPMN. Business process modeling problems. Business process modeling anti-patterns. Visual Feedback.

Uma Análise Complementar do comportamento de ferramentas baseadas em BPMN 2.0 sobre Problemas de Modelagem de Processos

RESUMO

Modelos de processo de negócio podem ser um recurso essencial para uma organização, visto que permitem que os participantes compreendam o processo no qual estão inseridos. Tais modelos são majoritariamente criados através de ferramentas de modelagem de processos e seguem a Notação e Modelo de Processos de Negócio (*Business Process Model and Notation* - BPMN) 2.0, um padrão ISO amplamente aceito pela comunidade. Todavia, problemas em modelos de processos podem gerar interpretações inconsistentes, que acarretam em soluções incorretas. Portanto, ferramentas de modelagem de processos que suportam BPMN 2.0 devem ser capazes de detectar tais problemas de modelagem. Entretanto, a literatura mostra que ferramentas se comportam de forma inconsistente quando deparadas com o mesmo problema. Dessa forma, esse estudo analisa o estado atual de como ferramentas de modelagem de processos baseadas em BPMN 2.0 reagem e provêm *feedback* sobre problemas de modelagem. Anti-padrões de modelagem de processos são empregados como caso de estudo, pois compõem um conjunto de problemas de modelagem recorrentemente recriados, além de variados em tipo, causa e impacto. Este estudo recria e complementa experimentos descritos na literatura, com o objetivo de entender o estado atual de detecção de problemas por ferramentas de modelagem. Para isso, dez anti-padrões são modelados em dez ferramentas e com os resultados, são avaliadas diferenças entre ferramentas proprietárias e livres, além da análise de quais tipos de problema são detectados com mais frequência. Todavia, além de detectar problemas, ferramentas devem apresentar *feedback* visual sobre eles de forma compreensível. Assim, modelos problemáticos são criados e o *feedback* visual recomendado pela literatura para eles é mostrado. Então, as dez ferramentas são expostas a tais modelos para avaliar os aspectos visuais de suas reações. Por fim, os comportamentos das ferramentas são comparados com a literatura, para entender quais recomendações de *feedback* visual sobre problemas são implementadas pelas ferramentas.

Palavras-chave: Gerenciamento de processos de negócio. BPMN. Problemas de modelagem de processos de negócio. Anti-padrões de modelagem de processos de negócio. *Feedback* Visual.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

BPM	Business Process Management
BPMN	Business Process Model and Notation 2.0
BPMS	Business Process Management System
ISO	International Organization for Standardisation
OMG	Object Management Group
RQ	Research Question

CONTENTS

1 INTRODUCTION

A business process is a set of activities performed by an organization to deliver value to its customers, and one of the most valuable resources of an organization are their business processes (DUMAS et al., 2013). Business Process Management (BPM) is the discipline that studies the discovery, modeling, execution, and monitoring of business processes. Through BPM, organizations can obtain consistent results and identify improvements in their processes (DUMAS et al., 2013).

A business process can be graphically represented by a process model, which is an output of the process modeling task, an essential activity of BPM (DUMAS et al., 2013). A process model can be a key asset while, for example, developing a software solution (ROZMAN; POLANCIC; HORVAT, 2008). Additionally, a process model can also be an important resource in communicating the processes of an organization, as it enables the participants to understand the process in which they are involved (DUMAS et al., 2013).

Several modeling notations are proposed in the literature, focusing on solving different problems. Some notable examples (MILI et al., 2010) are: Petri Nets, Unified Modeling Language (UML), Event-driven Process Chains (EPC), and Business Process Model and Notation (BPMN). In this study, we focus on BPMN, version 2.0 (MANCARELLA, 2011), because it is widely accepted by the BPM community (CHINOSI; TROMBETTA, 2012) and supported by a large set of commercial process modeling tools (GEIGER et al., 2018). Moreover, BPMN is specified by the Object Management Group (OMG) (OMG, 2011), and standardized by ISO[1].

1.1 Motivation

Modeling business processes is a complex and prone to error task, as ambiguity leads to multiple interpretations and the lack of a shared understanding of the process (DONGEN; AALST; VERBEEK, 2005). Moreover, the inability to continuously translate business requirements into process models is one of the main causes for failing software projects (BARJIS, 2008). Finally, a low-quality process model can generate poorly described software requirements, leading to a problematic information system (ROZMAN; POLANCIC; HORVAT, 2008).

[1]ISO/IEC 19510:2013: http://www.omg.org/spec/BPMN/ISO/19510/PDF

Modeling patterns, i.e., good practices, for business process models are described in the literature (FELLMANN et al., 2019), and process modelers can benefit from applying them (KOSCHMIDER; LAUE; FELLMANN, 2019) in their process modeling tasks. Contrarily, process modeling anti-patterns describe common errors detected in business process models (KOSCHMIDER; LAUE; FELLMANN, 2019).

A process modeling anti-pattern is not necessarily related to syntax errors. While classifying anti-patterns, Koschmider et al. (KOSCHMIDER; LAUE; FELLMANN, 2019) proposed seven anti-patterns categories, being “Syntax errors” only one of them. In addition, only 7 out of the 15 most common anti-patterns in BPMN process models detected by Rozman et al. (ROZMAN; POLANCIC; HORVAT, 2008) are related to syntax errors. The others are related to semantic or pragmatic problems. Many business process modeling tools can verify the syntax correctness of process models. However, validating, for example, the semantics of a model is still a challenge (ROZMAN; POLANCIC; HORVAT, 2008).

We only infer that providing feedback regarding anti-patterns, and modeling problems in general, is not a trivial task for business process modeling tools. However, considering that BPMN is standardized by ISO, we understand that the tools that support this process modeling notation should also behave in a standard way towards detected problems in this notation.

1.2 Research Questions and Objectives

To comply with the notation, business process modeling tools that support BPMN must provide feedback to the process modeler at least about syntactical errors. However, as learned from literature (ROZMAN; POLANCIC; HORVAT, 2008; KOSCHMIDER; LAUE; FELLMANN, 2019), common problems detected in process models are not only about syntactical errors. Therefore, our first research question is:

RQ1: What is the current state of modeling problems detection presented by business process modeling tools?

To answer this question, we follow descriptions of experiments reported in the literature (DIAS et al., 2019). We review and complement these experiments with the objectives to: i) evaluate if process modeling tools have been updated in the past years to provide better problem detection; ii) investigate if free and paid modelings tool react differently to modeling problems; and iii) verify if problem detection is more often perceived for problems classified as syntax problems, given semantic and pragmatic validations are more challenging for process modeling tools (ROZMAN; POLANCIC; HORVAT, 2008). As previous experiments (DIAS et al., 2019), we use anti-patterns (ROZMAN; POLANCIC; HORVAT, 2008) as a study case, as they represent a commonly repeated set of problems diverse in type, cause, and impact.

In addition to the first research question, we also consider that: process modeling tools usually present feedback about problems using non-instructive text messages (DIAS et al., 2019); and, particularly in cases of large process models, it is often not possible to identify in which process modeling element the issue is located (DANI; FREITAS; THOM, 2019b). As recommendations for visual feedback about problems are described in the literature (DANI; FREITAS; THOM, 2019b), our second research question is:

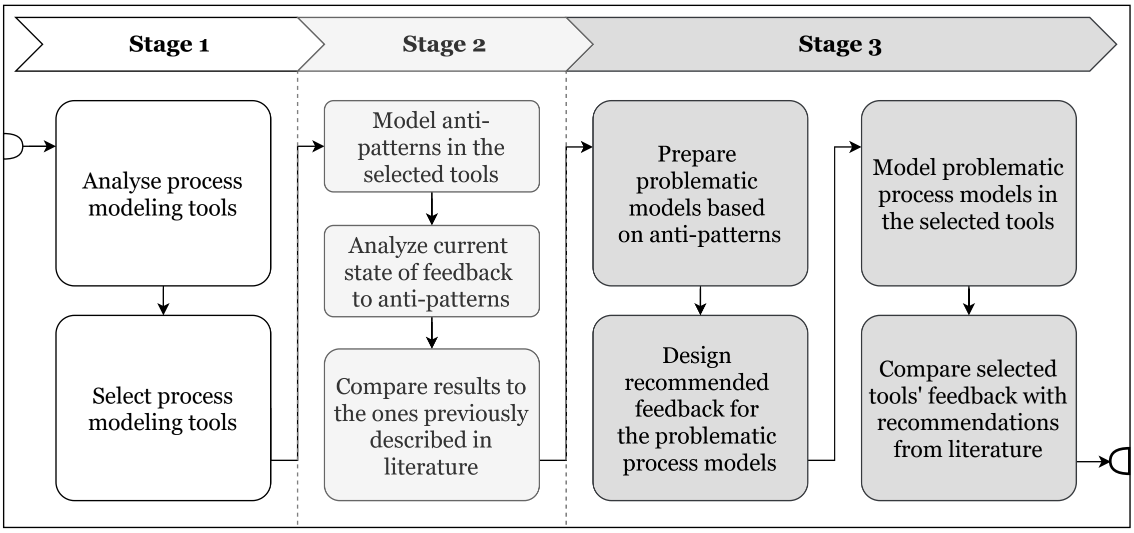
RQ2: Does the visual feedback about problems provided by business process modeling tools follow the literature recommendations (DANI; FREITAS; THOM, 2019b)?

To answer this question, we: i) measure how similar are process modeling tools, in terms of feedback about problems, when compared to recommendations for feedback about problems from literature (DANI; FREITAS; THOM, 2019b); and ii) identify which - if any - are the feedback gaps, i.e., recommendations not commonly implemented among the studied tools.

1.3 Research Methodology

Given our RQs and objectives, here we propose a three-stage methodology for this research, illustrated in Figure 1.1. In stage 1, we analyze available process modeling tools to select the ones used in further stages. In stage 2, we recreate the experiments performed by Dias et al. (DIAS et al., 2019), and then compare and analyze the differences, while complementing their study with a new set of analyzed process modeling tools. Finally, in stage 3, we create problematic process models based on anti-patterns, design the recommended feedback for these models based on literature (DANI; FREITAS; THOM, 2019b), model them in the selected tools and, at last but not least, compare the tools results with the recommendations from literature (DANI; FREITAS; THOM, 2019b).

Figure 1.1: Methodology for this study



Source: the authors.

In stage 2, we answer RQ1 by recreating and complementing the experiments described by Dias et al. (DIAS et al., 2019). Thus, we can understand the current state of feedback about process modeling anti-patterns, and verify the evolution on problem feedback among the tools analyzed in both our study and in (DIAS et al., 2019). Finally, in stage 3 we answer RQ2 by analyzing the visual feedback about problems provided by the selected modeling tools and comparing it with the recommendations proposed by Dani et al. (DANI; FREITAS; THOM, 2019b).

1.4 Text Organization

This work is organized as follows. First, in Chapter 2 we present the necessary background, covering the fundamentals of BPM, its life-cycle, and the process modeling task; and, we discuss the related works. Chapter 3 presents process modeling problems in business process models; syntactic, semantic, and pragmatic issues are illustrated, as well as process modeling anti-patterns. Chapter 4 introduces the selected business process modeling tools for this study, explaining the selection criteria, and providing relevant information about each tool. In Chapter 5, we report on the analysis of how each process modeling tool reacts to a set of anti-patterns. For the tools where there are results documented in the literature, we present a comparison between their previous and current feedback. In Chapter 6, we report on the analysis of how each process modeling tool behaves visually when exposed to problems in process models. Moreover, we present the recommended literature on visual feedback about problems, together with a comparison to the actual feedback currently perceived in the tools. Finally, the last chapter presents our conclusions, covering limitations, and suggestions for future works.

2 FUNDAMENTALS ON BUSINESS PROCESS MANAGEMENT AND RELATED WORKS

We start this chapter by providing fundamental concepts of BPM for this study (cf., Section 2.1), such as the BPM life-cycle, and the main BPMN elements. We close this chapter with a discussion about related works (cf., Section 2.2).

2.1 Business Process Management

BPM is the discipline responsible for understanding how work is performed in an organization. In the BPM context, organizations are not only enterprises but also governmental agencies, non-profit organizations, or any other institution that has to manage many processes, enabling the discipline to be relevant in multiple environments (DUMAS et al., 2013). Moreover, BPM is a multidisciplinary field of study, having its bases on multiple areas of business administration and computer science, receiving considerable attention from both communities (WESKE, 2007).

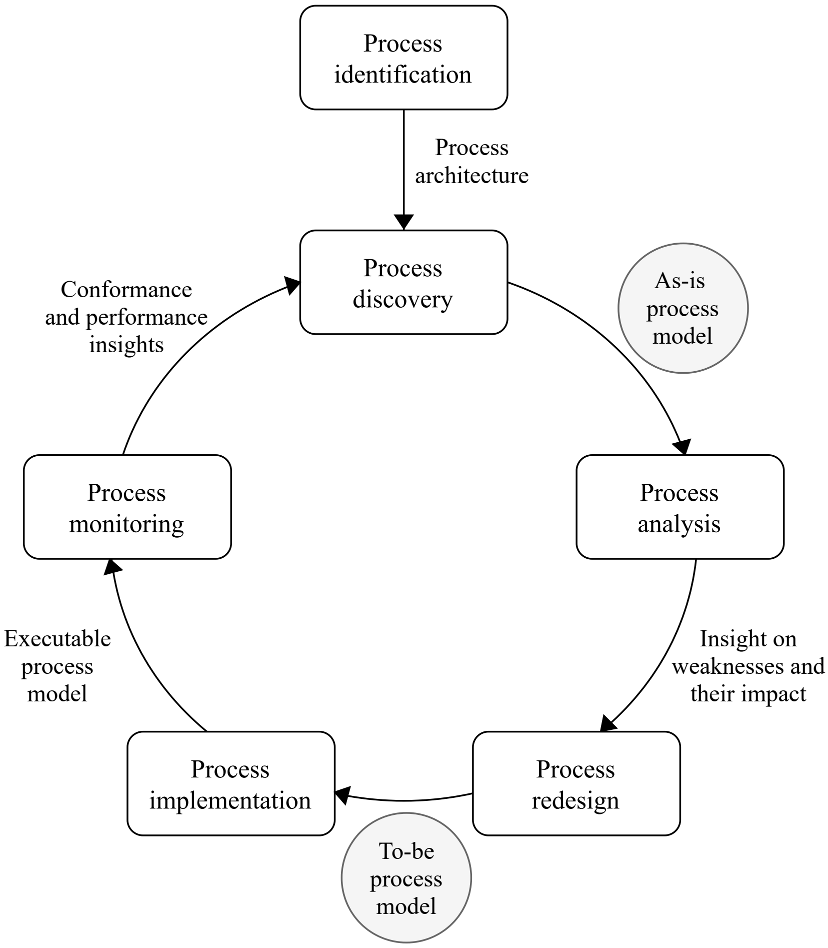
A business process is a set of activities performed in an organization to achieve a business goal (WESKE, 2007). As an example, the process of issue reporting on a software provider company can happen as follows: it starts when the customer notices an issue in the software, which they report to the software provider company. Next, the support engineer analyzes the issue, classifying it as a customer mistake or a software bug. The latter generates demand for the software developer responsible for fixing the software. Finally, the support engineer replies to the customer (either explaining their mistake or providing a solution), and the customer closes the issue. The goal of the software vendor with this process is to provide support for the customer.

However, as any other process, the one above must be constantly monitored, maintained, and enhanced to adapt to the always-changing customer needs. The lack of this continuous improvement leads the process to become degraded (DUMAS et al., 2013).

2.1.1 Business Process Management life-cycle

The BPM life-cycle is a set of phases that describe the existence of a business process. To enable the continuous improvement of the process, the BPM life-cycle must be cyclic, enabling the outcomes from the monitoring phase to serve as input for new iterations of the cycle (DUMAS et al., 2013). The phases proposed by (DUMAS et al., 2013) are process identification, process discovery, process analysis, process redesign, process implementation, and process monitoring. Figure 2.1 displays the life-cycle, highlighting phases where process models, the focus of our works, are generated.

Figure 2.1: BPM life-cycle highlighting phases where process models are generated. The *as-is* model is one outcome of process discovery, while the *to-be* model is produced in the process redesign



Source: adapted from (DUMAS et al., 2013).

During the process discovery, the process is understood to create the *as-is* process model, which represents the current state of the process. Further in the cycle, the process redesign phase proposes an improved version of the process through the *to-be* process model. The process models produced on the discovery and redesign phases are used throughout the entire life-cycle of BPM (DUMAS et al., 2013)

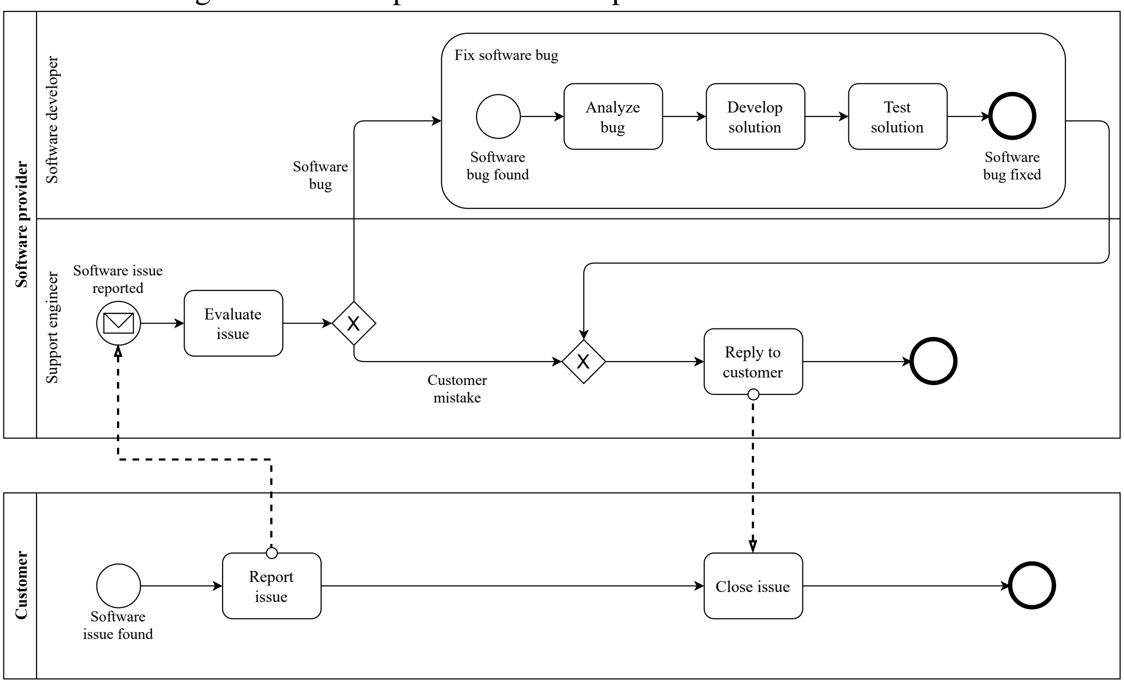
2.1.2 Business Process Model and Notation

Models are created using a notation. For business process models, BPMN is currently the standard. The notation is described as accessible for stakeholders (who are in charge of the business processes) while still having a technical aspect, which allows the translation into software (OMG, 2011). BPMN is composed of multiple modeling elements, being able to represent different types of processes. Some of the basic elements are: tasks, sub-processes, flows (sequence and message), events (start, intermediate, and end), gateways (*AND*, *OR*, and *XOR*), pools and lanes.

A process begins with a *start event*, and finishes with an *end event*. *Intermediate events* are the ones that happen during the process and trigger activities. A single activity is named by a *task*, while a group of them compose a *sub-process*.

Participants perform activities, e.g., a department in the organization, a business partner, or even the customer. They are represented by *pools*, and a pool can be split into multiple *lanes*. When on the same pool, activities are connected through a *sequence flow*. Otherwise, a *message flow* is used. Decisions in the process are represented by *gateways*. Figure 2.2 illustrates the BPMN model of the previously described issue reporting process in a software provider company.

Figure 2.2: Example of a business process modeled in BPMN



Source: the authors

Although simple, the model in Figure 2.2 presents several BPMN elements. It is composed of two pools, one for the customer and one for the company, where the latter is split into two lanes - one for the support engineer and one for the software developer. A decision is necessary, represented by an exclusive (*XOR*) gateway after the support engineer evaluates the issue. There is a sub-process, named *Fix software bug*, performed by the software developer. It is possible to find instances of other previously mentioned modeling elements throughout the process, such as sequence and message flow and start and end events.

2.1.3 Modeling problems and anti-patterns

Although BPMN is a standardized notation for process modeling, problems can exist in process models. A modeling problem is generated through the incorrect usage of BPMN elements, according to the notation syntax and semantic (ROZMAN; POLANCIC; HORVAT, 2008). However, a process model can be problematic by not matching business requirements (BARJIS, 2008) or by being designed in a way that impairs readability (ROZMAN; POLANCIC; HORVAT, 2008).

Widely discussed in the software engineering field, anti-patterns are faulty solutions that are recurrently reinvented (KOENIG, 1998). In the BPM context, a process modeling anti-pattern is the common inadequate usage of BPMN elements during the modeling task (KOSCHMIDER; LAUE; FELLMANN, 2019). Modeling problems and anti-patterns are discussed in detail in Chapter 3.

2.2 Related Works

In this section, we split the related works into two categories: business process modeling anti-patterns and visual feedback about problems in process models. Studies about anti-patterns are related to our discussions in Chapter 5, and the ones on visual feedback about problems serve as a basis for Chapter 6.

2.2.1 Anti-patterns on business process models

The topic of anti-patterns for business process models has been relevant for over a decade now. The work of Rozman et al. (ROZMAN; POLANCIC; HORVAT, 2008) lists the 15 most common modeling anti-patterns, based on a large set of models produced by BPM students. More recently, Koschmidel et al. (KOSCHMIDER; LAUE; FELLMANN, 2019) created a taxonomy of 48 articles regarding business process modeling anti-patterns, classifying them into seven categories. The authors highlight that there is a broad literature regarding categories *control-flow* (e.g. deadlocks) and *understandability* (e.g. complexity), but a lack of studies about *composition* (cooperation among participants) and *ecological impact* anti-patterns. A similar study, but covering modeling patterns, was produced by Fellmann et al. (FELLMANN et al., 2019).

Anti-patterns detection in business process models was studied by Koehler et al. (KOEHLER; VANHATALO, 2007). In this work, the authors analyzed hundreds of "real world" business process models to extract a group of anti-patterns and guide modelers on how to detect and avoid modeling them. The same guidance was provided by Rozman et al. (ROZMAN; POLANCIC; HORVAT, 2008) on their 15 anti-patterns, while the identification of anti-patterns was also covered by Lehmann et al. (LEHMANN et al., 2020). In the latter, the authors discuss modeling anti-patterns in Enterprise Architecture (EA) models, proposing a selection of 18 anti-patterns (based on (KOSCHMIDER; LAUE; FELLMANN, 2019)) for this specific class of models.

The subject of anti-patterns in modeling tools was researched by Dias et al. (DIAS et al., 2019). In this work, the authors selected ten anti-patterns among the ones described by Rozman et al. (ROZMAN; POLANCIC; HORVAT, 2008) and modeled them in four commercial modeling tools. The objective was to analyze how modeling tools react to these patterns.

As previously mentioned, one of our objectives is to recreate the experiments performed by Dias et al. (DIAS et al., 2019) to review if the modeling tools they studied were updated to be able to detect anti-patterns, while complementing with a new set of tools for which such results are not found in the literature. To the best of our knowledge, no other related works approach the current behavior of modeling tools regarding anti-patterns, motivating our RQ1.

2.2.2 Visual feedback about problems on business process models

Many authors have studied visual feedback about problems in process models in the last few years. For example, a Systematic Literature Review (SLR) was performed by Dani et al. (DANI; FREITAS; THOM, 2019a) regarding visualization of business process models. The authors selected 46 papers and then classified them into six categories. One of them is the *Visual feedback concerning problems detected in process models*, described by the authors as “less explore” and representing “challenges for further exploration”.

Based on their SLR (DANI; FREITAS; THOM, 2019a), the Dani et al. (DANI; FREITAS; THOM, 2019b) performed a survey with 57 participants to understand the modelers’ demands regarding visual feedback about problems in process models. Combining the survey answers, with the literature and the behavior of analyzed modeling tools, the author proposed recommendations on how a modeling tool should provide feedback to the modeler, covering scenarios with both small and large models (DANI; FREITAS; THOM, 2019b). No further studies were found about how modeling tools react visually to business process modeling problems.

Another example of related work is the one performed by Hipp et al. (HIPP et al., 2014), which proposes alternatives to visualize large and complex process models. However, as others found during our research, this visualization study does not cover problems in these models. Therefore, to the best of our knowledge, there is no study more updated than (DANI; FREITAS; THOM, 2019a) regarding the current state of visualization of problems in business process models.

2.3 Final comments

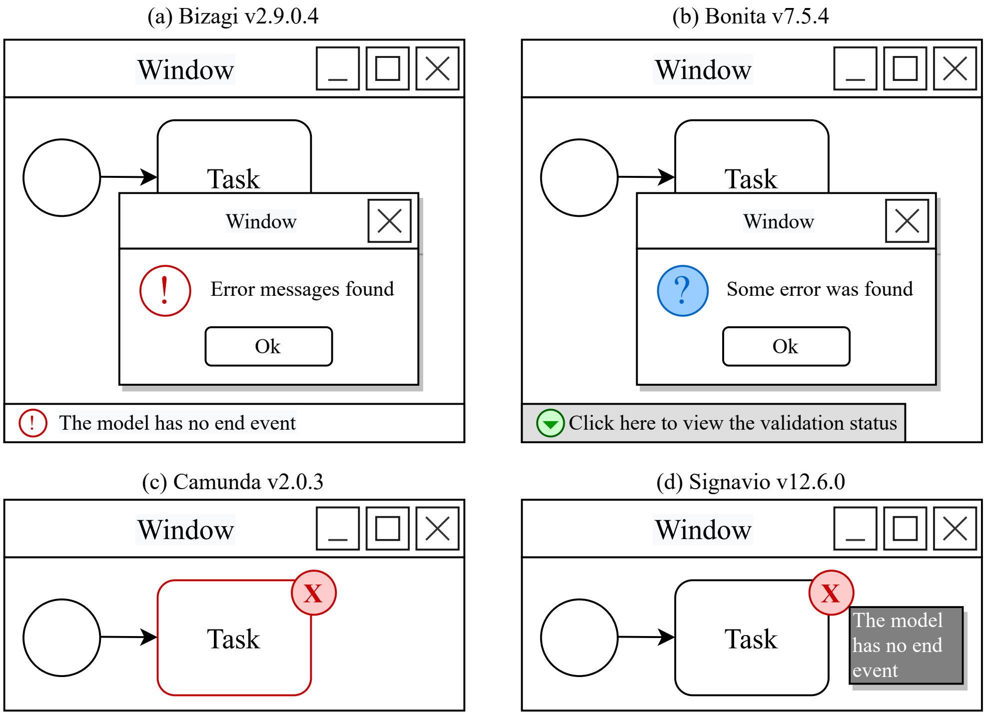
This chapter provided the fundamentals to understand our study better and then presented the related works. For fundamentals, we covered BPM, its life-cycle, BPMN, and a brief introduction about process modeling problems and anti-patterns. In the related works, we discussed studies about modeling anti-patterns and visual feedback about modeling problems.

3 PROBLEMS WITHIN BUSINESS PROCESS MODELS

Mapping business process into models is an essential step for organizations that desire to improve their productivity, quality, or compliance (MENDLING; REIJERS; AALST, 2010). However, although modeling software provides support to users, especially regarding syntax errors (ROZMAN; POLANCIC; HORVAT, 2008), the uncertainty on how to create understandable models for both analysts and business stakeholders (MENDLING; REIJERS; AALST, 2010) can lead to problematic models.

The feedback about problems provided by process modeling tools is very heterogeneous, even if they follow BPMN guidelines. For example, even when all tools can detect a problem in the model, the feedback differs visually (DANI; FREITAS; THOM, 2019b), as illustrated in Figure 3.1.

Figure 3.1: Modeling Tools displaying visual feedback differently for an anti-pattern. The pattern is "*Missing end event*.".



Source: (DANI; FREITAS; THOM, 2019b)

However, the feedback is not necessarily only visually divergent. When facing the same problems, modeling tools can behave significantly differently. For example, while modeling the anti-pattern "*Each lane in the pool contains start event*" in four commercial modeling tools, (DIAS et al., 2019) found that two provided errors in different moments, one provided a warning, and one did not detect the anti-pattern as a problem.

3.1 Types of business process modeling problems

The quality of a business process model can be evaluated through different perspectives. In the literature, there are concepts such as physical, empirical, social, and organizational quality of the process (KROGSTIE; SINDRE; JØRGENSEN, 2006). However, the three most recurrently related to modeling problems are syntax, semantic and pragmatical validations (ROZMAN; POLANCIC; HORVAT, 2008). Therefore, these three problem types are explained in detail.

Syntax problems in business process models

A syntax modeling problem is the one that violates the syntax of the notation, i.e., BPMN (CLAES; VANDECAVEYE, 2019). However, when the elements in the model are according to the vocabulary of the notation, syntactic quality is achieved (KROGSTIE; SINDRE; JØRGENSEN, 2006). Furthermore, most process model validation frameworks and modeling tools support syntax validation (BOCK; CLAES, 2018; ROZMAN; POLANCIC; HORVAT, 2008), which is the only type that can be completely achieved objectively, as others may need human interaction (KROGSTIE; SINDRE; JØRGENSEN, 2006).

The majority of syntax errors can be identified by automated tools (HAISJACKL et al., 2018), as the syntax description of BPMN is described objectively (DIJKMAN; DUMAS; OUYANG, 2008). However, some instances of syntax problems are pictured by modeling anti-patterns, where the less experienced users frequently generate a subset of common errors due to lack of notation comprehension (KOSCHMIDER; LAUE; FELLMANN, 2019; ROZMAN; POLANCIC; HORVAT, 2008).

Semantic problems in business process models

A semantic error impairs the semantic quality of the model. This quality is measured through them metrics of validity, completeness, and feasibility. The three compose the ability of the model to be complete and per its domain (KROGSTIE; SINDRE; JØRGENSEN, 2006). There is also the perceived quality of the model, which compares the participants' business knowledge to their interpretation of the model (KROGSTIE; SINDRE; JØRGENSEN, 2006).

Semantic problems can be an outcome of the confusion caused by the mix of constructs and instructions presented in BPMN and the lack of clarity on the semantics of such elements (DIJKMAN; DUMAS; OUYANG, 2008). An instance of semantic validation discussed in the literature is about start and events, where a validation method is proposed to guarantee the correct usage according to the notation semantic (CHINOSI; TROMBETTA, 2009). Although start and end events are optional (OMG, 2011), their omission characterize anti-patterns (ROZMAN; POLANCIC; HORVAT, 2008) and it is a modeling guideline to use one of each in a process model (MENDLING; REIJERS; AALST, 2010).

Pragmatic problems in business process models

Pragmatic problems are the ones that difficult understandability. The process participants should be able to understand the model (KROGSTIE; SINDRE; JØRGENSEN, 2006). Although some pragmatic issues can be detected by automated tools (WEBER et al., 2011), there is still the need for a human inspection of the model quality (HAISJACKL et al., 2018)

An example of a pragmatic issue is the usage of labels that are not useful to describe the purpose of an activity (WEBER et al., 2011). Furthermore, to achieve process model understandability, several definitions such as modeling guidelines (MENDLING; REIJERS; AALST, 2010) and modeling patterns (FELLMANN et al., 2019) are available in the literature. Modeling *smells* (i.e. bad practices) are also presented (WEBER et al., 2011). Finally, avoiding modeling anti-patterns (ROZMAN; POLANCIC; HORVAT, 2008; KOSCHMIDER; LAUE; FELLMANN, 2019) enables a better pragmatic quality of the process model.

3.2 Business process modeling anti-patterns

The lack of business domain and process modeling notation knowledge leads modelers to create problematic models. Problems commonly repeated in these models are named process modeling anti-patterns (ROZMAN; POLANCIC; HORVAT, 2008). These anti-patterns are diverse in problem type, impact, and cause. Therefore, they represent a good set of issues to understand how modeling tools react to problems. Now, we briefly review ten modeling anti-patterns learned from the literature (ROZMAN; POLANCIC; HORVAT, 2008). These specific set of anti-patterns was selected for this work because it was already used on experiments with modeling tools documented in the literature (DIAS et al., 2019).

Anti-pattern 1: Activities in one pool are not connected

Problem description: this anti-pattern is characterized by a missing sequence flow between activities of the same pool. This problem is illustrated on Tasks C and D of Figure 3.2.a. The technical implication is an unreachable activity since it has no incoming flow. This problem is often caused by the lack of understanding that the process is contained within a pool (OMG, 2011) and should not depend on other pools to be correct (ROZMAN; POLANCIC; HORVAT, 2008).

Problem classification: this pattern is both a syntactical and pragmatical problem (ROZMAN; POLANCIC; HORVAT, 2008). BPMN guidelines state that all elements within a process (pool) must be connected (OMG, 2011).

Anti-pattern 2: Process does not contain an end event

Problem description: the lack of end events, presented in Figure 3.2.b, in the process characterize this problem (ROZMAN; POLANCIC; HORVAT, 2008). This anti-pattern impairs the model readability due to uncertainty of when the process ends (ROZMAN; POLANCIC; HORVAT, 2008).

Problem classification: end events are optional in a model (OMG, 2011). Therefore, this is not a syntax error, but a pragmatical, as it decreases understandability (ROZMAN; POLANCIC; HORVAT, 2008).

Anti-pattern 3: Sequence flow crosses sub-process boundary

Problem description: this anti-pattern is caused by the usage of a sequence flow connecting elements inside and outside of a sub-process (ROZMAN; POLANCIC; HORVAT, 2008), as portrayed in Figure 3.2.c. This practice is incorrect since a sub-process is a separate entity referenced from a parent process, and the elements of both are not accessible to each other (OMG, 2011).

Problem classification: a sequence flow should not cross the sub-process boundary (OMG, 2011), hence this anti-pattern is a syntactical error. It also decreases the understandability of the model (ROZMAN; POLANCIC; HORVAT, 2008).

Anti-pattern 4: Sequence flow crosses pool boundary

Problem description: this anti-pattern is characterized by the usage of a sequence flow between activities located in different pools, as presented in Figure 3.2.d. The implication of this problem is creating an incorrect dependency among separate pools, which represent different processes (ROZMAN; POLANCIC; HORVAT, 2008; OMG, 2011).

Problem classification: sequence flows should not cross pool boundaries, since pools represent separate processes, for which the communication should use message flows (OMG, 2011). Therefore, this anti-pattern is a syntax error (ROZMAN; POLANCIC; HORVAT, 2008).

Anti-pattern 5: Gateway receives, evaluates or sends a message

Problem description: in this anti-pattern, a gateway incorrectly receives and evaluates a message (ROZMAN; POLANCIC; HORVAT, 2008). This problem is illustrated in Figure 3.2.e. A gateway is a modeling element used to diverge and converge sequence flows, therefore is not responsible for receiving, evaluating, and sending messages (OMG, 2011). The impact of this pattern is modelers assuming the message is used by the gateway, which is not capable of processing it (ROZMAN; POLANCIC; HORVAT, 2008).

Problem classification: apart from the incorrect usage of the gateway, message flows should only be used among activities (tasks and sub-processes) and should not be linked to a gateway (OMG, 2011). Therefore, this is a syntax error (ROZMAN; POLANCIC; HORVAT, 2008).

Anti-pattern 6: Intermediate events are placed on the edge of the pool

Problem description: placing intermediate events in the edge of the pool, such as the one in Figure 3.2.f, leads to events not having an incoming flow, hence being unreachable within the process. The incorrect usage of these elements may lead to an incorrect interpretation that events can happen anytime in the process (ROZMAN; POLANCIC; HORVAT, 2008).

Problem classification: an intermediate event should be fully connected within the process, with incoming and outgoing flows (OMG, 2011). Therefore, this problem is a syntax error (ROZMAN; POLANCIC; HORVAT, 2008).

Anti-pattern 7: Hanging intermediate events or activities

Problem description: this anti-pattern happens when an activity or event that is left hanging in the process, i.e., has no incoming flow (ROZMAN; POLANCIC; HORVAT, 2008). Task B is hanging on Figure 3.2.g. This problem generates uncertainty about where the process begins (ROZMAN; POLANCIC; HORVAT, 2008).

Problem classification: since start events are optional, the hanging activity can be interpreted as an implicit start to the process, not characterizing a syntax error (OMG, 2011). However, this is a semantic issue since such activity may not be reachable (ROZMAN; POLANCIC; HORVAT, 2008).

Anti-pattern 8: Each lane in the pool contains start event

Problem description: this anti-pattern is characterized by the usage of a start event for each lane of a process, as in Figure 3.2.h. People in the organization who read the model may consider that the activities performed by each participant (lane) are entirely independent, which may not always be the case (ROZMAN; POLANCIC; HORVAT, 2008).

Problem classification: BPMN allows multiple start events in a process (OMG, 2011), therefore this anti-pattern is not a syntactical error. Instead, it is considered pragmatically incorrect, with the implication of the model being ambiguous (ROZMAN; POLANCIC; HORVAT, 2008).

Anti-pattern 9: Exception flow is not connected to the exception

Problem description: an exception is an event that triggers an interruption of the normal flow. As illustrated in Figure 3.2.i, an exception event can be attached to an activity, and in this case, should have an exception flow, i.e., a path that will lead to an exception handling (OMG, 2011). This anti-pattern is caused by the lack of an exception flow outgoing of the exception event (ROZMAN; POLANCIC; HORVAT, 2008).

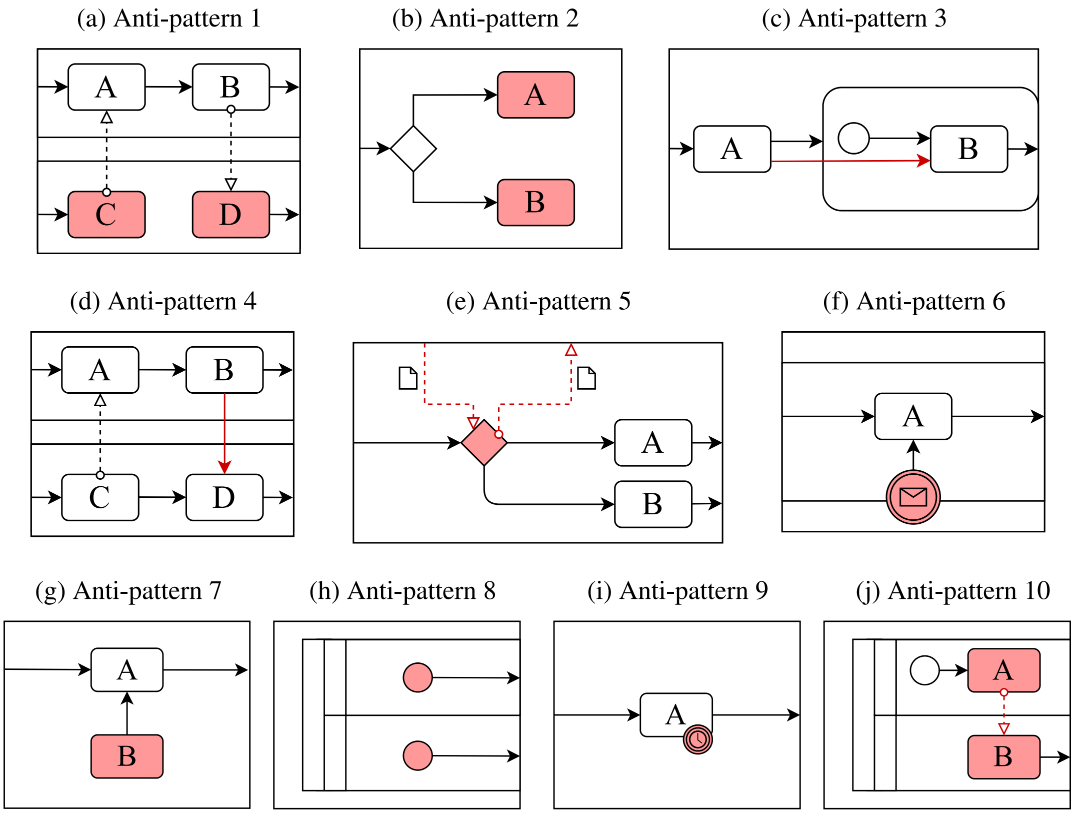
Problem classification: as the exception flow could be connected to the task where the exception event is attached, this problem is not necessarily a syntax error (ROZMAN; POLANCIC; HORVAT, 2008). Instead, this is semantically incorrect, as a separate activity would be necessary to handle the triggered exception (ROZMAN; POLANCIC; HORVAT, 2008; OMG, 2011).

Anti-pattern 10: Message flow used inside the pool

Problem description: the connection between activities of the same process is represented by a sequence flow, while the message flow is specific for communication among different pools (OMG, 2011). Using the message flow inside a pool, as in Figure 3.2.j, leads to uncertainty while executing the process (ROZMAN; POLANCIC; HORVAT, 2008).

Problem classification: as the usage of message flow inside the pool is not compliant to the notation, this pattern is a syntax error (OMG, 2011).

Figure 3.2: Ten business process modeling anti-patterns. Colored modeling elements either cause or are affected by the anti-pattern.



Source: adapted from (ROZMAN; POLANCIC; HORVAT, 2008)

3.3 Final Comments

This chapter presented the concept of problems in business process models. Then, it described different problem types. Finally, the ten modeling anti-patterns used in this work were presented. For each pattern, the problem description and classification are presented, explaining its cause and impact.

4 BUSINESS PROCESS MODELING TOOLS

In the real world, process models can be complex and large. These models often demand maintainability effort and must be accessible by several participants of the process. Drawing models in pen and paper, for example, can help sketch. However, it is not suitable for sharing among the organization (DUMAS et al., 2013). In this scenario, process modeling tools have enabled models to be more easily standardized, stored, and shared (MENDLING; REIJERS; AALST, 2010).

As described by our objectives, we want to understand how modeling tools are currently reacting to anti-patterns and later compare their provided visual feedback to the literature's recommendation. Therefore, this chapter lists the modeling tools under study, explaining the selection criteria and then introducing each tool.

4.1 Selection criteria

The selection criteria for the modeling tools focus mainly on their availability and support of BPMN. They are as follows:

1: The tool must support creating (or importing) BPMN models;

2: The tool must support validating the syntax and correctness of BPMN models;

3: The tool must be free or offer a free available version (e.g., trial or academic editions).

To start the selection, all modeling tools studied by (DIAS et al., 2019) were included. They are, in alphabetical order: Bizagi[1], Bonita[2], Camunda[3], and Signavio[4]. In (DUMAS et al., 2013), the authors enumerate a list of modeling tools, containing Adonis[5], ARIS[6], Bizagi, Camunda, IBM Blueworks Live[7], and Signavio. Then, we selected the top 6 modeling tools rated by customers (GARTNER, 2021), which are, as ranked: Microsoft Visio[8], ARIS, Adonis, Signavio, Mavim[9] and iGrafx[10]. Finally, an additional

[1] Bizagi: <http://www.bizagi.com>
[2] Bonita: <http://www.bonitasoft.com>
[3] Camunda: <http://www.camunda.com>
[4] Signavio: <http://www.signavio.com>
[5] Adonis <http://www.boc-group.com/en/adonis/>
[6] ARIS: <http://www.ARIScommunity.com/>
[7] IBM Blueworks Live: <http://www.ibm.com/products/blueworkslive>
[8] Microsoft Visio: <http://www.microsoft.com/microsoft-365/visio/flowchart-software>
[9] Mavim: <https://www.mavim.com/>
[10] iGrafx: <https://www.igrafx.com/>

research in the Internet found OracleBPM[11] and QualiBPMN[12] as viable options to be analyzed.

Except for iGrafx and Mavim, all tools fulfilled all three ICs and were selected for this study. The excluded tools both fail to satisfy selection criteria 3, as no version of their modeling solutions was found accessible.

4.2 Selected tools

According to the presented criteria, our study's selected business process modeling tools are Adonis, ARIS, Bizagi, Bonita, Camunda, IBM Blueworks Live, Microsoft Visio, OracleBPM QualiBPMN, and Signavio. The edition and version used for each tool can be found in Table 4.1. When the tool has only one available edition, the column is filled with *Standard*. No official version information was found for QualiBPMN, but the tool was studied from May 15th to May 23rd of 2021.

Table 4.1: Edition, availability and version of the selected process modeling tools

Modeling Tool	Edition	Availability	Version
Adonis	Community	Web	11
ARIS	Basic	Web	10.0.13.1
Bizagi Modeler	Standard	Local	3.8.0.191
Bonita BPM	Community	Local	7.12.1.1
Camunda Modeler	Standard	Local	4.6.0
IBM Blueworks Live	Standard	Web	June 2021
Microsoft Visio	Professional	Local	2106
Oracle BPM Studio	Standard	Local	12c
QualiBPMN	Standard	Web	-
Signavio Process Manager	Academic	Web	14.16.0

Source: the authors

Most of the selected tools are considered Business Process Management Systems (BPMS). A BPMS is a system that enables modeling, execution, and monitoring of processes (DUMAS et al., 2013). However, since we focus on business process models, we limit our study of these tools to their modeling and validation features. In 2020, the vendors of ARIS, Adonis, Camunda, and Signavio participated in the *BPMN in Action!*, an event hosted by OMG to showcase the current status of BPMN-based modeling tools (OMG, 2020).

[11]Oracle BPM Studio: <https://www.oracle.com/middleware/technologies/bpm.html>
[12]QualiBPMN: <http://cloudfreebpmnquality.herokuapp.com/analytics/>

Adonis

Adonis is a process management tool developed by BOC Group. The company was founded in 1995 and was related to the University of Vienna, Austria (BOC, 2021b). The first versions of Adonis were produced in the early years after the company foundation. Adonis is currently available in 3 editions: Adonis Community Edition (CE), Adonis Starter Edition (SE), and Adonis Enterprise Edition (EE). All versions are fully compliant with BPMN 2.0 (BOC, 2021a). Adonis CE is the only one free to use.

Adonis CE had its first versions between 2005 and 2010 (BOC, 2021b) and is currently available online (via browser), with no need to download or install locally. Adonis CE is considered a *freeware*, free distribution of proprietary software. From now on, when mentioning *Adonis*, we will be referring to Adonis CE since it is the edition used in our study.

ARIS

ARIS is a business process modeling tool maintained by German company Software AG (AG, 2021). The company also offers ARIS Community, a forum where ARIS users can interact and discuss the solution. Currently, ARIS Community has more than 600.000 registered members (COMMUNITY, 2021).

The ARIS modeler has three available editions: Basic, Advanced, and Enterprise, which vary in user capacity, support, and customization (BASTIAN, 2021). ARIS Express is also offered for free, but it does not support BPMN. The used version in this study is ARIS Basic, through a three-month academic license. From now on, we will refer to the modeler of ARIS Basic as simply *ARIS*.

Bizagi

The company Bizagi was founded in 1989 and currently offers a set of tools for BPM. The one used in this study is *Bizagi Modeler*, a modeling application. However, the company also provides Bizagi Studio - built to simulate and automate processes, and Bizagi Automation - built to execute and manage processes automatically. The company proposes that customers use them in order to model, automate, and execute their processes (BIZAGI, 2021b).

The first version of Bizagi Modeler was launched in 2008, and by 2015, the tool reached 3 million downloads (BIZAGI, 2021a). Bizagi Modeler is freeware, available through download and local installation. From now on, in this work, we will use *Bizagi* to refer to the Bizagi Modeler, unless if explicitly mentioned otherwise.

Bonita

Bonita BPM is a BPMS currently maintained by french company Bonitasoft. Even though Bonitasoft was only founded in 2009, Bonita, as a software, has existed since 2001 (BONITASOFT, 2021a). Furthermore, Bonitasoft's documentation frequently refers to Bonita BPM as simply *Bonita*, which we will adopt from now on.

Bonita is currently available in two editions: Bonita Community and Bonita Enterprise (BONITASOFT, 2021c). Both editions of Bonita are open-source[13]. The Community edition is free, available through download and local installation, and it is the one used in this study.

One of the differential features in Bonita is the REST API Interface, which allows integrating Bonita with other software (BONITASOFT, 2021b). Furthermore, developers can extend the API, enabling applications to integrate with Bonita according to their demands.

Camunda

The German company Camunda was founded in 2008, supplying BPM consulting. In 2013, they released the Camunda Platform, a set of software applications that enable modeling, executing, monitoring, and optimizing processes (CAMUNDA, 2021c). Camunda Platform comprises seven solutions, separated into three groups: design, automation, and improvement. For designing models, the focus of this study, the solutions are Camunda Modeler for developers, and Cawemo, a modeler for business stakeholders. (CAMUNDA, 2021c).

Camunda Modeler is open-source[14] and uses bpmn.io, a also open-source[15] modeling platform, developed and maintained by the Camunda community (CAMUNDA, 2021b; CAMUNDA, 2021a). The free edition of Camunda Modeler, used in this work,

[13]The source code for Bonita is at <https://github.com/bonitasoft>
[14]The source code for Camunda Modeler is at: <https://github.com/camunda/camunda-modeler>
[15]The source code for *bpmn.io* is at: <https://github.com/bpmn-io>

comes "without any warranties", as support is part of the Camunda Platform Enterprise edition, which is paid (CAMUNDA, 2021b). For simplification, further in this work, when mentioning *Camunda*, we will be referring to the Camunda Modeler, unless when explicitly mentioned otherwise.

IBM Blueworks Live

IBM Blueworks Live is a process modeling tool developed and maintained by the American software company IBM. Blueworks Live is part of *IBM cloud computing*, a set of cloud-native business solutions (IBM, 2021).

IBM Blueworks Live is accessible through a web client, with no local installation. The software is paid, with no permanently free edition. Instead, we analyzed the tool under a 30-day trial. From now on, we will refer to IBM Blueworks Live as just *Blueworks*.

Microsoft Visio

Microsoft Visio is a diagramming tool offered by American company Microsoft. Visio allows modeling different types of diagrams, from generic flowcharts to business process models (MICROSOFT, 2021). The tool is considered a generic modeling platform and not a specialized business process modeler (DUMAS et al., 2013).

The tool is available for both web and local clients. However, the web version of Visio does not support BPMN modeling, which is only available in the local version, obtained through download and installation (MICROSOFT, 2021). Microsoft Visio is a paid software with no permanently free edition. Instead, both web and local clients were analyzed within a one-month trial of the solution. From now on, we will refer as *Visio* to the local version of Microsoft Visio used in this study.

Oracle BPM Studio

Oracle BPM Studio is a BPM tool offered by American company Oracle. It is part of Oracle BPM Suite, a set of BPM tools offered by the company (ORACLE, 2021b).

Oracle BPM Studio is part of the Oracle JDeveloper IDE, accessible through local installation (ORACLE, 2021a). JDeveloper is a free development tool (ORACLE, 2021c), and so is Oracle BPM Studio. To simplify, we will now address Oracle BPM Studio as only Oracle BPM.

QualiBPMN

QualiBPMN is a free software[16] focused on validating business process models (FREEBPMNQUALITY, 2021b). The tool is part of *freebpmnquality*, a "digital ecosystem" for working with BPMN models (FREEBPMNQUALITY, 2021a). QualiBPMN development phase began in 2020, as the outcome of research from the Kharkiv Polytechnic Institute, in Ukraine (FREEBPMNQUALITY, 2021a). QualiBPMN can be used as a JavaScript library. However, we used a free online client available for the tool (FREEBPMNQUALITY, 2021b).

QualiBPMN differs from other tools in this study because *i)* it is not a commercial tool, and *ii)* does not have a native modeling feature. Therefore, it is necessary to import models to perform validation. However, since QualiBPMN is a tool specified in providing problem feedback in BPMN models and had its bases on academic research (FREEBPMNQUALITY, 2021a), we decided to consider the tool in our study. It is important to reinforce that QualiBPMN is a tool in development, and no official version information is available for the tool. For this study, we performed our experiments in QualiBPMN from May 15th to May 23rd.

Signavio

The company Signavio was founded in 2009 by students of the Hasso Plattner Institute in Germany (SIGNAVIO, 2021d). The founders claimed that companies were not benefiting from BPM due to the lack of an accessible platform, leading the BPMN practices to be non-existing or unmanageable for many organizations (SIGNAVIO, 2021d).

The platform Signavio first appeared in 2006, and together with the company foundation in 2009, it was launched as the first fully web-based collaborative BPM software (SIGNAVIO, 2021d). In 2021, Signavio was acquired by German company SAP SE (SIGNAVIO, 2021b). As part of Signavio Business Transformation Suite, the company offers solutions *Process Intelligence* for process mining, *Process Manager* for modeling, executing and optimizing, and *Workflow Accelerator*, focused on automating pro-

[16] The source code for QualiBPMN is at <https://github.com/freebpmnquality/qualibpmn>

cess (SIGNAVIO, 2021c).

As part of its BPM Academic Initiative (SIGNAVIO, 2021a), Signavio offers Process Manager and Workflow Accelerator platforms entirely free for students. In this work, we use the modeling solution of Signavio Process Manager, accessible through academic login in a web client, with no need for download or installation. Finally, from now on, we will use *Signavio* to refer to the modeler solution in Signavio Process Manager, unless if explicitly mentioned otherwise.

4.3 Final comments

In this chapter, we first presented the selection criteria for selecting process modeling tools. Then, we listed the analyzed tools and which were selected according to our ICs. Finally, we described the solution for each tool selected, covering its availability, developers, and general characteristics.

5 ANALYSIS OF BUSINESS PROCESS MODELING TOOLS BEHAVIOR REGARDING BUSINESS PROCESS MODELING PROBLEMS

After selecting the modeling tools, we can verify how they react to anti-patterns. For four out of the ten tools, there are results documented in the literature (DIAS et al., 2019). To review and complement this analysis, we model the ten most common modeling anti-patterns (ROZMAN; POLANCIC; HORVAT, 2008) in every selected tool, which enables us to understand the current state of problem feedback in modeling tools. At the end of this chapter, we use this outcome to answer RQ1 and achieve the research objectives related to the question.

Since QualiBPMN has no modeling feature, we consider that *i)* the tool allowed modeling if the model is successfully imported and the anti-pattern is visually displayed and that *ii)* modeling feedback is the outcome of validation performed on an imported model. As the other nine tools have native modeling functionalities, we model each anti-pattern from scratch on them.

5.1 Anti-patterns in business process modeling tools

Modeling tools can respond differently when facing an anti-pattern. Tools may allow the anti-pattern to be modeled or not, and in both scenarios, may provide feedback about the problem. All four combinations can be observed, as illustrated in Figure 5.1, where anti-pattern *Message flow used inside pool* is used as an example.

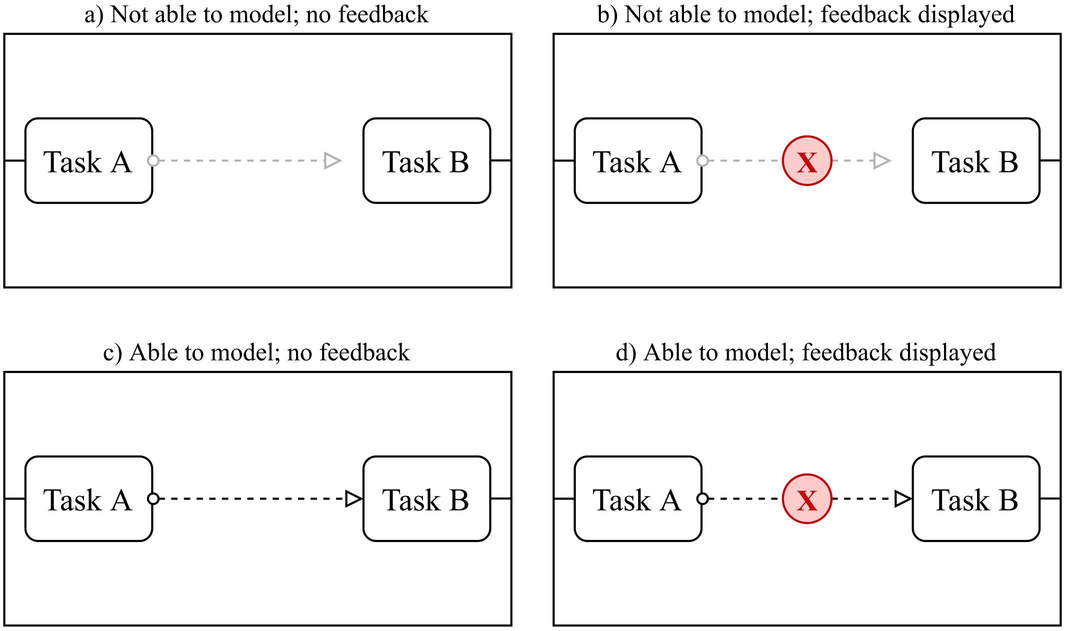
In the first two cases, modeling the anti-pattern is not possible, but case B displays problem feedback. The same is valid for the last two cases, where modeling is possible, but only case D provided feedback. Therefore, for every anti-pattern in each tool, we will inform if the tool *i)* enabled modeling the problem and *ii)* provided any feedback about it.

Anti-pattern 1: Activities in one pool are not connected

Adonis, *Bizagi* and *Camunda*: allowed modeling. Did not provide any form of feedback about the problem.

Aris, *Oracle BPM*, *QualiBPMN* and *Visio*: allowed modeling. Triggered errors about the miss-connected tasks not having incoming and outgoing sequence flows.

Figure 5.1: Possible response of a modeling tool when facing an anti-pattern. In this example, the modeler attempts to connect two elements in the same pool with a message flow.



Source: the authors

Blueworks: did not allow modeling. The tool only allows adding new activities connected to an existing sequence flow. When trying to remove the flow between the tasks, provides a message explaining the action would lead to an invalid process.

Bonita: allowed modeling. Raised a warning informing the user that the task without the incoming flow will be used as a implicit start to the process.

Signavio: allowed modeling. Advised the user, through a warning, to connect the modeling elements such that it originates in a start event and leads to an end event.

Table 5.1: Anti-pattern 1 - current problem detection and feedback. A marked cell identifies that the tool presented the behavior described on the column.

Modeling Tool	Allowed modeling	Raised warning	Raised error
Adonis	•		
Aris	•		•
Bizagi	•		
Blueworks			•
Bonita	•	•	
Camunda	•		
Oracle BPM	•		•
QualiBPMN	•		•
Signavio	•	•	
Visio	•		•

Source: the authors

Anti-pattern 2: Process does not contain an end event

Adonis, *Bizagi* and *Camunda*: allowed modeling. Did not provide any form of feedback about the problem.

Aris: allowed modeling. Showed an error attached to start event, advising that an end event should exists since a start event is used.

Blueworks: did not allow modeling. The tool already puts an end event in the process by default and it is not possible to remove it. The user is notified that all elements should be connected with incoming and outgoing flows.

Bonita and *Signavio*: allowed modeling. Provided warnings that end events might be missing.

Oracle BPM, *QualiBPMN* and *Visio*: allowed modeling. Raised error, claiming that there are tasks without an outgoing flow. Oracle BPM and Visio also informed that the process should have an end event.

Table 5.2: Anti-pattern 2 - current problem detection and feedback. A marked cell identifies that the tool presented the behavior described on the column.

Modeling Tool	Allowed modeling	Raised warning	Raised error
Adonis	•		
Aris	•		•
Bizagi	•		
Blueworks			•
Bonita	•	•	
Camunda	•		
Oracle BPM	•		•
QualiBPMN	•		•
Signavio	•	•	
Visio	•		•

Source: the authors

Anti-pattern 3: Sequence flow crosses sub-process boundary

Adonis, *Bizagi*, *Blueworks*, and *Bonita*: did not allow modeling. The sequence flow crossing the sub-process boundary is dismissed without any feedback on why.

Aris, *Camunda* and *Oracle BPM*: did not allow modeling. When dragging the sequence flow to an activity inside the sub-process, the tools indicate the operation is not valid. Aris and Oracle BPM display a forbidden icon, while Camunda painrs the destiny task as red.

QualiBPMN: did not allow modeling. Triggers an error that the sequence flow should connect to the sub-process.

Signavio and *Visio*: allows modeling. Both tools raise errors informing that a sequence flow should not cross a process boundary.

Table 5.3: Anti-pattern 3 - current problem detection and feedback. A marked cell identifies that the tool presented the behavior described on the column.

Modeling Tool	Allowed modeling	Raised warning	Raised error
Adonis			
Aris			•
Bizagi			
Blueworks			
Bonita			
Camunda			•
Oracle BPM			•
QualiBPMN			•
Signavio	•		•
Visio	•		•

Source: the authors

Anti-pattern 4: Sequence flow crosses pool boundary

Adonis, *Signavio* and *Visio*: allowed modeling, but provided errors. Adonis informs that elements connected by a sequence flow should in the same pool, while Signavio and Visio notify that such flow should not cross process boundaries.

ARIS: did not allow modeling. The sequence flow is automatically changed to a message flow, without feedback,

Bizagi: did not allow modeling. The flow is only visually connected, and the tool raises a generic message about incorrectly connected elements. We do consider that the pattern was neither possible to model or detected by Bizagi, in compliance to previous experiments present in the literature (DIAS et al., 2019).

Bonita: did not allow modeling. Showed a forbidden icon when dragging the flow to an activity in another pool.

Blueworks and *Oracle BPM*: not possible to model. The tools only allow modeling one process (pool) at a time.

Camunda: did not allow modeling. The flow is dismissed without feedback.

QualiBPMN: did not allow modeling. Raised errors that the tasks connected by the crossing sequence flow have too several incoming/outgoing flows.

Table 5.4: Anti-pattern 4 - current problem detection and feedback. A marked cell identifies that the tool presented the behavior described on the column.

Modeling Tool	Allowed modeling	Raised warning	Raised error
Adonis	•		•
ARIS			
Bizagi			
Blueworks			
Bonita			•
Camunda			
Oracle BPM			
QualiBPMN			•
Signavio	•		•
Visio	•		•

Source: the authors

Anti-pattern 5: Gateway receives, evaluates or sends a message

Adonis and *Bizagi*: did not allow modeling. In Adonis, the message flow is dismissed without feedback. In Bizagi, the flow is visually displayed, but not connected, similarly to the last pattern.

Aris, *Bonita* and *Camunda*: did not allow modeling. When dragging the message flows to the gateway, both tools show icons indicating the operation is invalid.

Oracle BPM and *Blueworks*: not possible to model. As the tools only allow modeling one pool at a time, there is no option to add message flows.

Signavio: did not allow modeling. Automatically places the message flow in edge of the tool, and then provide a warning that it will be ignored.

QualiBPMN: not possible to model. Since QualiBPMN has no native modeling feature, the model has to be imported to it. Only Visio allowed modeling this problem, but the tools are not interoperable, hence we are not able to analyze this pattern in QualiBPMN.

Visio: allowed modeling. The tool provided an error informing that a gateway must not have incoming or outgoing flows.

Anti-pattern 6: Intermediate events are placed on the edge of the pool

Adonis, *ARIS*, *Bonita* and *Camunda*: did not allow modeling. All three tools extend the pool length to keep the event inside of it. Adonis and ARIS display a forbidden icon when placing the event on the edge, while Bonita and Camunda informs that an

Table 5.5: Anti-pattern 5 - current problem detection and feedback. A marked cell identifies that the tool presented the behavior described on the column.

Modeling Tool	Allowed modeling	Raised warning	Raised error
Adonis			
Aris			•
Bizagi			
Blueworks			
Bonita			•
Camunda			•
Oracle BPM			
QualiBPMN			
Signavio		•	
Visio	•		•

Source: the authors

incoming flow is missing in the event through error and a warning, respectively.

Bizagi, *Oracle BPM*, *Signavio* and *Visio*: allowed modeling and provided errors. All tools notify that an incoming flow is missing to the event.

Blueworks: not possible to model. The tool only allows adding messages between activities. When trying to remove the flow to event, the tools does not allow, informing that such event needs an incoming flow.

QualiBPMN; did not allow modeling. The tools informs that a task in which the event is connected has too many incoming flow.

Table 5.6: Anti-pattern 6 - current problem detection and feedback. A marked cell identifies that the tool presented the behavior described on the column.

Modeling Tool	Allowed modeling	Raised warning	Raised error
Adonis			•
ARIS			•
Bizagi	•		•
Bonita			•
Blueworks			•
Camunda		•	
Oracle BPM	•		•
QualiBPMN			•
Signavio	•		•
Visio	•		•

Source: the authors

Anti-pattern 7: Hanging intermediate events or activities

Adonis and *Bizagi*: allowed modeling. Did not detect the anti-pattern as problem, ignoring the hanging task without any feedback.

ARIS, *Oracle BPM*, *QualiBPMN* and *Visio*: allowed modeling. Raised errors notifying the user that the hanging task should have an incoming flow.

Blueworks: not possible to model. The tool only allow adding tasks to an already existing sequence flow. When trying to delete this flow to force the pattern, the tool provide a message that such action is invalid, as it leads to an inconsistent state of the process.

Bonita, *Camunda* and *Signavio*: allowed modeling. Camunda and Bonita provided warnings that an incoming flow is missing to the hanging task. Signavio warns that all elements should be connected from start to end events.

Table 5.7: Anti-pattern 7 - current problem detection and feedback. A marked cell identifies that the tool presented the behavior described on the column.

Modeling Tool	Allowed modeling	Raised warning	Raised error
Adonis	•		
ARIS	•		•
Bizagi	•		
Blueworks			•
Bonita	•	•	
Camunda	•	•	
Oracle BPM	•		•
QualiBPMN	•		•
Signavio	•	•	
Visio	•		•

Source: the authors

Anti-pattern 8: Each lane in the pool contains start event

Adonis, *ARIS*, *Bonita*, *Oracle BPM*, *QualiBPMN* and *Visio*: allowed modeling. Enabled multiple start event without any type of feedback.

Bizagi, *Camunda*, and *Signavio*: allowed modeling. Provided messages that too many start events were used. For Signavio, this message is a warning, while the other consider it an error.

Blueworks: not possible to model. The tool already start the model with a start event and does not allow inserting a new one.

Table 5.8: Anti-pattern 8 - current problem detection and feedback. A marked cell identifies that the tool presented the behavior described on the column.

Modeling Tool	Allowed modeling	Raised warning	Raised error
Adonis	●		
ARIS	●		
Bizagi	●		●
Blueworks			
Bonita	●		
Camunda	●		●
Oracle BPM	●		
QualiBPMN	●		
Signavio	●	●	
Visio	●		

Source: the authors

Anti-pattern 9: Exception flow is not connected to the exception

Adonis and *Blueworks*: allowed modeling. The tools did not detect the missing exception flow as a problem, and did not provide feedback about it.

ARIS, *Bonita*, *Camunda*, *Oracle BPM*, *QualiBPMN*, *Signavio* and *Visio*: allowed modeling. All tools raised errors regarding the exception flow not connected to the event.

Bizagi: did not allow modeling. Bizagi automatically adds a sequence flow leaving the exception event. If an activity is not added to handle the exception, the tool triggers an error claiming that the flow is disconnected.

Table 5.9: Anti-pattern 9 - current problem detection and feedback. A marked cell identifies that the tool presented the behavior described on the column.

Modeling Tool	Allowed modeling	Raised warning	Raised error
Adonis	●		
ARIS	●		●
Bizagi			●
Blueworks	●		
Bonita	●		●
Camunda	●		●
Oracle BPM	●		●
QualiBPMN	●		●
Signavio	●		●
Visio	●		●

Source: the authors

Anti-pattern 10: Message flow used inside the pool

Adonis, *ARIS* and *Camunda*: did not allow modeling. The tools automatically change from a sequence to a message flow, without any feedback.

Bizagi: did not allow modeling. Similarly to other patterns, Bizagi leaves the flow only visually placed in the model, without it being connected to its destiny.

Blueworks and *Oracle BPM*: not possible to model. The tools only allow modeling one pool at a time, therefore is not possible to add message flows.

Bonita: not possible to model. The tools provides fields to fill the sequence flow source and destiny, and only allows selecting activities places in different pools.

QualiBPMN, *Signavio* and *Visio*: allowed modeling. Both provide messages about the wrong usage of the sequence flow. This message was a warning for Signavio and error for the other tools.

Table 5.10: Anti-pattern 10 - current problem detection and feedback. A marked cell identifies that the tool presented the behavior described on the column.

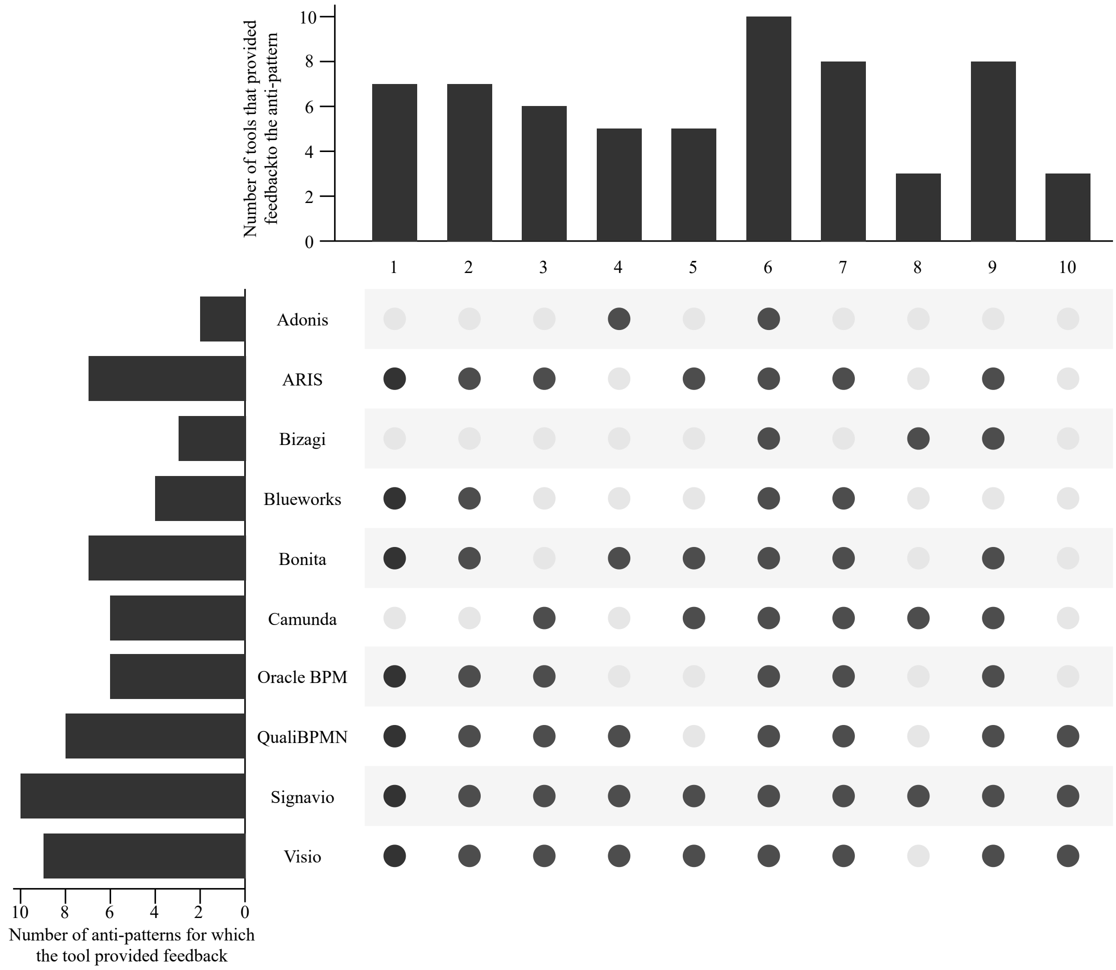
Modeling Tool	Allowed modeling	Raised warning	Raised error
Adonis			
ARIS			
Bizagi			
Blueworks			
Bonita			
Camunda			
Oracle BPM			
QualiBPMN	•		•
Signavio	•	•	
Visio	•		•

Source: the authors

5.2 Current state of feedback about problems in modeling tools

Figure 5.2 presents grouping the results for the ten patterns. A dark dot in this plot means the tool provided visual feedback about the problem. Signavio is the only tool that provided feedback for all anti-patterns, while Adonis only presented feedback for two out of ten. Anti-pattern 6 is the only pattern for which feedback was observed in all ten tools. In contrast, for pattern 10, feedback was perceived in only three tools, which were the only ones where the problem could be modeled.

Figure 5.2: Updated results for anti-pattern feedback in modeling tools. A marked dot means the tool provided feedback for the pattern, e.g., Adonis provided feedback for patterns 4 and 6, and feedback was only observed for anti-pattern 10 is on QualiBPMN, Signavio and Visio



Source: the authors

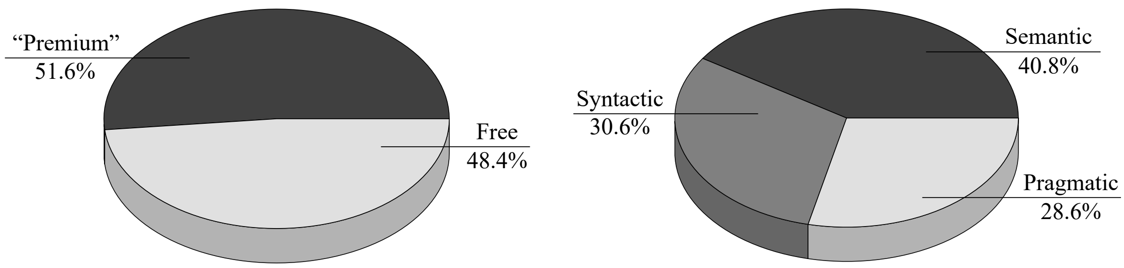
Out of the ten tools, five of them were ranked in the top ten business process tools: Visio, ARIS, Adonis, Signavio, and Blueworks (GARTNER, 2021). These tools are all proprietary and related to large companies (MICROSOFT, 2021; AG, 2021; BOC, 2021b; SIGNAVIO, 2021b; IBM, 2021), and we call them "premium" for simplification. The other five modeling tools all have at least one version permanently available (BIZAGI, 2021a; BONITASOFT, 2021a; CAMUNDA, 2021c; ORACLE, 2021a; FREEBPMN-QUALITY, 2021a). As presented in Figure 5.3.a, from the instances where visual feedback was observed, 51.6% were in the "premium" tools, while the remaining 48.4% were in free tools. Although results are slightly better for the top-ranked solutions, several tools from both groups were not consistent in detecting and providing feedback about problems. Therefore, the fact that a tool is paid and maintained by a large vendor does not appear to correlate directly with it providing better feedback about modeling problems.

When analyzing the anti-patterns by their problem classification, six out of the ten anti-patterns are syntax errors, while three are pragmatical and two are semantic issues, as pattern 1 is both syntactically and pragmatically incorrect (ROZMAN; POLANCIC; HORVAT, 2008). The anti-patterns detected with feedback by the ten modeling tools were semantic issues in 40.8% of the cases, and syntax and pragmatic problems are 30.6% and 28.6%, respectively, as illustrated in Figure 5.3.b. Therefore, although syntax problems are more commonly detected by tools (ROZMAN; POLANCIC; HORVAT, 2008), this tendency is not observed in the visual feedback offered by such tools.

Figure 5.3: Results of modeling anti-patterns in business process modeling tools.

(a) For the instances an anti-pattern was detected with feedback, the modeling tool was

(b) For the instances an anti-pattern was detected with feedback, its error type was



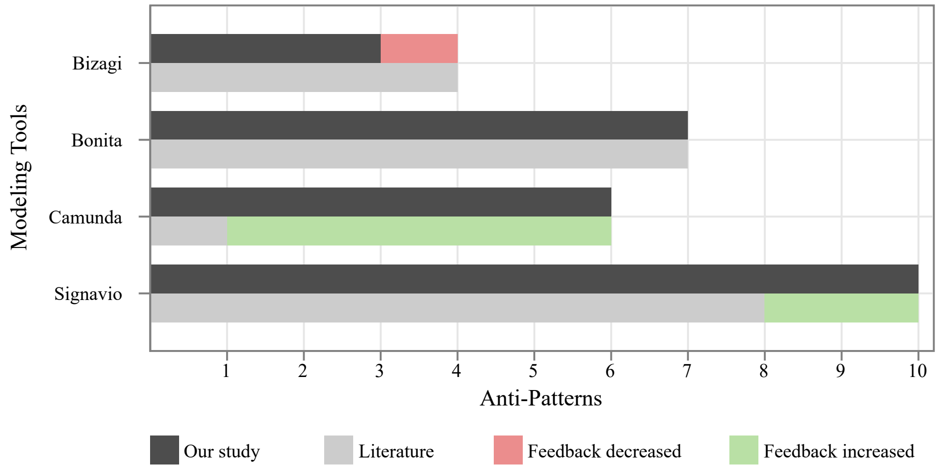
Source: the authors

Although better feedback results were observed for semantic issues, the validation of these problems can be complicated, even demanding a human expert who owns business knowledge (WEBER; HOFFMANN; MENDLING, 2008). Our models have no business context, being less challenging to detect for automated tools. Therefore, the results observed do not indicate that good semantic feedback was perceived, but instead that BPMN syntax validation is not necessarily trivial for modeling tools.

5.3 Comparison with previous results from the literature

Figure 5.4 compares our results for the four tools documented by the literature. The number of anti-patterns for which feedback was provided increased for Camunda and Signavio. No differences were found for Bonita as visual feedback was observed for the same patterns. Finally, the number of anti-patterns covered by Bizagi decreased. The tool currently prohibits modeling a problem that was possible to model and detected as incorrect. However, Bizagi does not provide any guidance on why modeling is not allowed.

Figure 5.4: Comparison of number of anti-pattern detected per modeling tool



Source: the authors, with literature results from (DIAS et al., 2019)

In the results documented in the literature, each modeling provided feedback for 5 anti-patterns, on average (DIAS et al., 2019). In our study, this average increased to 6.2. Therefore, although there is an improvement in the presented feedback, it is too discrete to confirm that the tools were updated with this purpose.

5.4 Final comments

In this chapter, we first presented the possible outcomes of designing an anti-pattern in a modeling tool. Then, we reviewed the experiments performed by (DIAS et al., 2019) and complemented them with six additional modeling tools (Adonis, ARIS, Blueworks, Oracle BPM, QualiBPMN and Visio). After our analysis, we have the updated state of problem feedback about anti-patterns in modeling tools, answering our RQ1, and achieving its related objectives.

This chapter also verified that well-ranked tools from large vendors are not necessarily better in detecting and providing feedback about problems. Then, we analyzed the provided feedback by problem type, which indicates syntax validation can still be challenging for modeling tools. Finally, a comparison with previous results presented a slight improvement in the average feedback about problems perceived in the modeling tools.

6 VISUAL FEEDBACK IMPLEMENTED BY MODELING TOOLS IN COMPARISON WITH THE LITERATURE

This chapter reviews and complements the analysis of how feedback is visually provided by modeling tools. First, we present examples of models with modeling problems based on anti-patterns. Then, we illustrate how the literature recommends that visual feedback should be presented for such models. Finally, we model them in the modeling tools and compare the results to the literature recommendations. With these steps, we intend to answer our RQ2 and fulfill the objectives related to this question.

6.1 Problematic models based on anti-patterns

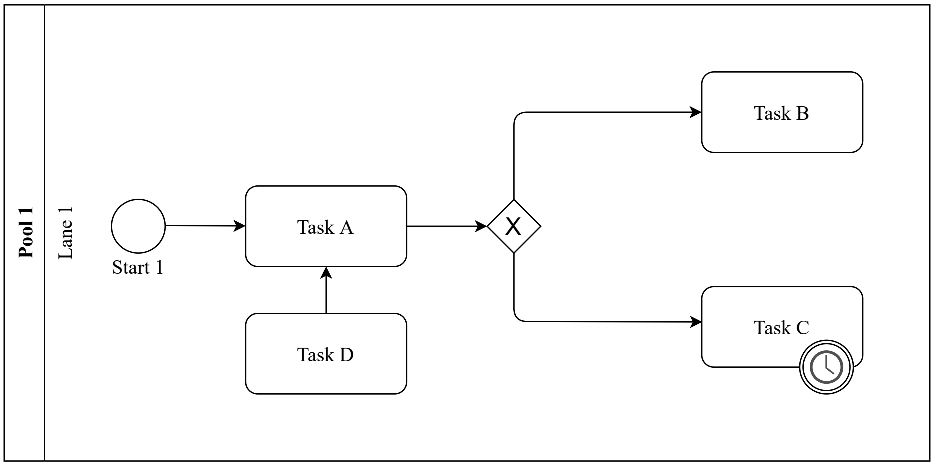
The results presented in the last chapter of how modeling tools react to problems were highly heterogeneous. No pair of modeling tools behaved equally for all anti-patterns. Additionally, there is no single pattern for which all tools were in both allowed to model and provided feedback. Therefore, is not possible to select a group of modeling anti-patterns patterns possible to model with feedback in all ten tools.

Consequently, when analyzing how tools behave visually while facing problems, we are not able to use the same group of patterns in all tools. Hence, we expose each modeling tool to a set of problems for which the tool is knowingly capable of detecting with feedback. An example is the subset composed by patterns 2, 7, and 9, used to evaluate visual feedback in ARIS, Bonita, Camunda, Oracle BPM, QualiBPMN and Visio.

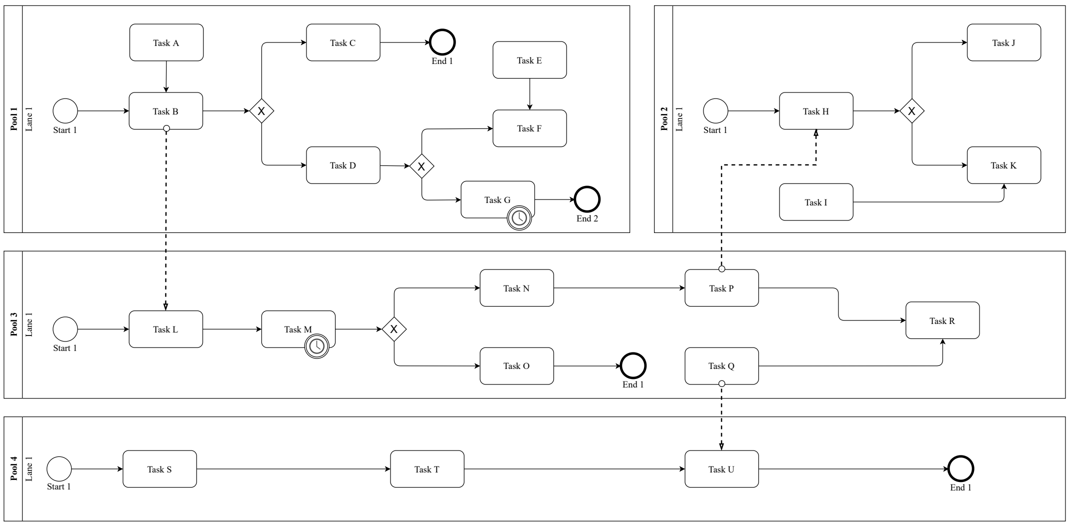
To help modelers on fixing the model, visual feedback about problems can be presented differently for small and large models (DANI; FREITAS; THOM, 2019b). Consequently, in each modeling tool, we design two models, one small and one large. Figure 6.1 display examples of small and large problematic models, compose of anti-patterns 2, 7 and 9. A model can be considered large when it cannot be adequately displayed in the modeling area of a process modeling tool (DANI; FREITAS; THOM, 2019b). Both large models proposed were tested in the tools analyzed in this study and were considered large due to a high impairment in readability of their elements. In addition, BPMN modeling guidelines a threshold of 50 modeling elements per model, as larger models are more likely to generate errors (MENDLING; REIJERS; AALST, 2010; MENDLING; NEUMANN; AALST, 2007). Since the concept of modeling elements comprehend tasks, flows, events, lanes, and any other visual element described in the notation (OMG, 2011), both of the proposed large models pass this threshold.

Figure 6.1: Example of small and large models with modeling problems. Both models contain anti-patterns 2, 7 and 9. These specific models were used to evaluate visual feedback about problems on six out of the ten tools.

(a) A small model with modeling problems



(b) A large model with modeling problems



Source: the authors

6.2 Visual feedback according to literature recommendations

After proposing and validating the process models, we can design how the literature recommends that feedback should be displayed for these problems. The recommendations documented in the literature were based on a SLR, modelers demands and what four commercial modeling tools offered and lacked in terms of visual feedback about problems (DANI; FREITAS; THOM, 2019b).

Applying recommendations for small process models

Table 6.1 presents the visual feedback recommendations about problems presented in the literature for small models (DIAS et al., 2019). An example of the outcome of applying such recommendations in our models is presented in Figure 6.2.

Table 6.1: Recommendations for visual feedback about problems on small business process models

Recommendations	
1	Highlight the problem with a visual element (e.g., an icon)
2	Highlight the problem through the coloring of the problematic element
3	Provide explanations when hovering the mouse on the problem
4	Provide a list of all the problems in the model
5	Link the problematic element with the respective entry of the problem list
6	Provide problem documentation or suggestions on how to fix it

Source: (DANI; FREITAS; THOM, 2019b)

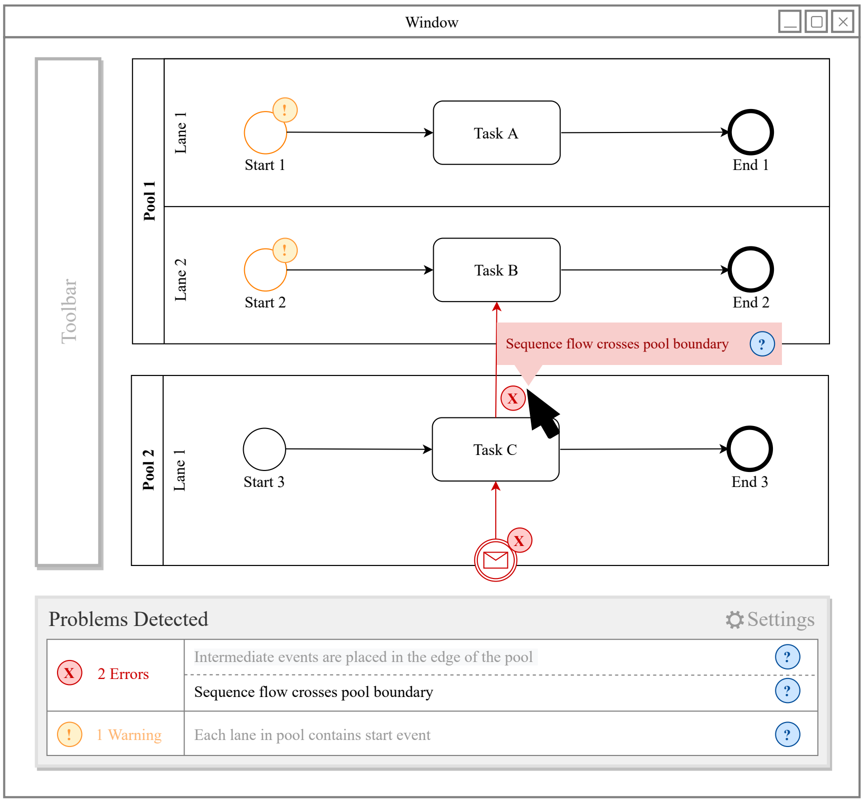
All problems in Figure 6.2 are highlighted with icons and colors, fulfilling recommendations 1 and 2. Per recommendation 3, a explanatory floating message is provided when hovering the mouse on the problem. While the list called *Problems Detected* (following recommendation 4) presents all problems, this one hovered problematic element is highlighted on the list, as described on recommendation 5. Finally, recommendation 6 is implemented by providing access to the problem documentation through the blue icon with an interrogation point in the pop-up and the overall list.

Applying recommendations for large process models

For large process models, two complementary visualizations are recommended. The first one provides a view of the entire model, grouping problems into problematic areas. The second is shown when the user zooms the screen in a problematic area. In the latter, details for each problem is provided through the same visual indicators used for small models (e.g., icons and coloring) (DANI; FREITAS; THOM, 2019b).

Recommendations for large models are presented in Table 6.2 (DANI; FREITAS; THOM, 2019b). An example of applying these recommendation in a model with anti-patterns is presented in Figure 6.3.

Figure 6.2: Recommended feedback applied in a small process model. Here, problems are highlighted through icons and colors. When hovering the mouse over the problematic element, an explanation is given, highlighting this issue in the list of *Problems Detected*, while still listing the others.



Source: the authors following recommendations of (DANI; FREITAS; THOM, 2019b)

Table 6.2: Recommendations for visual feedback about problems on large business process models

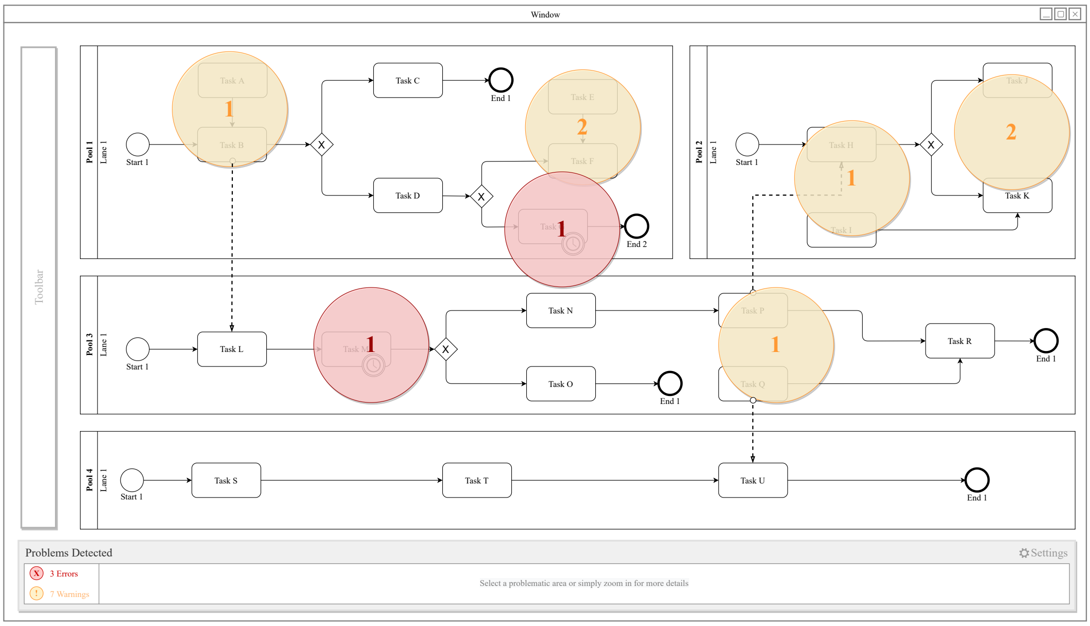
Recommendations	
1	When viewing the entire model, without zoom, problems should be grouped into problematic areas, according to their type (error or warning)
2	When zooming in a problematic area, the modeling tool should focus on the problems included in the area, providing details about them
3	When zooming in a problematic area, the problem list should contain the problems included in the area
4	When zooming in a problematic area, the modeling tool should visually show where the other problems are in the model

Source: (DANI; FREITAS; THOM, 2019b)

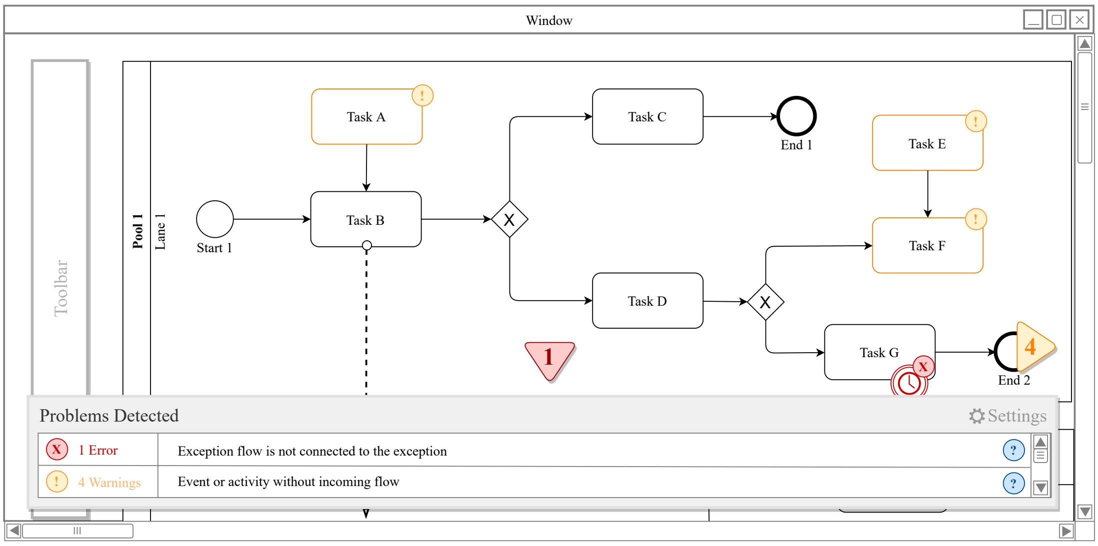
When displaying the entire model, as in Figure 6.3.a, recommendation 1 is implemented through colored circles grouping problems by their type (red for errors, yellow

Figure 6.3: Recommended feedback applied in a large model when (a) viewing the entire model and (b) zooming into a part of the model.

(a) Recommended feedback when viewing an entire large model. Problematic areas are presented through circles in red for errors and in yellow for warning. The *Problems Detected* list displays the total number of problems in the models, but does not provide details about specific issues.



(b) Recommended feedback when zooming into a problematic area in a large model. Details for specific problems are presented through icons, colors, and explanations. Colored arrows provide navigation to other problems in the models.



Source: the authors following recommendations of (DANI; FREITAS; THOM, 2019b)

for warnings). The zoomed visualizations, portrayed in Figure Figure 6.3.b, follow the other three recommendations by providing details about the problems in the area in both the model and the list while indicating where other problems are through colored arrows, following the same color-coding. These arrows contain a number, informing how many problems of that type can be found in the pointed directions, and allow the modeler to decide which problem area they want to attack first (DANI; FREITAS; THOM, 2019b).

6.3 Visual feedback observed in modeling tools

After designing the visual feedback recommended by the literature, we now review and complement the analysis of how modeling tools behave visually when reacting to problems. Finally, we compare the tools with the literature (DANI; FREITAS; THOM, 2019b) to understand how similar the two are.

Visual feedback about problems observed in modeling tools for small models

Adonis: when problems are detected, Adonis use icons to indicate where the problems are located. Errors and warnings are both represented by the symbol "!" painted as red and orange, respectively. The tool provides a list with all the problems, and for which one, a suggestion of how to fix it is available. When clicking on a listed problem, the problematic element is highlighted with coloring. However, no explanation is provided when hovering the mouse over a problem.

ARIS: in some instances, the tool blocked incorrect modeling preemptively by showing a grey icon with a forbidden icon. When problems are indeed modeled, colored icons (red, orange and yellow for errors, warnings and notes) are place in the issued elements. An explanation is presented when the mouse is passed over a problem, but no suggestions or access to documentation is available. Also, ARIS does not provide any list of the problems and does not color problematic elements.

Bizagi: the tool provides a list displaying the problems in the model. When selecting a problem in the list, Bizagi highlights the issue in the model with a box. No other visual helpers about problems were observed.

Blueworks: in general, Blueworks disables any action that will lead the model to a state considered as incorrect by the tool. Feedback about problems are only perceived when trying to perform such actions, through a floating explanatory message.

Bonita: when it detects modeling problems, Bonita places a red icon in the incorrect elements, and when hovering the mouse on it, problem descriptions are presented. Additionally, the tool provides a list with all the problems in the model, and when clicking on an item listed, the issue is accentuated in the model. Coloring elements is not used by Bonita, which also does not provide access to problem documentation.

Camunda: the tool highlights problems in the model with colored icons. A red "X" is used for errors, and a yellow "!" for warnings. A brief explanation about the issue is provided when passing the cursor over it. When blocking an incorrect operation (e.g. using a sequence flow to connect elements in different pools), Camunda colors the troubled elements in red. No other visual indicator about problems were observed in the tool.

Oracle BPM: the tool uses red and yellow icons to indicate where errors and warnings, respectively, are located. When hovering the mouse over such icons, problem description is presented using a floating message. The only other visual feedback feature provided by Oracle BPM is a list with all the problems in the model.

QualiBPMN: the used version of QualiBPMN highlights problematic elements with red and warning icons. When passing the mouse on the issues, the tool presents a floating description about it. A list of all the problems is available, with suggestions on how to fix the issues. No other visual indicators are presented.

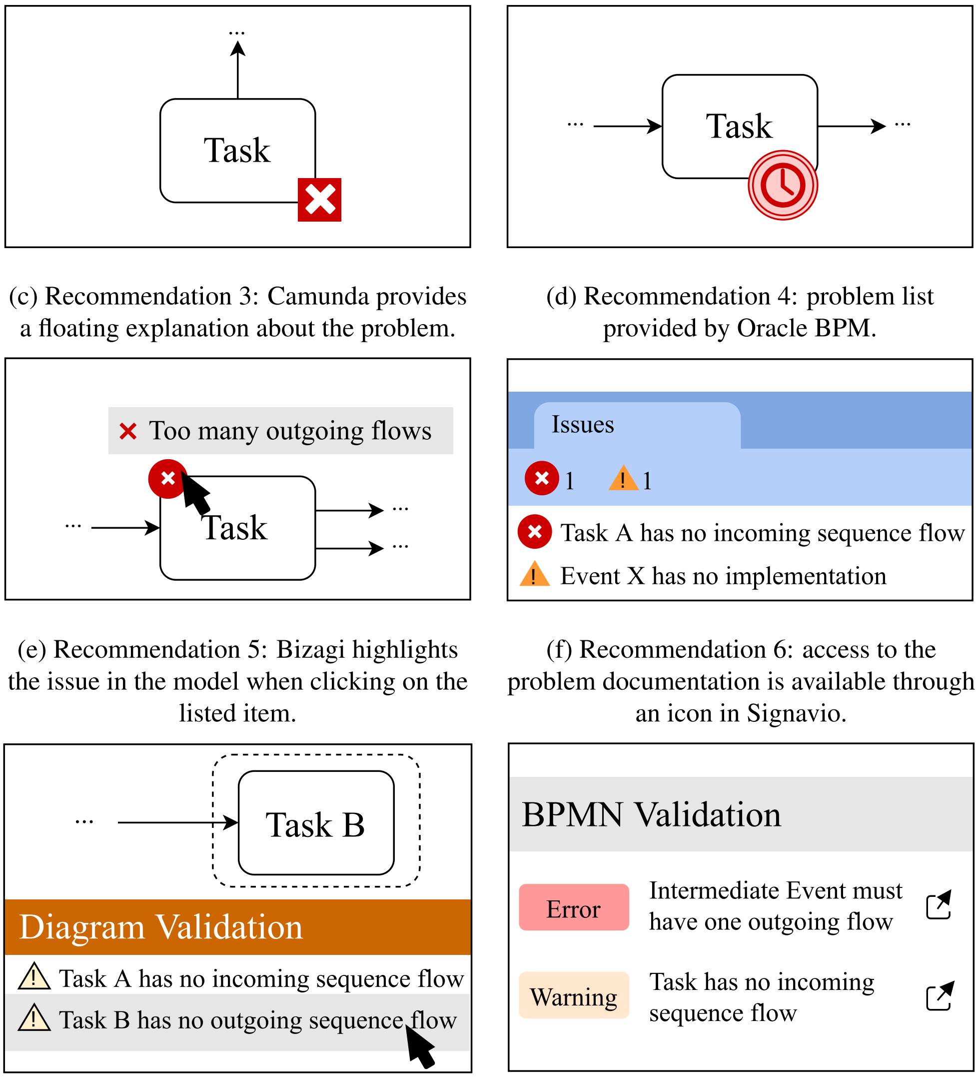
Signavio: in the tool, problems are highlighted by an icon with symbol "!", painted in red and yellow for errors and warnings. An explanation about the issue is presented when passing the mouse on it. A list with all the problems is present, and through it, the user can access documentation about each problem. Furthermore, when clicking on an item on the list, the respective problem is highlighted in the model with a box. Coloring is the only visual recommendation not observed in Signavio.

Visio: the tool does not highlight elements with problems, therefore icons, coloring and floating explanations are not applicable. The tool provides a list with descriptions for all the problems, but access to documentation is not available. No other visual feedback techniques were observed while analyzing Visio.

Figure 6.4 present instances of the six recommendations about feedback in small models found in the modeling tools. For example, Figure 6.4.e illustrate how Bizagi highlights Task B, which has no incoming sequence flow, when the correspondent problem description is selected in the issue list. We make available the experiment data[1] for future comparative research. For every tool, screenshots illustrate the observed visual elements used for problem feedback.

[1]Notes for our experiments are available at <https://camargodev.github.io/bpmn-problem-analysis/>

Figure 6.4: Examples of visual feedback about problems presented by modeling tools in accordance to the literature



(a) Recommendation 1: ARIS places a square red icon on hanging task.

(b) Recommendation 2: Adonis coloring a problematic intermediate event.

(c) Recommendation 3: Camunda provides a floating explanation about the problem.

(d) Recommendation 4: problem list provided by Oracle BPM.

(e) Recommendation 5: Bizagi highlights the issue in the model when clicking on the listed item.

(f) Recommendation 6: access to the problem documentation is available through an icon in Signavio.

Source: the authors

Visual feedback about problems observed in modeling tools for large models

None of the ten evaluated modeling tools implement any of the four recommendations for visual feedback about problems in large models. All tools behaved equally to what they did when facing problems in a small model.

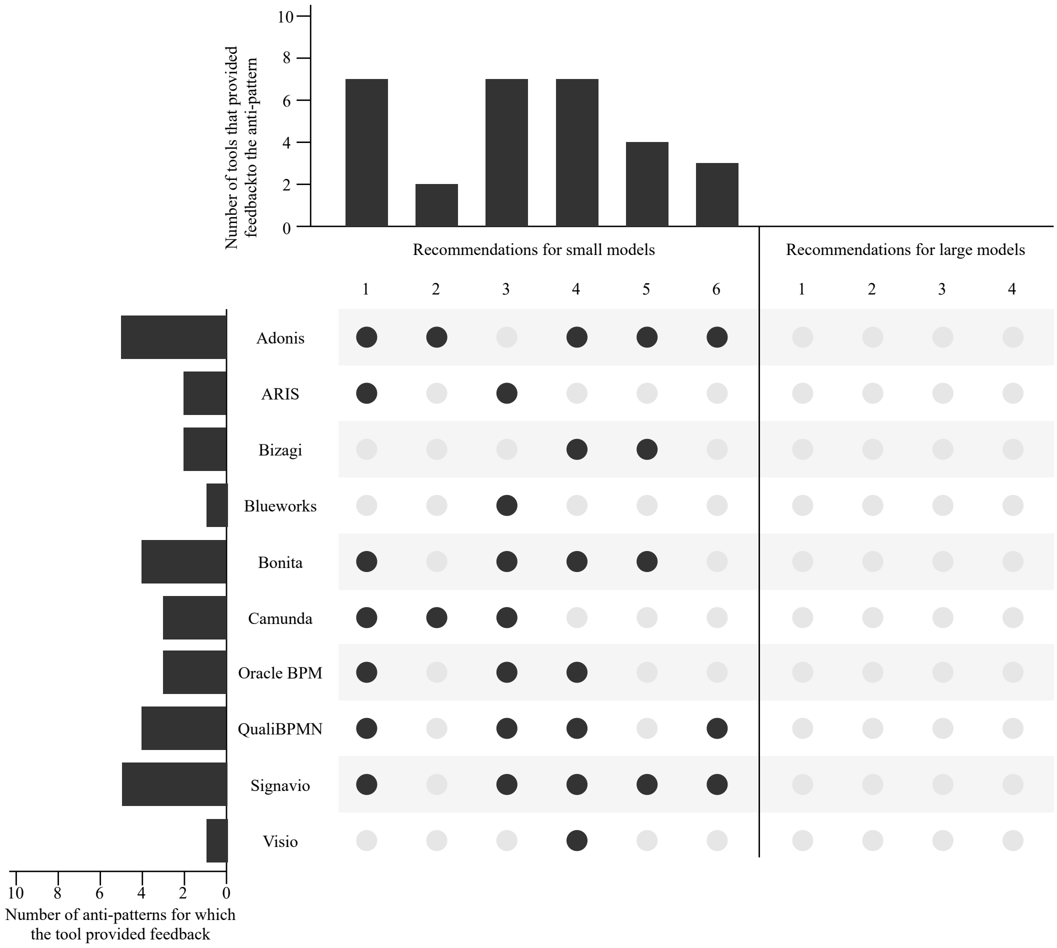
When viewing the entire model without zoom, details are still presented about each error individually. The lack of the problematic areas, as suggested by recommendation 1, leads to one visualization with several errors and warning symbols, which difficult understanding the problems.

No visual feedback changes are observed when the zoom is applied to an area in the model. In addition, there is no form of highlight for the problems in the area, and the list of detected problems does not focus on the issues on screen, not following recommendations 2 and 3. Finally, no navigation helpers, defined by recommendation 4, were provided to indicate where other problems are located in the model.

6.4 Comparison of the literature and the visual feedback provided by tools

As presented by Figure 6.5, Adonis and Signavio both implement five out of the ten recommendations. Among the ten tools, the visual feedback behavior observed on these two is the most similar to what the literature recommends. In average, each modeling tool follows three out of ten recommendations on visual feedback about problems.

Figure 6.5: Visual feedback about problems recommendations implemented by each modeling tool. A dark dot identifies that the recommendation was implemented by the tool, e.g., recommendation 2 is present in Adonis and Camunda.



Source: the authors

Recommendations 1, 3 and 4 - highlighting issues with icons, displaying floating problem messages and providing a problem list - are the most recurrent among tools, being observed in seven out of ten each. For small models, recommendation 2, coloring the problematic element, was only found in Adonis and Camunda, being the less common. However, no tool implements any of the large visualization recommendations, which implies in the major difference between what tools provide to what the literature recommends. Furthermore, apart from the ones covered by the recommendations, no other distinct visual feedback feature was offered by the tools. These findings answer our RQ2.

6.5 Final comments

In this chapter, we understood how the literature (DANI; FREITAS; THOM, 2019b) recommends that visual feedback about problems should be presented by modeling tools. Then, we exposed each of these tools to incorrect process models, and analyzed their visual reaction. Finally, a comparison between tools and literature is performed to understand how similar are the two.

7 CONCLUSION

Incorrect business process models result in unexpected outcomes in the process execution. Since process models are designed with process modeling tools, such tools should provide feedback about problems to the modelers. Therefore, this study presented the current state of visual feedback about problems presented by business process modeling tools.

Using process modeling anti-patterns learned from the literature, we reviewed and complemented the experiments of Dias et al. (DIAS et al., 2019), verifying that there is still no common ground on how process modeling tools react to modeling problems. Out of ten modeling tools, all behaved differently when facing problems, as no pair of the ten studied tools provided problem feedback about the same set of issues. Additionally, only one out of ten modeling problems was detected by all tools, which still reacted differently when exposed to it.

Our analysis also shows no significant differences in problem detection and feedback are observed when comparing “premium” (i.e., top-ranked paid tools maintained by large vendors) and free tools. Furthermore, although the literature indicates that syntax validation on business process models is less challenging for automated process modeling tools, our results indicate that detecting this class of problems is still not trivial for these tools.

When analyzing the visual aspects of feedback about problems presented by process modeling tools, we verified that there is still a distance between what the literature recommends and what modeling tools actually present to their users. By using recommendations on visual feedback about problems proposed by Dani et al. (DANI; FREITAS; THOM, 2019b), we identified that process modeling tools are inconsistent while implementing recommendations for small models and do not use any of the suggested visual helpers for visualization of problems in large business processes models.

Finally, we believe that the presented complementary analysis can be useful to help both process modelers and process modeling tools developers to understand a bit more about problems on these models and how modeling tools currently detect and visually respond to them.

7.1 Limitations

A possible limitation of our study is the unavailability of some business process modeling tools, such as iGrafx and Mavim. No free or trial versions of these two top-ranked commercial modeling tools were found, limiting our analysis of how they react to problems and provide visual feedback. Additionally, short trial periods, such as 30 days for Blueworks and Visio, difficult recreating and revalidating the experiments.

7.2 Future Works

Future work is suggested on understanding why business process modeling tools lack on providing specialized feedback about large models. A reason could be that larger models are considered a modeling *smell* by the literature, and keeping models short is considered a process modeling guideline. However, none of the ten studied process modeling tools provided any indication that the model should be decomposed.

Another possible complementary study would be to build a process modeling prototype, which follows the visual feedback about problems recommendations of Dani et al. (DANI; FREITAS; THOM, 2019b). With this prototype, a survey could be performed to evaluate user satisfaction with the presented visual feedback elements.

REFERENCES

AG, S. **About Software AG**. 2021. (Accessed: 20.10.2021). Available from Internet: <https://www.softwareag.com/en_corporate/company.html>.

BARJIS, J. The importance of business process modeling in software systems design. **Science of Computer Programming**, Elsevier, v. 71, n. 1, p. 73–87, 2008.

BASTIAN, E. **ARIS Editions – What's it all about!** 2021. (Accessed: 20.10.2021). Available from Internet: <https://www.ariscommunity.com/users/elba/2021-07-08-aris-editions-\%E2\%80\%93-what\%E2\%80\%99s-it-all-about>.

BIZAGI. **About Bizagi**. 2021. (Accessed: 07.09.2021). Available from Internet: <https://www.bizagi.com/en/about>.

BIZAGI. **Bizagi: Platform Overview**. 2021. (Accessed: 07.09.2021). Available from Internet: <https://www.bizagi.com/en/platform>.

BOC. **Adonis: Comparison of Editions**. 2021. (Accessed: 07.09.2021). Available from Internet: <https://www.boc-group.com/en/adonis/comparison-of-editions/>.

BOC. **BOC Group**. 2021. (Accessed: 07.09.2021). Available from Internet: <https://boc-group.com/boc-group/>.

BOCK, J. D.; CLAES, J. The origin and evolution of syntax errors in simple sequence flow models in bpmn. In: SPRINGER. **International Conference on Advanced Information Systems Engineering**. [S.l.], 2018. p. 155–166.

BONITASOFT. **Bonita: Digital process automation for a competitive edge**. 2021. (Accessed: 07.09.2021). Available from Internet: <https://www.bonitasoft.com/bonita-platform>.

BONITASOFT. **REST API Overview with a Getting started example**. 2021. (Accessed: 09.09.2021). Available from Internet: <https://documentation.bonitasoft.com/bonita/2021.1/rest-api-overview>.

BONITASOFT. **What is Bonita?** 2021. (Accessed: 09.09.2021). Available from Internet: <https://documentation.bonitasoft.com/bonita/2021.1/what-is-bonita>.

CAMUNDA. **About bpmn.io**. 2021. (Accessed: 09.09.2021). Available from Internet: <https://bpmn.io/about/>.

CAMUNDA. **Camunda Modeler**. 2021. (Accessed: 09.09.2021). Available from Internet: <https://camunda.com/products/camunda-platform/modeler/>.

CAMUNDA. **Camunda Platform**. 2021. (Accessed: 09.09.2021). Available from Internet: <https://camunda.com/products/camunda-platform/>.

CHINOSI, M.; TROMBETTA, A. Modeling and validating bpmn diagrams. In: IEEE. **2009 IEEE Conference on Commerce and Enterprise Computing**. [S.l.], 2009. p. 353–360.

CHINOSI, M.; TROMBETTA, A. Bpmn: An introduction to the standard. **Computer Standards & Interfaces**, Elsevier, v. 34, n. 1, p. 124–134, 2012.

CLAES, J.; VANDECAVEYE, G. The impact of confusion on syntax errors in simple sequence flow models in bpmn. In: SPRINGER. **International Conference on Advanced Information Systems Engineering**. [S.l.], 2019. p. 5–16.

COMMUNITY, A. **Welcome to ARIS Community!** 2021. (Accessed: 20.10.2021). Available from Internet: <https://www.ariscommunity.com/about-us>.

DANI, V. S.; FREITAS, C. M. D. S.; THOM, L. H. Ten years of visualization of business process models: a systematic literature review. **Computer Standards & Interfaces**, Elsevier, v. 66, p. 103347, 2019.

DANI, V. S.; FREITAS, C. M. D. S.; THOM, L. H. Visual feedback about problems in business process models: systematic literature review, survey, case studies and recommendations. 2019.

DIAS, C. L. de B. et al. Anti-patterns for process modeling problems: An analysis of bpmn 2.0-based tools behavior. In: SPRINGER. **International Conference on Business Process Management**. [S.l.], 2019. p. 745–757.

DIJKMAN, R. M.; DUMAS, M.; OUYANG, C. Semantics and analysis of business process models in bpmn. **Information and Software technology**, Elsevier, v. 50, n. 12, p. 1281–1294, 2008.

DONGEN, B. F. van; AALST, W. M. Van der; VERBEEK, H. M. Verification of epcs: Using reduction rules and petri nets. In: SPRINGER. **International Conference on Advanced Information Systems Engineering**. [S.l.], 2005. p. 372–386.

DUMAS, M. et al. **Fundamentals of business process management**. [S.l.]: Springer, 2013.

FELLMANN, M. et al. Business process model patterns: state-of-the-art, research classification and taxonomy. **Business Process Management Journal**, Emerald Publishing Limited, 2019.

FREEBPMNQUALITY. **Free Online BPMN Quality Analysis Service**. 2021. (Accessed: 11.10.2021). Available from Internet: <https://freebpmnquality.github.io/>.

FREEBPMNQUALITY. **QualiBPMN - Free Online Quality Analysis Tool for BPMN Models**. 2021. (Accessed: 11.10.2021). Available from Internet: <https://cloudfreebpmnquality.herokuapp.com/analytics/>.

GARTNER. **Enterprise Business Process Analysis (EBPA) Reviews and Ratings**. 2021. (Accessed: 20.10.2021). Available from Internet: <https://www.gartner.com/reviews/market/enterprise-business-process-analysis>.

GEIGER, M. et al. Bpmn 2.0: The state of support and implementation. **Future Generation Computer Systems**, Elsevier, v. 80, p. 250–262, 2018.

HAISJACKL, C. et al. How do humans inspect bpmn models: an exploratory study. **Software & Systems Modeling**, Springer, v. 17, n. 2, p. 655–673, 2018.

HIPP, M. et al. Enabling a user-friendly visualization of business process models. In: SPRINGER. **International Conference on Business Process Management**. [S.l.], 2014. p. 395–407.

IBM. **IBM Blueworks Live**. 2021. (Accessed: 20.10.2021). Available from Internet: <https://www.ibm.com/products/blueworkslive>.

KOEHLER, J.; VANHATALO, J. Process anti-patterns: How to avoid the common traps of business process modeling. **IBM WebSphere Developer Technical Journal**, v. 10, n. 2, p. 4, 2007.

KOENIG, A. Patterns and antipatterns. **The patterns handbook: techniques, strategies, and applications**, v. 13, p. 383, 1998.

KOSCHMIDER, A.; LAUE, R.; FELLMANN, M. Business process model anti-patterns: a bibliography and taxonomy of published work. 2019.

KROGSTIE, J.; SINDRE, G.; JØRGENSEN, H. Process models representing knowledge for action: a revised quality framework. **European Journal of Information Systems**, Springer, v. 15, n. 1, p. 91–102, 2006.

LEHMANN, B.-D. et al. Towards the identification of process anti-patterns in enterprise architecture models. In: **QuASoQ@ APSEC**. [S.l.: s.n.], 2020. p. 47–54.

MANCARELLA, S. Business process modelling notation–a tutorial. **OMG SOA Healthcare. SPARX Systems**, 2011.

MENDLING, J.; NEUMANN, G.; AALST, W. V. D. Understanding the occurrence of errors in process models based on metrics. In: SPRINGER. **OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"**. [S.l.], 2007. p. 113–130.

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

MICROSOFT. **Visio**. 2021. (Accessed: 20.10.2021). Available from Internet: <https://www.microsoft.com/pt-br/microsoft-365/visio/flowchart-software>.

MILI, H. et al. Business process modeling languages: Sorting through the alphabet soup. **ACM Computing Surveys (CSUR)**, ACM New York, NY, USA, v. 43, n. 1, p. 1–56, 2010.

OMG. **Business Process Model and Notation (BPMN), Version 2.0**. 2011. (Accessed: 20.09.2021). Available from Internet: <https://www.omg.org/spec/BPMN/2.0/PDF>.

OMG. **BPMN in Action!** 2020. (Accessed: 07.09.2021). Available from Internet: <https://www.omg.org/events/2020Q2/special-events/BPMN-in-Action.htm>.

ORACLE. **Introduction to Oracle BPM Studio**. 2021. (Accessed: 20.10.2021). Available from Internet: <https://docs.oracle.com/cd/E21764_01/doc.1111/e15176/intro_bpm_studio.htm#BPMPD143>.

ORACLE. **Oracle BPM**. 2021. (Accessed: 20.10.2021). Available from Internet: <https://www.oracle.com/middleware/technologies/bpm.html>.

ORACLE. **Productive Java-based Application Development**. 2021. (Accessed: 20.10.2021). Available from Internet: <https://www.oracle.com/application-development/technologies/jdeveloper.html>.

ROZMAN, T.; POLANCIC, G.; HORVAT, R. V. Analysis of most common process modeling mistakes in bpmn process models. **Eur SPI'2007**, 2008.

SIGNAVIO. **BPM Academic Initiative**. 2021. (Accessed: 09.09.2021). Available from Internet: <https://www.signavio.com/bpm-academic-initiative/>.

SIGNAVIO. **SAP to Acquire Business Process Intelligence Company Signavio**. 2021. (Accessed: 09.09.2021). Available from Internet: <https://www.signavio.com/news/signavio-joins-sap/>.

SIGNAVIO. **Signavio Business Transformation Suite**. 2021. (Accessed: 09.09.2021). Available from Internet: <https://www.signavio.com/products/business-transformation-suite/>.

SIGNAVIO. **The Story Behind Signavio**. 2021. (Accessed: 09.09.2021). Available from Internet: <https://www.signavio.com/story-behind-signavio/>.

WEBER, B. et al. Refactoring large process model repositories. **Computers in industry**, Elsevier, v. 62, n. 5, p. 467–486, 2011.

WEBER, I.; HOFFMANN, J.; MENDLING, J. Semantic business process validation. In: CITESEER. **Proceedings of the 3rd International Workshop on Semantic Business Process Management (SBPM'08), CEUR-WS Proceedings**. [S.l.], 2008. v. 472.

WESKE, M. **Business process management architectures**. [S.l.]: Springer, 2007.