

**FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO
PORTO**

Change Impact Analysis Based on Business Process Model

Acilio António Correia Silva



Mestrado em Engenharia de Software

Supervisor: Nuno Flores

Second Supervisor: Ana Paiva

October 13, 2020

Change Impact Analysis Based on Business Process Model

Acilio António Correia Silva

Mestrado em Engenharia de Software

Approved in oral examination by the committee:

Chair: Prof. João Carlos Pascoal Faria

External Examiner: Prof. Alberto Silva

Supervisor: Prof. Nuno Flores

October 13, 2020

Abstract

This report is part of the dissertation in Software Engineering course to obtaining a master degree. It describes the work performed to solve the problem presented on the theme "Change Impact Analysis based on Business Process Model." The impact analysis describes the process that identifies the consequences and side effects that a given change may produce. Following a manual approach to perform the impact analysis turns the process slow, and in most cases, fails to cover all the amplitude of a project. Software engineering helps to change this paradigm, allowing automatic approaches to simplify the process.

Our goal is to create a tool that allows the comparison between two different business process model file, presenting, in the end, all the changed elements and impacted activities if the tool identifies a change in any elements on each model. To do so, the user has to upload the BPMN files that he/she wants to analyze. The tool uses a change patterns catalog to apply the searching process over the BPMN files looking for change patterns. After this search, the user is redirected to the result page, containing the analysis's result information. For the implemented change patterns included on the tool, we have inserted, removed, and updated activity change patterns, inserted and removed gateways and artifacts change patterns and, control-flow dependencies difference change patterns.

Future work would involve further improvements of the solution at the level of creating new features like user profile, BPMN management, and implementation of the following change patterns: Refined activities and corresponded collections of activities change patterns; Authorization differences change patterns; Additional dependencies, activities occur at different moments in processes, and the iterative vs. once-off occurrence change patterns.

Keywords: Change Impact Analysis, Business Process Model, BPMN

Resumo

Este relatório faz parte da dissertação do curso de Engenharia de Software para a obtenção do grau de mestre. Descreve o trabalho realizado para solucionar o problema apresentado através do tema "Análise de Impacto baseada em Modelo de Processos de Negócio". A análise de impacto descreve o processo que identifica as consequências e efeitos que uma determinada mudança possa produzir. Seguir uma abordagem manual na realização deste tipo de análise torna o processo lento e, na maioria dos casos, não cobre a totalidade e a amplitude do projeto. A engenharia de software ajuda a mudar esse paradigma, permitindo assim a implementação de abordagens automáticas de modo a simplificar tal processo.

O objetivo principal consiste em criar uma ferramenta que permita a análise comparativa entre dois arquivos de modelo de processo de negócio diferentes, apresentando, no final, todos os elementos alterados e quais as atividades impactadas com as alterações identificadas pela ferramenta. Para isso, o utilizador efetua o upload dos ficheiros BPMN que deseja analisar. A ferramenta usa um conjunto de padrões das alterações previamente implementadas em catálogo aplica-lhe no processo de pesquisa pelas alterações nos arquivos BPMN. Após esta busca, o utilizador é redirecionado para a página dos resultados, contendo as informações do resultado da análise. Dos padrões de alterações implementados na ferramenta destacamos: inserir, remover e atualizar atividades, inserir e remover gateways e artefatos e padrão de identificação das diferenças de fluxos e dependências.

Como trabalho futuro envolveria melhorias adicionais da solução a nível de criação de novos recursos como perfis de utilizador, gestão das BPMN e implementação dos seguintes padrões das alterações: Atividades refinadas e coleções correspondentes, diferenças de autorização, atividades ocorrem em diferentes momentos no processo e ocorrência iterativa vs. única.

Keywords: Análise de Impacto, Modelo de Processos de Negócio, BPMN

Acknowledgements

I want to express my sincere thanks to everyone who has helped and supported me while working on this thesis. I want to thank you professor Ana Paiva and Nuno Flores for taking the time to advise and help me write this thesis and always show the availability to clear any doubts that I have. I thoroughly enjoyed our discussions, and I firmly believe that the feedback generated from them contributed drastically to the work's accomplishment.

To all my colleagues from MESW, who have put up with me for the past two years, that have taught me so many things, and that has made my time in FEUP unforgettable for all the best reasons, an enormous thank you. Lastly, I want to express my most sincere thanks to my parents, who have always supported and believed in me, and without whom I would not be the person that I am today.

Acilio Silva

*“Knowledge is not what you know,
but what you do with what you know.”*

Aldous Huxley

Contents

1	Introduction	1
1.1	Context	1
1.2	Motivation	2
1.3	Problem	2
1.4	Thesis Structure	3
2	Background	4
2.1	BPM - Business Process Model	5
2.2	BPMN - Business Process Model and Notation	5
2.3	Change Impact Analysis	7
2.4	Model-Based CIA Technique	8
2.5	Pattern-Driven Impact Analysis	8
2.6	Key Challenges of CIA	9
2.7	Summary	10
3	Change Pattern Catalog	11
3.1	Artifact Differences	11
3.2	Gateway Differences	12
3.3	Authorization Differences	13
3.4	Activity Differences	14
3.5	Control-flow Differences	16
3.6	Summary	17
4	Methodology	19
4.1	Introduction	19
4.2	Approach and Model	19
4.3	Impact Analysis in N'steps	21
4.4	Summary	23
5	Change Impact Analysis Tool	24
5.1	Purpose	24
5.2	Architecture	24
5.3	Dependencies	25
5.4	User Interface	26
5.5	Technologies	28
5.5.1	Presentation Layer	29

5.5.2	Business Layer	29
5.5.3	Build, Deploys and Source Control	29
5.6	Summary	30
6	Validation	31
6.1	Example 1 - Hardware Retailer	31
6.1.1	Insert, Update and Remove Activities	32
6.1.2	Result	33
6.2	Example 2 - Incident Management	34
6.2.1	Insert and Remove Gateways	34
6.2.2	Result	35
6.3	Example 3 - Email Voting	37
6.3.1	Inserted and removed artifacts	39
6.3.2	Result	39
6.4	Summary	40
7	Conclusions and Future Work	41
7.1	Final Considerations	41
7.2	Difficulties	42
7.3	Further Work	42
	References	44
A	BPMN Model Code Snippets	46
A.1	Hardware retail example - First Model	46
A.2	Hardware retail example - Update Model	50

List of Figures

2.1	Literature review topics	4
2.2	Basic BPMN modeling elements. From [13]	7
3.1	An example of removed artifact change pattern	11
3.2	An example of inserted artifact change pattern	12
3.3	An example of an inserted gateway change pattern	12
3.4	An example of removed gateway change pattern	13
3.5	An example of different role authorization pattern	13
3.6	An example of a single role vs. collections of roles authorization pattern	13
3.7	An example of different collections of roles authorization pattern	14
3.8	An example of removed activity change pattern	14
3.9	An example of updated activity	15
3.10	An example of refined activity change pattern	15
3.11	An example of inserted activity change pattern	16
3.12	An example of different dependency change pattern	16
3.13	An example of additional dependency change pattern	17
3.14	An example of activity occur at different moments in processes change pattern	17
3.15	An example of iterative vs. once-off occurrence change pattern	18
4.1	Change Impact Analysis Model	20
5.1	Tool Architecture	25
5.2	First Page of the application	26
5.3	Impact Analysis page with model files uploaded 27	
5.4	Change Impact Analysis Result - Impacted Activies	28
5.5	Change Impact Analysis Result - Graphical comparison	28
6.1	Shipment process of a hardware retailer.	32
6.2	Shipment process of a hardware retailer updated version.	32
6.3	Shipment process of a hardware retailer changed elements.	33
6.4	Shipment process of a hardware retailer impacted activities.	33
6.5	Incident management business process	35
6.6	Incident management business process updated model	36
6.7	Impacted activities CIA result	36
6.8	Impact analysis result statistic	37

6.9	Email voting business process	38
6.10	Email voting business process updated model	39
6.11	Email voting changed elements	40
6.12	Email voting impacted activities	40

Listings

A.1	Hardware retail example - First Model	46
A.2	Hardware retail example - Update Model	50

Abbreviations

API	Application Programming Interface
BP	Business Process
BPM	Business Process Model
BPMN	Business Process Modeling and Notation
BPEL4WS	Business Process Execution Language for Web Services
CIA	Change Impact Analysis
CSS	Cascading Style Sheets
EJB	Enterprise JavaBean
HTML	Hypertext Markup Language
IA	Impact Analysis
JSF	JavaServer Faces
PaaS	Platform as a Service
UI	User Interface
UML	Unified Model Language
XML	Extensible Markup Language
XPDL	XML Process Definition Language

Chapter 1

Introduction

This dissertation was developed for the Master in Software Engineering in Faculty of Engineering, University of Porto. This chapter introduces the work conducted during this thesis's writing, outlining the context, motivation, problem, and the structure of this document.

1.1 Context

Organizations are constantly changing their (business goals, strategies, process, structure) and continuously adapting the core business processes to continue on the market. Business processes are made of activities, tasks, and events performed to satisfy business goals or services. In order to visualize the execution of a business process, organizations can make use of models diagram to illustrate the execution of each activity and task. The use of models will allow organizations to create graphically all the sequences steps taken during a business process. If this sequence steps change, we create a new model by updating the previous one to reflect the change.

The change introduced into the business process's sequence steps will create an impact on the already existing activities and tasks. The impact of a change is hard to predict. A Change impact analysis (CIA) should be done to ensure that side effects and potential change consequence are identified earlier, before its implementation [3]. Using the CIA helps to get in advance an abstract view of the side effects a change can produce on a business process [2].

Different notations are used to create graphical representation and abstraction of the business process[20]. Business Process Model and Notation (BPMN) is a commonly used

notation to design a graphical business process model. BPMN specification provides a graphical notation for expressing business processes in a business process diagram that can be intuitive to business users and design complex process semantically [17]. The section 2.2 provides more information regarding BPMN.

1.2 Motivation

Business process management is a set of techniques that aim to design, analyze, implement, and improve organizational processes [22]. The increase in business process management is directly connected to the increase in interest and process modeling usage. Business process modeling describes a set of techniques used to design a process workflow, providing visual diagrams that show the way operations are performed to accomplish the organization's planned goals [22]. Business process modeling helps organizations visualizing their processes graphically and allows them to optimize the process structure or execution sequence steps [10].

Researches like [15, 16] shows how business process modeling methodology became a managerial tendency in the last years and subject of several types of research in the literature, indicating that the method has reached a significant level of maturity. This level of maturity also exists for research covering the change impact analysis based on models. However, a few research subjects study the CIA based on the business process model and use BPMN. This dissertation's primary goal is to create a tool based on an approach of change impact analysis to calculate the impact of change on a BPMN business process model file. When we have two versions of the BPMN model file, this tool can analyze both of them and show all the impacted and changed elements. This analysis is done based on the BPMN files' information or in the XML version file.

1.3 Problem

Business process modeling and process management have increased in the last few years, and the software solutions to create and manage an organization's graphical representation of the business process. Most of the solutions present on the market are developed to create and manage BPMN models but rather to compare different versions of models to get information about changes and impact analysis. Because of that, it was found the need to create a tool that allows users to compare different versions of BPMN models and show the change impact analysis report based on the information collected from the models.

With this in mind, this dissertation comes to study change impact analysis based on BPMN models and developed a tool that automates the process of impact analysis calculation between different versions of BPMN models. With this tool, people can upload two different versions of the BPMN model and retrieve information about activities, process flows, artifacts, gateways, and other elements that make part of a BPMN model and calculate the difference between them. This tool will be a simple way to get a report with all the details about changed elements and all the impacted ones. The main contributions of this work intend to have are:

- Simplify the process of change impact analysis based on BPMN models
- Automates existing knowledge regarding change patterns and presents new ones.

1.4 Thesis Structure

This thesis is organized in the following chapters:

- **Chapter 1 (Introduction)**, this chapter presents the introduction of the theme that is discussed in this thesis, providing the context, motivation, problem definition and the structure of the document.
- **Chapter 2 (Background)**, this chapter is dedicated to presenting a literature review about information and concepts considered essential to understanding this thesis's content.
- **Chapter 3 (Change Pattern Catalog)**, this chapter describes the state of the art and literature review regarding change patterns. It contains a catalog that can be used to identify change patterns in a business process model.
- **Chapter 4 (Methodology)**, this chapter presents the approach that was followed to solve the problem described in chapter 1.
- **Chapter 5 (CIA Based on Business Process Model - Tool)**, this chapter describes the implementation and the usage of a tool that automatically apply change patterns to a business process model and create a change impact analysis report.
- **Chapter 6 (Research Validation)**, this chapter describes the examples and case studies performed in order to validate the contribution of this thesis.
- **Chapter 7 (Conclusion and Future Work)**, this chapter addresses the results of the main contribution of this thesis and their respective future work.

Chapter 2

Background

This chapter aims to summarize the most relevant parts of the literature review and pre-requisite information that helps understand the main body of this thesis. The concepts addressed in this chapter include Business Process Model (BPM), Business Process Model and Notation (BPMN), Change Impact Analysis (CIA), Model-based CIA Techniques and Key challenges of CIA.

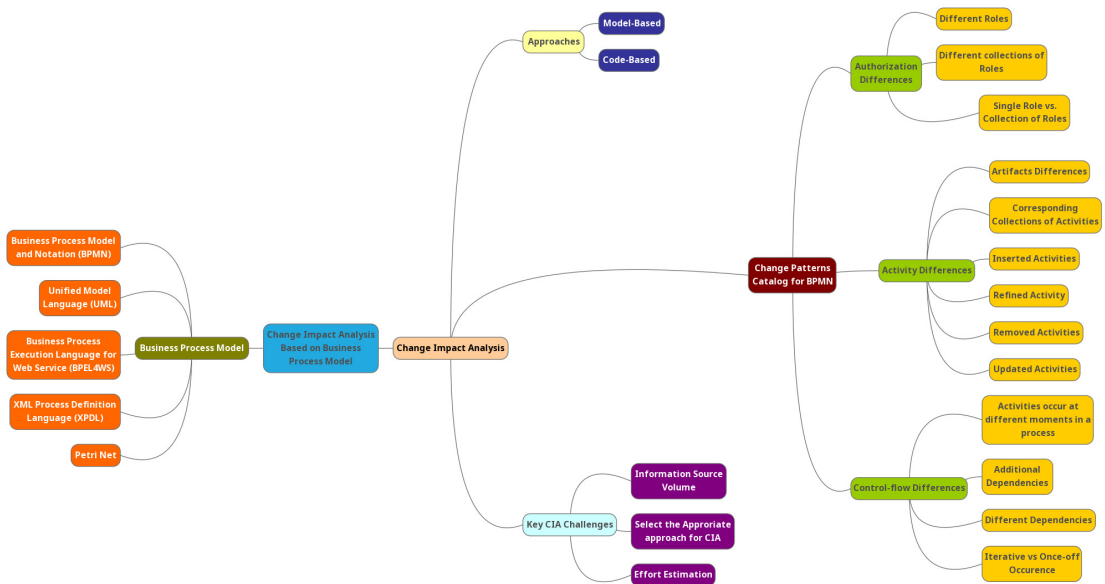


Figure 2.1: Literature review topics

2.1 BPM - Business Process Model

Business Process (BP) is defined by Hammer [8] as a collection of activities that takes one or more kinds of input and creates an output. BP represents an organization's activities/tasks performed to produce a service or product. In order to have a graphical representation of each operation in a BP, companies can use model notations to create this design. The use of graphical notation to describe a Business Process is defined by Business Process Model (BPM) [20]. Using BPM allows companies to better understanding the execution of business processes. We can use different notations to create a BPM, as we can see in the next paragraph.

Organizations like *Object Management Group*, *Business Process Management Initiative*, *Workflow Management Coalition*, and others have proposed during the last years some process modeling methods and standards that can be used to represent a business process in a model. Examples of business process modeling include Business Process Modeling and Notation (BPMN), Unified Model Language (UML), Business Process Execution Language for Web Services (BPEL4WS), XML Process Definition Language (XPDL), Petri Net and others [20]. BPMN is commonly used to understand business operations, while UML based models are more prevalent in software engineering and software development processes documentation. In the next section, the BPMN will be described in more detail, once it is the model notation used in this dissertation to calculate Change Impact Analysis.

2.2 BPMN - Business Process Model and Notation

BPMN is a standard for business process modeling that provides a graphical notation for specifying business processes in a Business Process Diagram (BPD) [14]. It is a commonly used notation for business process modeling, developed by BPMN and maintained by *Object Management Group* to address business process modeling needs, becoming since the reference in this type of modeling. Based on flowcharts, which can be quite expressive regarding their tasks, events, sub-processes, and conditions. It also allows the documentation of data flow in specific processes through its various participants. For these purposes, the notation presents a set of graphic elements that make it more intuitive for everyone involved in the project context. These elements divide into four categories [13]:

1. Flow Objects

- Event

- Activity
- Gateway

2. **Connecting Objects**

- Sequence Flow
- Message Flow
- Association

3. **Swimlanes**

- Pool
- Lane

4. **Artifacts**

- Data Object
- Group
- Annotation

In the first category, we can find flow objects (event, activity, and gateways). These objects refer to a set of flow elements that connect and form a process. An event is something that “happens” throughout a process. Events affect the process flow and usually have a cause or an impact. We represent an event with a circle. Activities represent points in a process flow where works are done. The types of activities that make parts of a process are task, sub-process, and call activity, which allows the inclusion of re-usable tasks and processes in the diagram. A task is used when the work in the process cannot be broken down to a more refined detail level. A task object shares the same shape as the sub-process, a rectangle with rounded corners. A sub-process is an activity where internal details have been modeled using activities, gateways, events, and sequence flows. We use gateways to control how sequence flows interact as they converge and diverge within a process. If the flow does not need to be controlled, then a gateway is not needed.

Another category is connecting objects. There are four ways of connecting flow objects to each other or other information: sequence flow, message flow, association, and data association. A sequence flow shows the order of flow elements in a process, and it is represented by a solid arrowhead drawn among a single solid line. A message flow shows the stream of messages between two participants. In BPMN, two separate pools in a collaboration diagram will represent two different participants. The message flow is a line with an open circle line at the start and an open arrowhead line at the end drawn among a single dashed line. An association links information and artifacts with other BPMN graphical elements, and it is drawn as a single dotted line. Data associations are

used to move data between data objects, properties, inputs and outputs of activities. It is represented with a line arrowhead indicating the association's directional and must be drawn with a single dotted line.

Another category is the swimlanes (pool and lane). They are used to group the primary modeling elements and describe different participants in a process. A participant can be a specific entity (e.g., a company) or a more general role (e.g., a buyer, seller, or manufacturer). A pool is the graphical representation of a participant in a collaboration diagram. It is a square-cornered rectangle drawn by a single solid line. A lane is a sub-partition within a process, and it split a pool into small groups or fragments. In the last category, we have artifacts (data object, group, annotation). They are used to model process data. Data object represents the data created during or at the end of process execution. A group is a mechanism used to grouping elements inside the same category, and annotations allow a modeler to provide additional text information about an element to a BPMN diagram reader.

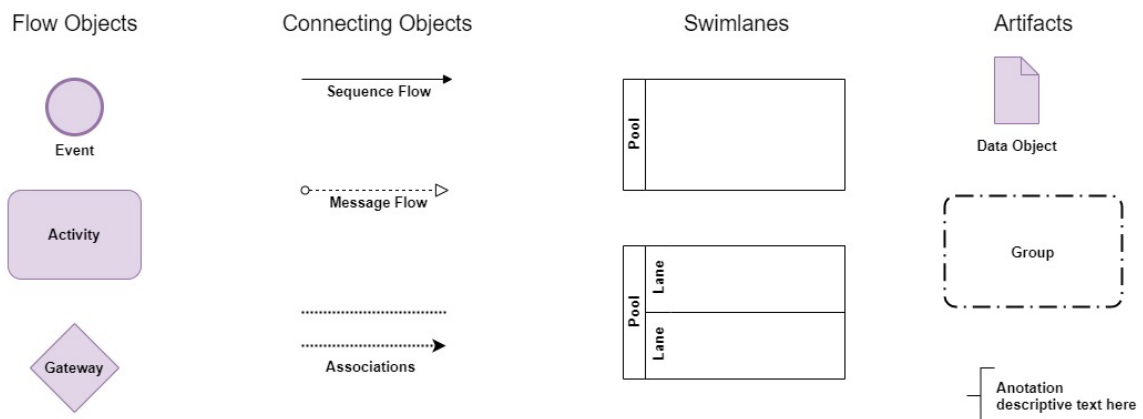


Figure 2.2: Basic BPMN modeling elements. From [13]

2.3 Change Impact Analysis

We use Change Impact Analysis (CIA), or Impact Analysis (IA) in a situation of change of plan, and it is defined as the process to identify all the side effects a given change will affect existing BPM, [3]. The CIA's implementation gave the organization helpful information to calculate the estimation and effort required to perform a change and relevant information about direct and indirect activities (and other business process elements), which would be affected by the introduced change. The impact analysis is an action to be

performed before maintaining a business process that needs to be changed [7]. In order to apply a CIA technique, firstly, a change request must be proposed. Something that must be changed and a way to estimate the effort to implement it. The "technique" means tools, semi-automatic procedures, or manual procedures used to perform the Change Impact Analysis. In the next section, we outline the model-based change impact analysis technique.

2.4 Model-Based CIA Technique

The model-based impact analysis technique, as the name suggests using models to perform the change impact analysis. Before implementing a change, organizations can create models as a business process concept and then calculate the impact analysis. The change impact analysis techniques must be applied to the current modified model and compared with the previous version to determine all the affected elements by introducing change [5]. Previous researches already have tackled the problem of CIA for business processes before. Briand et al.[4, 5], present an automated impact analysis of the UML models approach that can be applied to impact analysis of UML diagrams before the implementation of changes. Their approach can automatically determine the difference between two versions of a consistent model using change taxonomy, including directly and indirectly impacted elements identified by formally defined impact analysis rule (Defined with Object Constraint Language).

Another research that addresses the CIA problem was presented by Watcharin and Twittie [18, 19]. The impact analysis is performed using different business process change patterns applied to the old business process model and the new version of the created model.

2.5 Pattern-Driven Impact Analysis

One of the approaches to perform change impact analysis is Pattern-Driven [18]. This approach is defined by a set of change patterns (described in chapter 3), that can be used to find changes between two versions of the business process model. Previous research on BPM impact analysis already addresses and identifies a set of patterns that can be implemented and used to analyze BPM. Wang et al. [21], on their *Change impact analysis in service-based business processes* article, defined a set of change impact patterns and change types based on the study of the dependencies between services and business

processes. They classify the process change as insert/remove/move an activity, replace/parallelize/sequence activities, embed in conditional branches, embed in the loop and Update conditions and each change impact pattern captures a specific type of change effect. The impact patterns are classified as: add operation impact pattern, remove operation impact pattern, change operation granularity, and transition sequence impact pattern, add conditional or looping transition sequence impact pattern.

Dijkman Remco [6] on their research *A Classification of Differences between Similar Business Processes*, presents a classification that describes frequently occurring differences between business processes. This classification is helpful in the difference detection step. Each model can be analyzed following those classifications differences: authorization differences, activity differences, collections of activities differences, and control flow differences. Another research was presented by Xiao et al. [23]. They describe some primitive changes in a business process level such as inner property modification, input data modification, output data modification, task addition with matched interfaces, etc. These change types are linked with a specific business component impact set which is a set of tasks in the business process affected by a particular service change.

2.6 Key Challenges of CIA

Managing the project changes is difficult since the impact of the changes is not readily known in advance. There are different studies conducted in this subject for the last two decades and several approaches have been proposed, but organizations got challenged in developing and implementing successful strategies for managing change. This section will describe some of the main challenges of Impact analysis, that can turn the change management more difficult in almost corporation and software system development environment.

One of the main challenges for Change Impact Analysis is Information Source Volume [1, 3]. Another main challenge is selecting the appropriate approach of the Impact Analysis [1]. The problem behind this difficulty is the lack of dimensions and hard to know if enough information is available to compare one impact analysis approach with another [2]. On the other hand, lack of a solid framework for classifying and comparing such approaches [11], and the difficulty to find a suitable technique with minimal effort turn this process much harder.

The framework presented by Arnold and Bohner [2] is one of the few works on this subject. This was a three-part framework for characterizing and comparing IA approaches regarding how an approach is used to accomplish the impact analysis, how the impact

analysis process is done internally, and the effectiveness of each approach. Additionally, other main challenges for Change Impact Analysis are Effort estimation and precise Impact Analysis [9]. With a useful change effort estimation and a precise impact analysis result, software project managers can take better decisions on applying changes during the software development phase. Effort estimation is about predicting how much work and how many hours are needed to complete the work. The effort affects directly the project's schedule and costs [12].

2.7 Summary

The subjects and concepts addressed in this chapter are essential in the development of this thesis's works. Without this information, it will not be possible to move forward and implements the process of change impact analysis automatically. The first part of this chapter talks about the business process model and business process model and notation. It goes through the different business process modeling in marketing, but it focuses more on BPMN modeling styles. BPMN is a well-defined modeling notation standard, and it became the most commonly used techniques for business process modeling.

Chapter 3

Change Pattern Catalog

This chapter provided information regarding the change patterns catalog used throughout this thesis and implemented in the framework tool. The list of change patterns present in this catalog was described by [6], except artifact differences, gateway differences, updated and inserted activities differences, that are defined and presented as a new and additional pattern in the catalog.

3.1 Artifact Differences

Artifacts inside the BPMN standard represent a set of items or process data used to express a related kind of information in a business process modeling [13]. The BPMN standard divides artifacts into three types, such as data objects, groups, and annotations. The artifact differences look for a changing pattern only on data object artifact, and it can be distinguished into *Removed and Inserted artifact change pattern*.

If an activity has an artifact reference on the previous process model (A), but no reference representing an artifact equivalent exists on the updated process model (B), as shown in figure 3.1. We identify the presence of **Removed artifact change pattern**.



Figure 3.1: An example of removed artifact change pattern

On the other hand, we will be in the presence of **Inserted artifact change pattern** if an activity does not have an artifact reference on the previous process model (A), but exist an artifact reference on the updated process model (B), as shown on the figure 3.2.



Figure 3.2: An example of inserted artifact change pattern

3.2 Gateway Differences

Gateways are an important BPMN element, so they can control how the sequence flow interacts within a process. Its use is necessary for the context that requires the control of the sequence flow, allowing the modeler to specify a condition and different paths regardless of whether the condition is satisfied. The BPMN standard [13] distinguishes different types of gateways: exclusive, event-based, parallel event-based, inclusive, complex, and parallel. From the change impact analysis point of view, we identify two change pattern differences regarding gateway differences pattern: *Inserted and removed gateway change pattern*.

Inserted gateway change pattern represents the case when a gateway element does not exist on the previous process model (A), but exists on the updated process model (B), as shown on the figure 3.3.

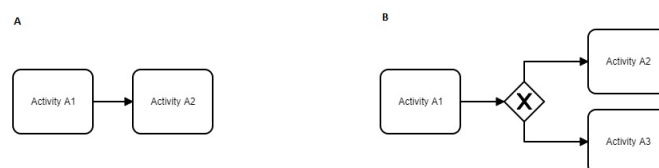


Figure 3.3: An example of an inserted gateway change pattern

Removed gateway change pattern represents the case when a gateway element exists on the previous process model (A). However, no equivalent gateway exists on the updated process model (B), as shown on the figure 3.4.

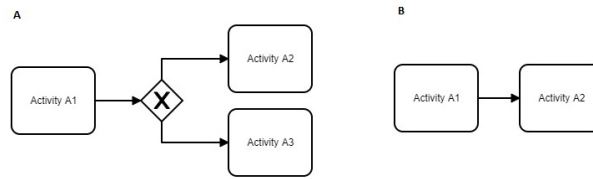


Figure 3.4: An example of removed gateway change pattern

3.3 Authorization Differences

Business authorization roles can be mapped to a BPMN model using pools and lanes. They represent the role that each participant uses to perform an activity within the business process. A difference in the authorization to perform an activity exists if we assign two different roles to activity in two versions of the business process model. There are three different types of authorization differences identified from two models version: *Different roles*, *single role vs. a collection of roles*, and *different collections of roles*.

Different role change pattern represents the case in which an activity assigned to one role in a previous process model (A) has a different role in an updated version of the model (B), as reproduced on the figure 3.5.

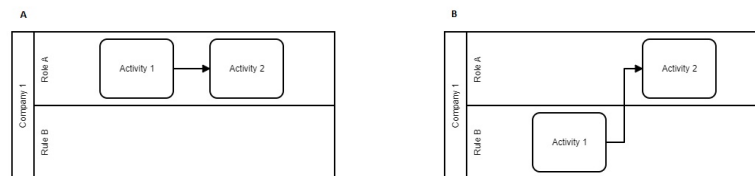


Figure 3.5: An example of different role authorization pattern

Single role vs. collections of roles describe the authorization difference change pattern in which an activity is assigned to a single role in a previous process model (A). However, it is assigned to multiple roles on the updated version of the model (B), as shown in figure 3.6.

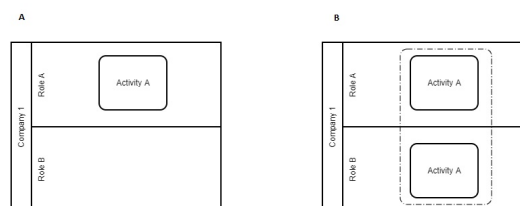


Figure 3.6: An example of a single role vs. collections of roles authorization pattern

Different collections of roles describe the authorization difference change pattern in which an activity is performed by one collection of roles in one process (A), performed as an interaction between those roles, and another collection of roles in the other process. As shown by the example in figure 3.7.

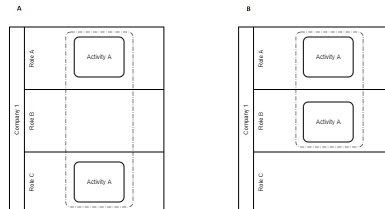


Figure 3.7: An example of different collections of roles authorization pattern

3.4 Activity Differences

Activities and tasks are some of the main elements that appear inside a business process model. The activities represent the business process events performed to achieve a goal. A difference between activities exists if a unit of work represented by a specific collection of activities in one process is represented by a different collection of activities, or not at all, in the other process. This section defines different types of activity differences change patterns used to identify changes between two versions of models. *Removed activity*, *updated activity*, *refined activities*, *corresponding collections of activities*, *inserted activity*.

Removed activity. Describes the activity differences change pattern performed to identify if an activity exists on the previous process, model (A). However, no activity representing an equivalent exists on the update process model (B). Figure 3.8 shows an example of this type of activity difference change pattern.



Figure 3.8: An example of removed activity change pattern

Updated activity. An updated activity change pattern exists if an activity that exist in the previous process model (A) still exists on the updated process model (B), but an update

occurs to the activity name on the last version of the model. An example of an updated activity change pattern was presented in the figure 3.10.



Figure 3.9: An example of updated activity

Refined activities. Sometimes, we can split an activity into collections of small activities to satisfy its accomplishment. The refactoring of activity is identified on the change impact analysis using the refined activities change Pattern. This pattern represents the case in which an activity exists in a previous process model (A) but represented by a collection of activities on the updated process model (B). The refined activity change pattern is a semi-automate pattern. Once, the user must identify the collection of activities representing the equivalent ones on the updated model.

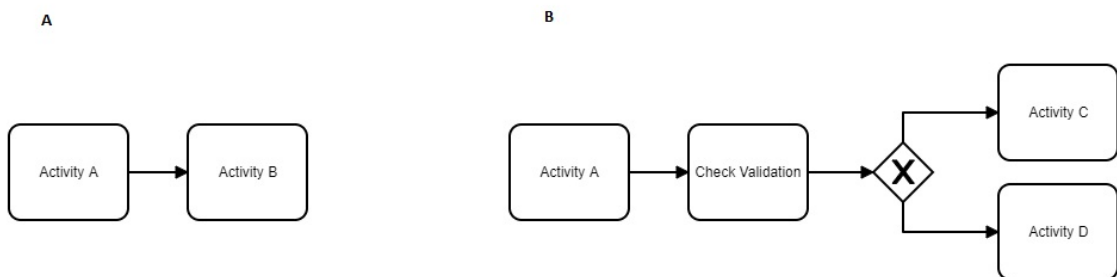


Figure 3.10: An example of refined activity change pattern

Corresponding collections of activities. As described on the previous change pattern (refine activities), corresponding collections of activities describe the activity difference change pattern in which a set of activities in the previous process model (A) are represented by a collection of equivalent activities on the updated process model. Another semi-automated activity difference change pattern, user interventions, is necessary to identify and map both models' activities.

Inserted activity. Describes the activity difference change pattern performed to identify if an activity that does not exist in the previous process model (A) exists in the update

process model (B). Figure 3.12 shows an example of this type of activity differences change patterns.



Figure 3.11: An example of inserted activity change pattern

3.5 Control-flow Differences

Control-flow represents the way nodes connected inside a business process model. They are used to create the path from one element to another. Control-flow differences change patterns exist if (collections of) activities in one process have different control-flow relations with each other than equivalent (collections of) activities in the other process. This section outlines the different types of control-flow differences used to identify change pattern between two versions of models: *Different dependency*, *Additional dependency*, *activity occur at different moments in processes*, *iterative vs. once-off occurrence*.

Different dependency. Describes the control-flow differences change patterns performed to identify if equivalent activities from two processes have different dependencies on which they depend. Figure 3.12 shows an example of this type of control-flow difference change pattern.

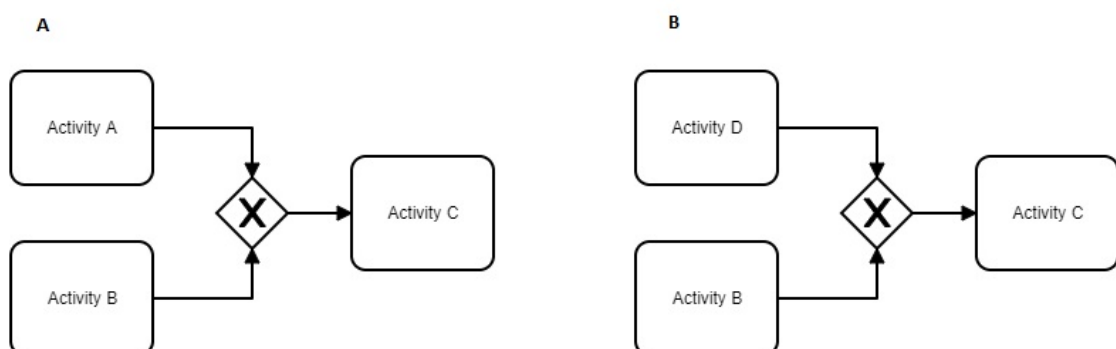


Figure 3.12: An example of different dependency change pattern

Additional dependency. Describes a particular case of having different dependencies change pattern where a set of activities includes the other. Figure 3.13 shows an example of additional dependency change pattern.



Figure 3.13: An example of additional dependency change pattern

Activity occur at different moments in processes. A particular case of different dependencies change patterns, which allows users to distinguish the control-flow differences change pattern when collections of activities are disjoint. Look at the example shown in figure 3.14.

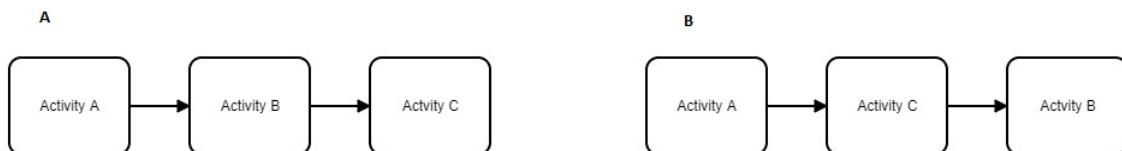


Figure 3.14: An example of activity occur at different moments in processes change pattern

Iterative vs. once-off occurrence. Another particular case of different dependency change patterns. The activity execution follows the one go direction in the previous process model (A). In contrast, in the updated process model (B), the activity is performed in a loop until it is completed. Look at the example shown in figure 3.15.

3.6 Summary

The pattern catalog present in [6] describes the main aspects of changes related to BPMN. The descriptions are self-explanatory and straightforward to understand and to be implemented in the tool. Apart from Remco's patterns, the catalog does not cover all the changes in every element. That is why we introduce two more patterns to the catalog and

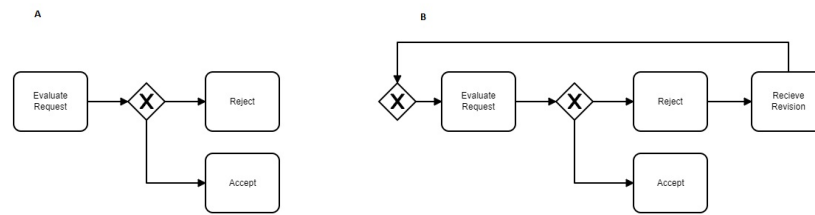


Figure 3.15: An example of iterative vs. once-off occurrence change pattern

update the activity differences change patterns with the insert activity, and update activity change types. We believe that more patterns will appear in the future. By creating specific examples and model cases, new patterns can arrive from that.

The catalog of change patterns represents the tool's base to compare BPMN model files and retrieve impact analysis information from that. Implementing these patterns will give more options to identify changes and consequently impact elements from that change. The more change patterns identified and inserted into the catalog, the more complete will be the tool after implementing these patterns.

Chapter 4

Methodology

This chapter describes the approach followed to solve the problem described in chapter 1. It includes a simplified representation of how the problem of change impact analysis was implemented, including model comparison and difference calculation between them. It also describes the N'steps change impact analysis algorithm, used to calculate all the impacted elements following the number of steps chosen by the user.

4.1 Introduction

There is a need for a tool that allows the comparison between two different versions of the BPMN model and presents a report of changed and impacted elements based on the information kept on those models. This chapter explains the proposed model that shows the process followed to perform a change impact analysis between two versions of the BPMN model. The model shows all the steps behind the scenes of change impact analysis process, implemented in a BPMN 2.0 model notation.

4.2 Approach and Model

The change impact analysis tool approach model consists of choosing two versions of equivalent models and comparing them to see which elements change from one model to another (See fig. 4.1). The model changed elements will be saved in a list, and then all the impacted elements are calculated for each element on the list. With the list of changed elements and impacted elements, a report can be built and presented to the user. To get comparison differences between the models, the tool relay on the change patterns

catalog (See chapter 3) to automatically looking and identify correspond patterns within a business process model.

In order to each change patterns on the catalog works as expected and identify all changed elements with the expected behavior, we recommend that the last version of the model is an upgrade from the old version and not a different version build from scratch. A model version built from scratch creates a different Ids on their elements. In this case, the result of changed elements will not be as accurate as expected, and the change impact analysis will not show the accurate result. To not going into this scenario, the result of the CIA depends on the correct model update.

With the comparison of the model task done, the tool's next task is the impact analysis. At this time, it performs the calculation of the impacted elements, and then the report of the impact analysis is built. The calculation of the impacted elements for each element of the list of changed elements is done by identifying their target element and using the N'Steps impact analysis algorithm (more info on section 4.3) to identify indirectly impacted elements. In the end, a report is build based on the information store in the comparison task and impacted elements calculation. The report will show the list of changed elements and all impacted elements. Based on the information of changed elements, we built two types of charts. The first one shows the percentage of unchanged elements compared to the changed ones. The second one shows the difference between old and updated models regarding the number of each type of element present on each model.

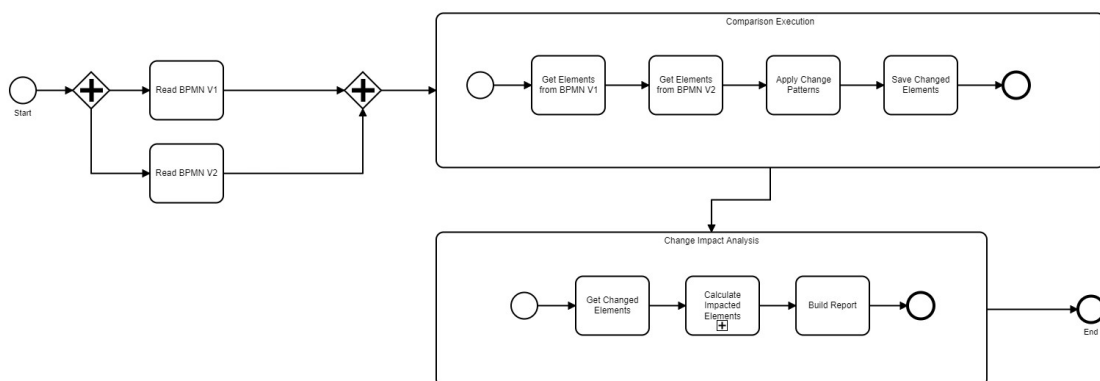


Figure 4.1: Change Impact Analysis Model

4.3 Impact Analysis in N's steps

Here, we present an algorithm that allows end-users to define how long a changed element can impact forward elements. To accomplish this, users must choose the number of steps to go forward before he/she starts the analysis. The number of steps must be an integer and greater than or equal to one. If the user chose an invalid number, the system would set the step number equal to one. Using the following definitions allows the accomplishment of the process of indirect impacted elements calculation.

Definitions:

Let E be the set of all elements, n an integer greater than or equal to one be the number of steps and E' be the set of all changed elements ($E' \subset E$). We use these definitions to explain the calculation formula of direct and indirect impacted elements.

Direct Impact

A changed element directly impacts an activity element if both elements are directly connected through a sequence flow or connected through a gateway element. In this case, the formula rule that describes the direct impacts is given in the following way. Given a changed element $e' \in E'$ and an activity element $e \in E$, the direct impact I that e' has in e when $N = 1$ is represented by association relation $(e' \times e, N)$.

Indirect Impact

On the other hand, a changed element indirectly impacts an activity element, when both elements are not directly connected, but connected by set of elements between them. In this case the formula rule that describes the indirect impacts is given in the following way. Given a changed element $e' \in E'$, an activity element $e \in E$ and n be the number of steps to calculate all directed impacts of elements in the middle. The indirect impact I that e' has in e given $N = n$ is defined by a set of directed impacts $\{I_1, \dots, I_n\}$ of all elements between e' and e .

Apart from the definitions presented above, the rest of this chapter shows the algorithms used to calculate impacted elements. The first algorithm *GetTargetElement* is an auxiliary algorithm that aims to retrieve all elements connected to the changed element. Besides that, the section presents *CalculateImpactedElements* and *GetImpactedElementsInSteps* algorithm which calculate direct and indirect impacted elements respectively.

The *GetTargetElement* algorithm receives two input parameters, an element (e) and the list of all elements (E), and returns a set of target elements (TE) sub-set elements of E . Firstly, it defines two variables, the fn that converts the input element e to a *FlowNode* type, and the variable O , to be a set of sequence flows. After that, it performed a repeat cycle over the sequence flow list to get the target element and then added to the target elements list (TE).

Algorithm1: GetTargetElement

```

1: Input  $e \in E$  and  $E$ 
2: Output  $TE \subset E$ 
3: Let  $fn = (FlowNode)e$ 
4: Let  $O = fn.getOutGoing()$ 
5:  $TE \leftarrow \emptyset$ 
6: for all  $o \in O$  do
7:    $TE \leftarrow o.getTarget()$ 
8: end for

```

The second algorithm (*CalculateImpactedElements*) can be used to retrieves directly impacted elements. It receives a list of changed elements and a list of all elements as input parameters and returns a list of impacted elements.

Algorithm2: CalculateImpactedElements

```

1: Input  $E'$  and  $E$ 
2: Output  $I \subset E$ 
3:  $I \leftarrow \emptyset$ 
4: for all  $e' \in E'$  do
5:    $I \leftarrow GetTargetElement(e', E)$ 
6: end for

```

The last algorithm represented on this chapter is *GetImpactedElementsInSteps*. It determines the steps necessary to retrieve the indirectly impacted elements. As we saw in the definition of indirect impacts, the changed element and the impacted ones are not directly connected. In this case, the user must define the number of steps whereby the impacted elements are calculated. The algorithm receives as a parameter a list of changed elements, a list of all elements, the number of steps, and returns the list of indirectly impacted elements.

Algorithm3: GetImpactedElementsInSteps

```

1: Input  $E', E$  and  $N$ 
2: Output  $I \subset E$ 
3:  $I \leftarrow \emptyset$ 
4: Let  $c \leftarrow 1$ 
5: while  $N \geq 1$  do
6:   if  $c == 1$  then
7:      $I \leftarrow \text{CalculateImpactedElements}(E', E)$ 
8:   else
9:      $I \leftarrow \text{CalculateImpactedElements}(I, E)$ 
10:  end if
11:   $c \leftarrow c + 1$ 
12:   $N \leftarrow N - 1$ 
13: end while

```

4.4 Summary

This chapter presents the approach and meta-model defined to solve impact analysis and BPMN model differences comparison. The representation approach model is constituted by two main sub-process, "the comparison execution" and "change impact analysis activities." The first one, get the elements from BPMN files and apply change patterns to identified changed elements. The elements that change from one model to another are saved and used in the next activity. The change impact analysis task gets the list of changed elements and starts to calculate the impacted elements. In the end, the information gathered during the analysis is present on the report page. This chapter also presents the definition of the direct and indirect impact and the algorithms to calculate the impacted elements.

Chapter 5

Change Impact Analysis Tool

This chapter presents all the details about the implementation of the tool that automates change impact analysis. The tool can extract information stored in a BPMN file and present it to the end-user simply and intuitively. Many features and functionalities are present on this tool, such as BPMN file upload, shows model file information, compare two models and change impact analysis. This tool has already implemented 50% of the change patterns described in chapter 3.

5.1 Purpose

There are many cases in which the business process must change in order for organizations to accomplish new goals and objectives. Changing an existing business process can cause many problems for the organization if they did not know all the effects of changed elements previously. Following a manual approach to achieve this task can take a long time on its execution and proportionally more error-prone. Thus the benefit in automating the process of change impact analysis becomes even necessary in this situation. As such, this tool's primary goal is to provide the means to automatically apply change patterns to get comparison information between two versions of the BPMN model and show the impact analysis report regarding each change.

5.2 Architecture

In the case of a web platform, one of the primary purposes is to provide users with information. Usually, the information is accessed by users through a web client that connects

to the server to extract the information. For this, the tool's architecture developed during this work is based on Java EE multilayer client-server architecture (See fig. 5.1). This architecture style allows the separation of the application into different tiers and layers such as client tier working on the client machine (presentation layer), web and business tier working on a Java EE server (business layer), and Enterprise information system tier working on the database server (data layer). The client tier consists of a dynamic Web page built using markup languages (HTML and XML), which presents the server's data. The client tier does not perform complex tasks with business rules or database queries. They delegate such operations to the components of working on the business tier.

Business logic tier is where data processing and solving problems faced by the application are done, such as calculating the difference between the two models. This layer is composed of two different containers. The web container includes components such as Servlets and JavaServer Faces (JSF) pages used to receive and forward information to clients. In contrast, the enterprise java bean (EJB) container does the information processing tasks. This tool does not save any data in a database so that it does not include the data layer. The tool uses the information present on the model files.

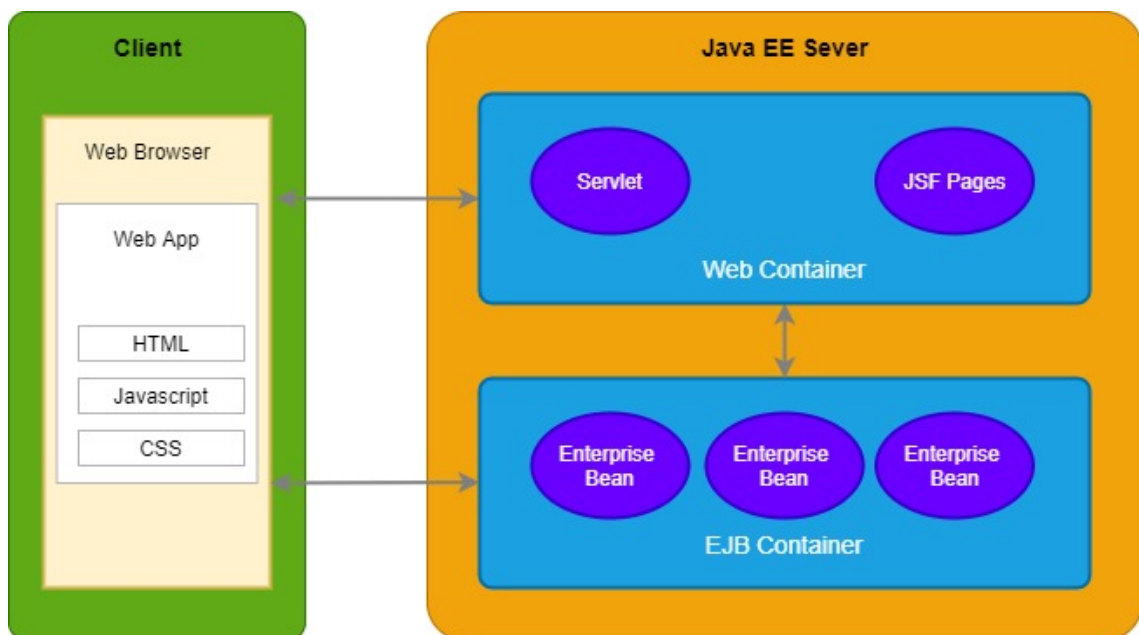


Figure 5.1: Tool Architecture

5.3 Dependencies

This tool was developed in Java, targeting the Java EE 8 development kit, and compiled with JRE compiler version 1.8. It makes use of the Apache Maven to manage the build-

ing aspects of the project (how we built it and its dependencies). We use the Maven Webapp Archetype to create the packaging structure of the project. The project contains some dependencies used for many purposes and reasons. However, we will highlight the Camunda BPM Model used to manage BPMN files and Primefaces used together with Bootstrap to create the user interfaces, and JUnit to implement the unit testing approach.

5.4 User Interface

The tool was built with the principle of simplicity since it begins. In that case, the user interfaces must be simple that any user can use it without much learning effort. The simple interface must keep all critical information to the user simultaneously that skips not relevant ones. The application comprises three different sections, such as the top bar section, side menu section, and content area section. The top bar section contains the name of the application on the left side, and on the right, it shows a dropdown menu which the user can use to change the language. Only the English language is supported now, but in the future other languages will be included.

The side menu section contains links to several navigation options to pages inside the application. It also has links that redirect the user to online resources out of the tool (BPMN Standard and BPMN Examples). For each menu option selected by the user, a different content appears on the main content area section. The main content area section is an active and typical section of the application that shows information regarding the selected menu. Next, we present the first page of the application.

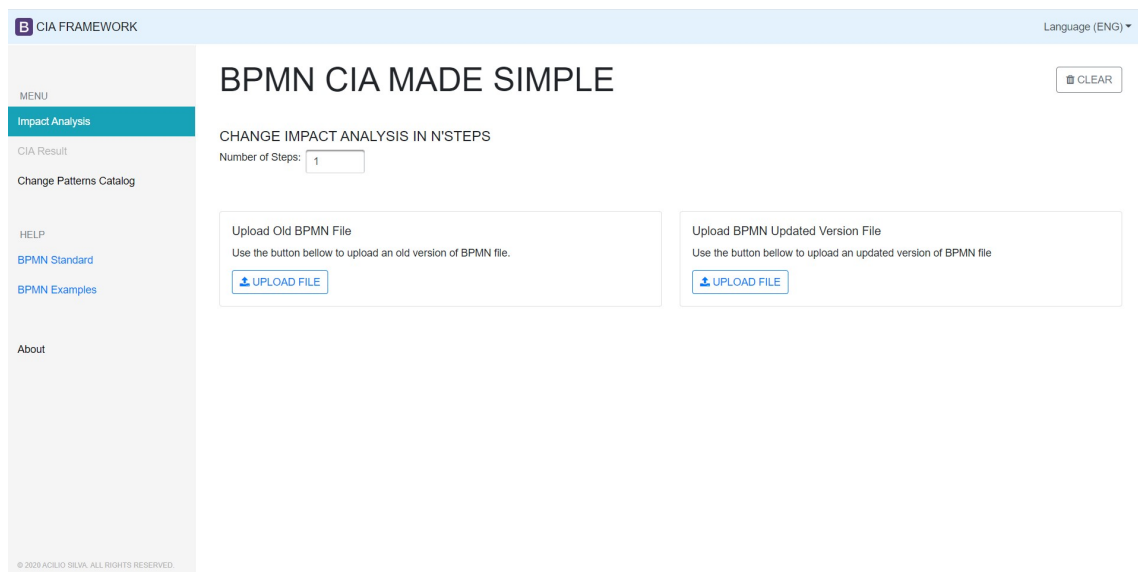


Figure 5.2: First Page of the application

Figure 5.2 shows the interface related to the first page that appears to the users on the first entry into the application or when the user reloads the page. The Impact Analysis menu option is selected by default when we enter the application or reloads the page. The content area shows, in this case, the upload file buttons, the text field with numbers of steps defined to one, and a button on the top right corner to reset those values if changed. By uploading the BPMN files, the application content page will change to show the files' names and the model content, as we can see on the figure 5.3. The analyze button also appears on the top right corner of the main content area, showing that the user can now analyze the BPMN models.

Figure 5.3: Impact Analysis page with model files uploaded

By default, the CIA Result option on the side menu is disabled (See fig. 5.2). This option will activate at the end of the user's analysis process execution, clicking on the analyze button on the impact analysis page. The user is automatically redirected to the result page after the analysis process execution finalizes. On this page, the user will see the information regarding changed elements and impacted activities (See fig. 5.4). At the end of the page, a graphical comparison between the old and updated models shows the percentage of changed and unchanged activities, and the number of flow elements that exist on each model, figure 5.5. At the top of the main content area on the page, the users can see the name of the analyzed models and the export result button at the top right corner. The export result functionality does not work now but is added in the future release version.

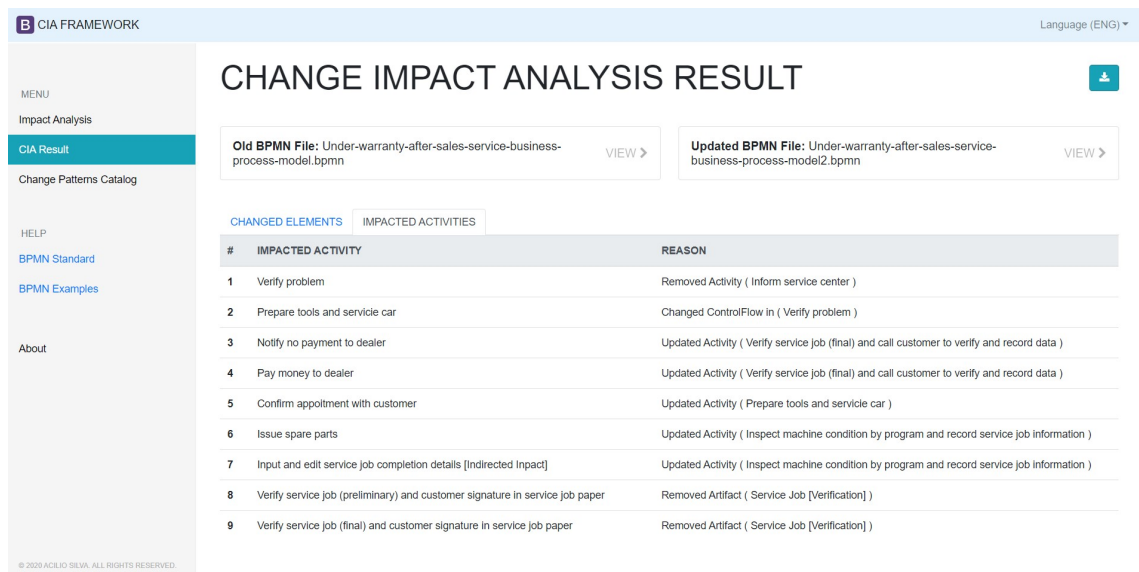


Figure 5.4: Change Impact Analysis Result - Impacted Activities

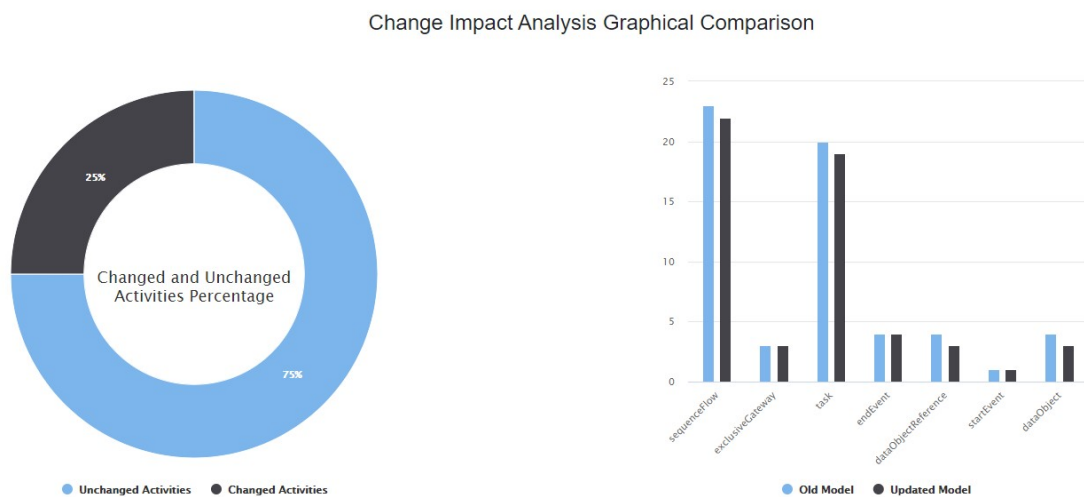


Figure 5.5: Change Impact Analysis Result - Graphical comparison

5.5 Technologies

To implement the tool, we rely on a set of web technologies used to create a monolithic structure that supports all the necessary solutions to automate the process of change impact analysis. Based on the application’s architecture, we divide them into three categories: presentation layer, business layer, build, deploys, and source control.

5.5.1 Presentation Layer

Primefaces

Primefaces is an open-source and cross-platform user interface (UI) library for JSF (JavaServer Faces) based applications. With more than 100 UI components widget, it is easy to use primefaces to implement any UI requirements.

Bootstrap

Bootstrap is an open-source CSS framework directed at responsive, mobile-first front-end web development. It is also very used to build web applications of any kind. It is using allows a quick layout definition and responsiveness achievement.

jQuery

jQuery is a JavaScript library designed to simplify HTML DOM tree manipulation, event handling, CSS animation, and Ajax. Its scripts run directly on the client-side.

5.5.2 Business Layer

Java EE

The Java Platform, Enterprise Edition (Java EE) or Java 2 platform (J2EE) is a collection of Java APIs used to write web applications. It provides APIs that simplify the most common challenges facing developers when building modern web applications.

Camunda BPMN Model

The Camunda BPMN model API enables easy extraction of information from an existing process definition, editing an existing process definition or creating a complete new one without manual XML parsing.

5.5.3 Build, Deploys and Source Control

Apache Maven

The Apache Maven is an automation tool used to build and manage the dependencies of any Java-based project. Their objectives are to making the build process easy, providing a uniform build system, providing quality project information, and encouraging better development practices.

Git

Git is a distributed version-control system for tracking changes in source code during

software development. It is designed for coordinating work among developers, but it can be used to track changes in any set of files.

Heroku

Heroku is a container-based cloud Platform as a Service (PaaS), allowing developers to deploy, manage, and scale modern apps. By using Heroku's services, developers have more freedom to focus on their core product without the distraction of maintaining servers, hardware, or infrastructure.

5.6 Summary

This chapter describes the tool developed throughout this dissertation. A simple and innovative web application that combines BPMN files' information, change pattern catalog, and impacts analysis algorithm to generate a report of change impact analysis to the end-user. The tool was built over modern web application development technology, showing an easy to use and intuitive user interface. The user does not need to have any previous knowledge about the tool to use it.

Chapter 6

Validation

This chapter presents a set of BPMN examples and case studies that validate the tool developed during this thesis. The examples BPMN files used in the tool's validation process are available on OMG's website and, more specifically, in the about page of the business process model and notation specification version 2.0.

6.1 Example 1 - Hardware Retailer

In this first example, we can find a business process model that describes the preparation steps a hardware retailer must fulfill before the ordered goods can be shipped to the customer. Figure 6.1 shows a basic BPMN model example, composed of a pool and different lanes for the different participants involved in this process and other flow elements. The straight start event "goods to ship" indicates that this preparation should be done now. Right after the instantiating of the process, there are two things done in parallel, as the parallel gateway indicates; While the clerk has to decide whether this is a regular postal or a particular shipment, the warehouse worker can already start packaging the goods.

The clerk's task is followed by the exclusive gateway "mode of delivery," which works as a route to the sequence flow based on the previous task's result. If needs a particular shipment, the clerk requests quotes from different carriers, then assigns a carrier and prepares the paperwork. However, if a regular post shipment is sufficient, the clerk needs to check if extra insurance is necessary. If that extra insurance is required, the logistics manager has to take out that insurance in a parallel task, while the clerk has to fill in a postal label for the shipment. Furthermore, the synchronizing gateways are included to

ensure that everything has been fulfilled before the last task, "add paperwork and move package to pick area," is executed.

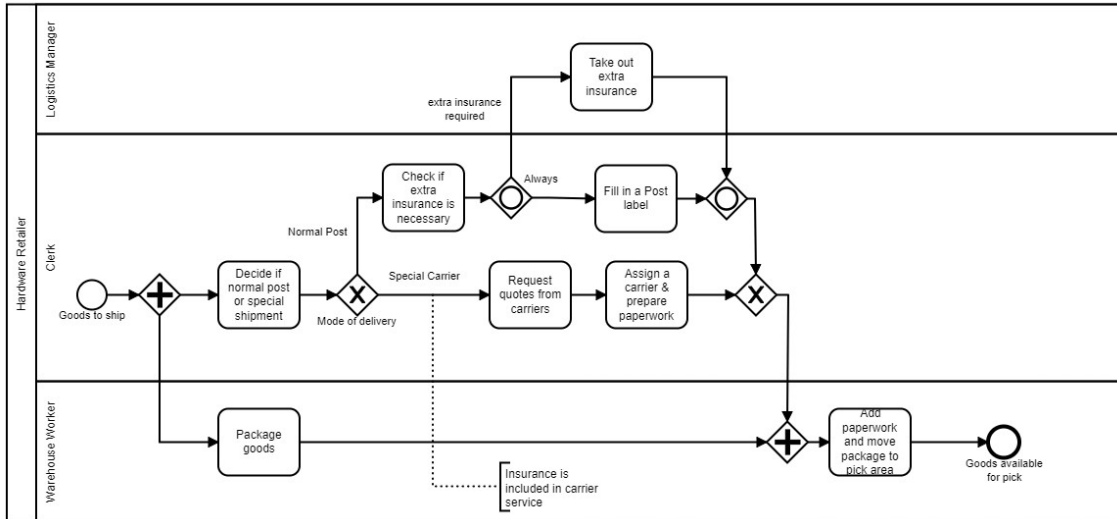


Figure 6.1: Shipment process of a hardware retailer.

6.1.1 Insert, Update and Remove Activities

The first scenario will show the demonstration of three principal changes that occur in a BPMN model file. The change types under test are: insert, update, and remove activities. Figure 6.1 represent the original version of the process model. While figure 6.2 represents the updated version that reflects the changes made to the original model. The introduced

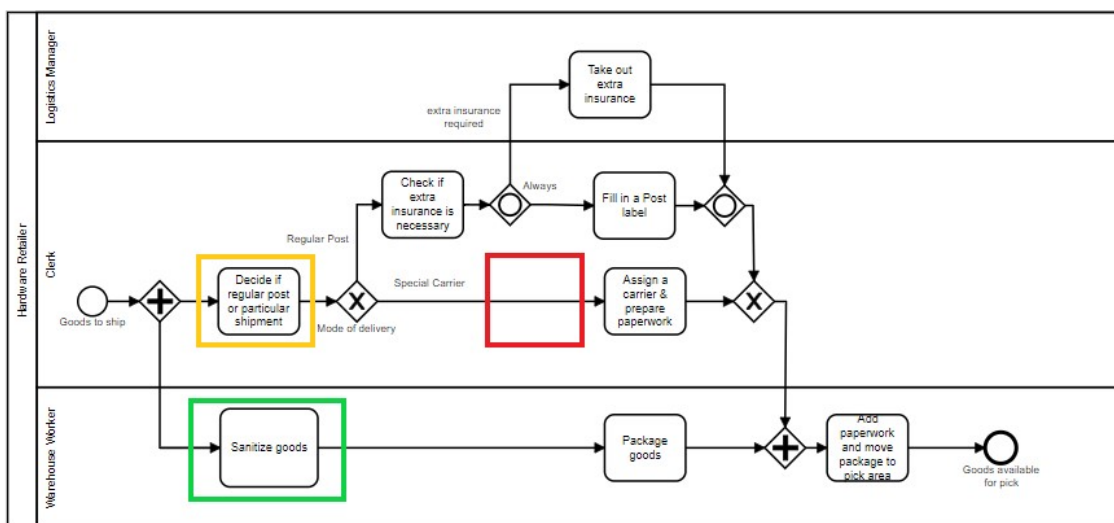


Figure 6.2: Shipment process of a hardware retailer updated version.

changes have a signaled color: green, yellow, and red that corresponds to the inserted, updated, and removed activity changes performed on the original file.

6.1.2 Result

By comparing the original file with the updated one, this analysis's expected result includes a list of changed elements and the introduced changes' impacted elements. The result of this analysis is related to the change introduced on the tasks "Decide if normal post or special shipment," "Request quotes from carriers," and "Sanitize goods," where this latter being inserted in the updated version.

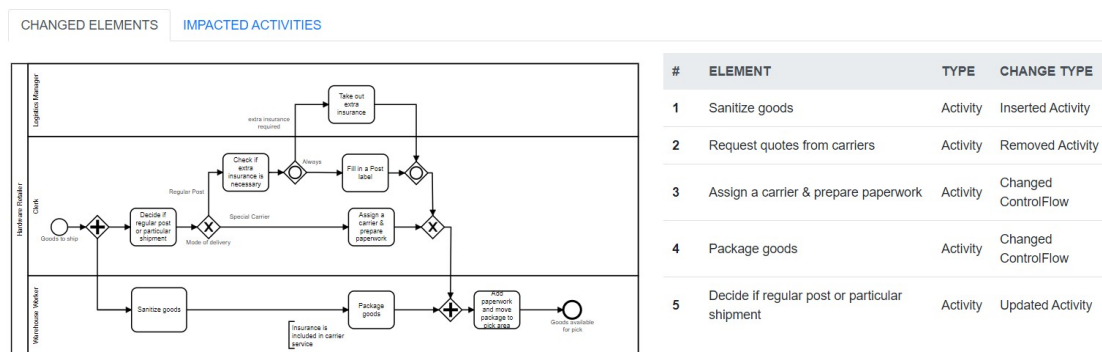


Figure 6.3: Shipment process of a hardware retailer changed elements.

As we can see in Figure 6.3, the application screen of the change impact analysis result shows the list of changed elements, and the updated model version. The list of changed elements presents more two changed elements than the ones described in the section above. "Assign a carrier and prepare paperwork and "package goods" appear on this list because these elements' control flow changes with the inserted and removed element update changes. In this case, the control flow change pattern identifies this change and shows it to the end-user.

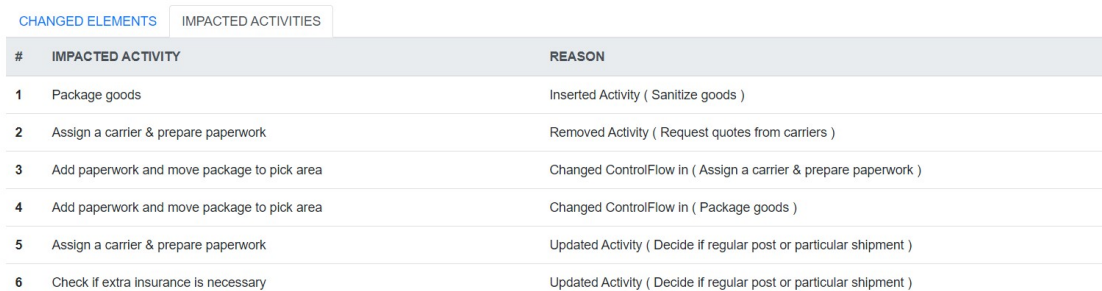


Figure 6.4: Shipment process of a hardware retailer impacted activities.

Apart from the changed elements, the page has another tab denominated by impacted activities. This tab contains the list of activities impacted by the changes introduced on elements that appear in the first tab regarding the number of steps. For this example, the number of steps chosen is one. Figure 6.4, shows the result of impacted activities for the example of changes presented in the above section. The list contains the impacted activity's name in one column, and in another column, the reason for the impact describes the type of change pattern and the name of the activity where that changes occur.

6.2 Example 2 - Incident Management

In this section, we want to present a second example for the tool validation process using an incident management business process BPMN model example. This section's primary purpose is to demonstrate the usage of gateway difference change patterns to identify gateways changes.

The incident management process of a software manufacturer shown in figure 6.5, is an entirely human-driven process triggered by a customer requesting help from the account manager because of a problem in the purchased product. First of all, the account manager should try to handle that request on his own and explain the solution to the customer, if possible. If not, the account manager will hand over the issue to a 1st level support agent, who will hand over to 2nd level support, if necessary. The 2nd level support agent should figure out if the customer can fix the problem on her own, but if the agent is not sure about this, he can also ask a software developer for his opinion. The 2nd level support agent will insert a request for a feature in the product backlog if the software product's current release cannot cover the customer's demand satisfactorily. In any case, in the end, the account manager will explain the solution to the customer.

6.2.1 Insert and Remove Gateways

Another scenario inside the change impact analysis is shown by gateway differences change patterns. This pattern describes how to identify the changes in a gateway element from one model to another. During this thesis's work, the tool developed can identify two types of changes related to gateways elements: insert and remove actions. Figure 6.6 shown the updated version of the incident management model. It has already signaled the changed elements with green and red color to demonstrate the insertion or deletion action performed on the updated version. Apart from the changes in gateway elements, the figure also includes the change of an inserted activity element.

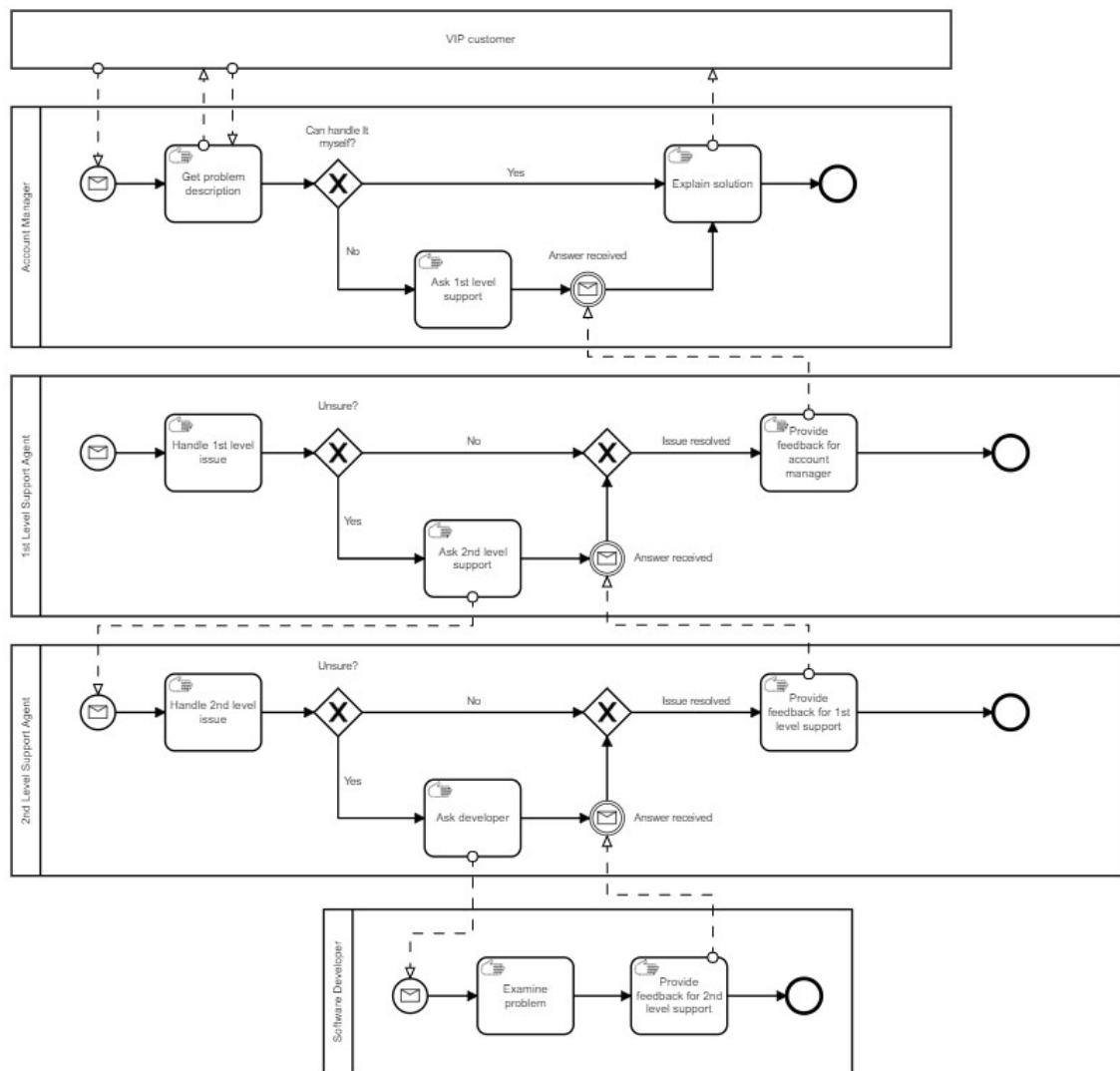


Figure 6.5: Incident management business process

6.2.2 Result

The change impact analysis of the original model file with the updated one produces the following result. One activity was inserted in the updated model, representing 8% of changed activities. One gateway was inserted, and another one was removed from the updated model. In this case, the number of gateways remains the same on both models, as shown in figure 6.8. The list of all changed elements includes two gateways and one activity. Like in the first example, the application was able to identify all the elements

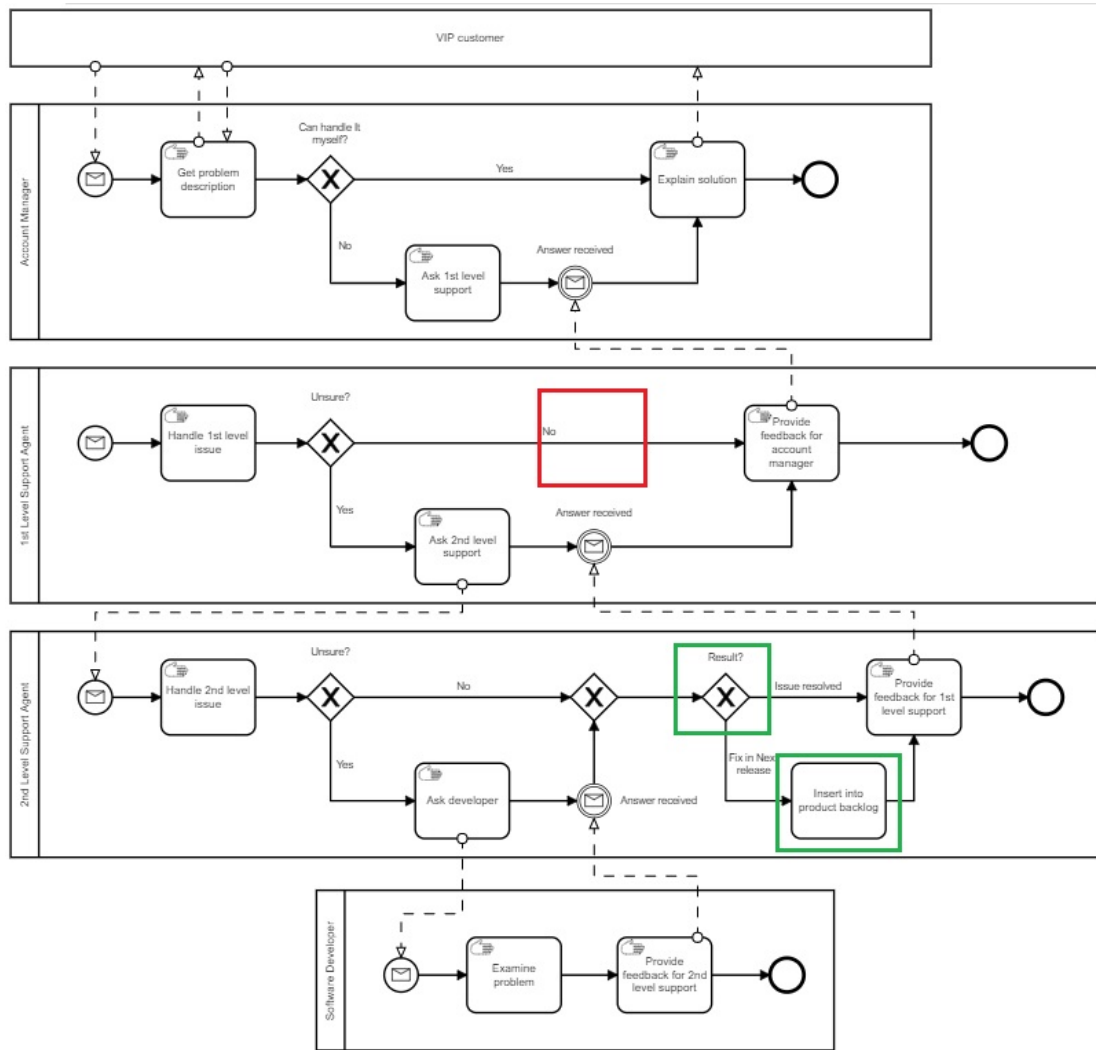


Figure 6.6: Incident management business process updated model

that have changed from the old to the updated model. Figure 6.7 shows a list of activities impacted by the changes performed in the model.

CHANGED ELEMENTS		IMPACTED ACTIVITIES
#	IMPACTED ACTIVITY	REASON
1	Provide feedback for 1st level support	Inserted Activity (Insert into product backlog)
2	Provide feedback for 1st level support	Inserted Gateway (Result?)
3	Insert into product backlog	Inserted Gateway (Result?)
4	Provide feedback for account manager	Removed Gateway ()

Figure 6.7: Impacted activities CIA result

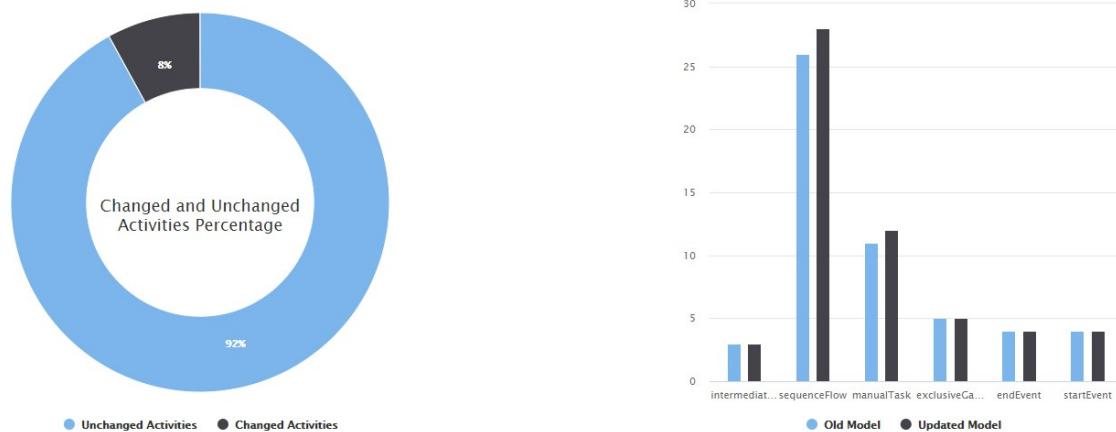


Figure 6.8: Impact analysis result statistic

6.3 Example 3 - Email Voting

This section describes the third example performed to demonstrate the validation of the change patterns used in the impact analysis tool. Apart from the change patterns demonstrated in the previous examples, this section's primary purpose is to demonstrate the usage of artifacts differences change patters. In this case, it uses an example of a business process for resolving issues through e-mail votes (fig. 6.9) in the demonstration purpose.

The process defines the point of view from the manager of the Issues List's perspective and the discussion around this list. From that point of view, the working group's voting members are considered external Participants who will communicate messages. The Issue List Manager reviews the list and determines if any issues are ready for going through the discussion and voting cycle. Suppose there are no issues ready, the process over for that week. Otherwise, the process will continue with the discussion cycle. The "discussion cycle" is a sub-process that includes other tasks, intermediate events, and decision gateways. Before the "Collect Votes" sub-process, the issue manager list sends out an email to announce to the working group and the voting members that the issues are now ready for voting. The last section of the process goes through a set of decisions and loops to prepare the voting results, email them to the voting members, and post them on a web site.

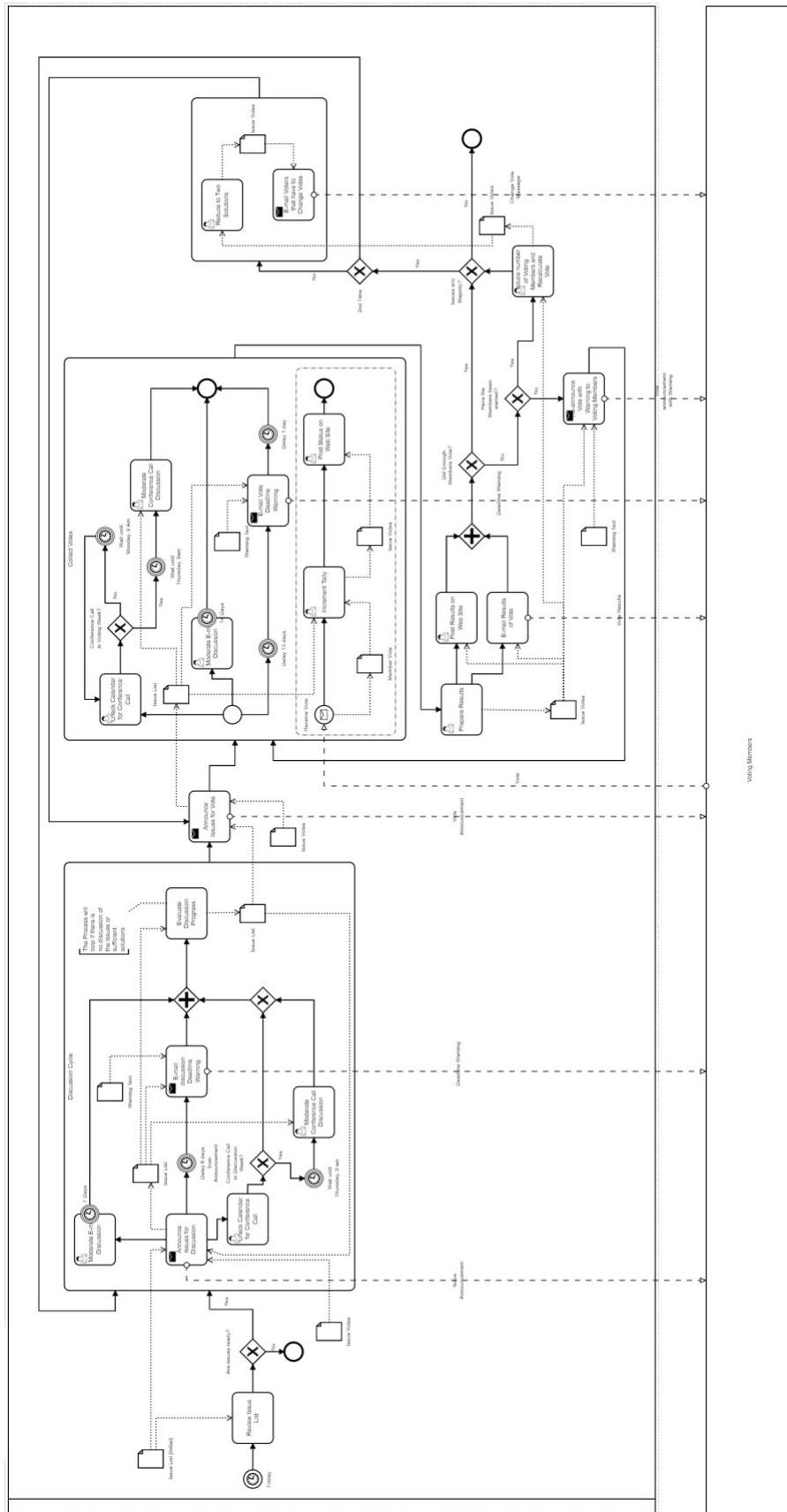


Figure 6.9: Email voting business process

6.3.1 Inserted and removed artifacts

Artifact objects outline the process of data information in a business process. Artifact differences change patterns show another scenario inside the change impact analysis process that describes how to identify change patterns related to the artifact element. The artifact differences change patterns identified by the tool are: inserted and removed change patterns. Figure 6.10 shown the updated version of the "email voting model". Each change is already signalized with green and red color to demonstrate the insertion or deletion change action performed on the updated version.

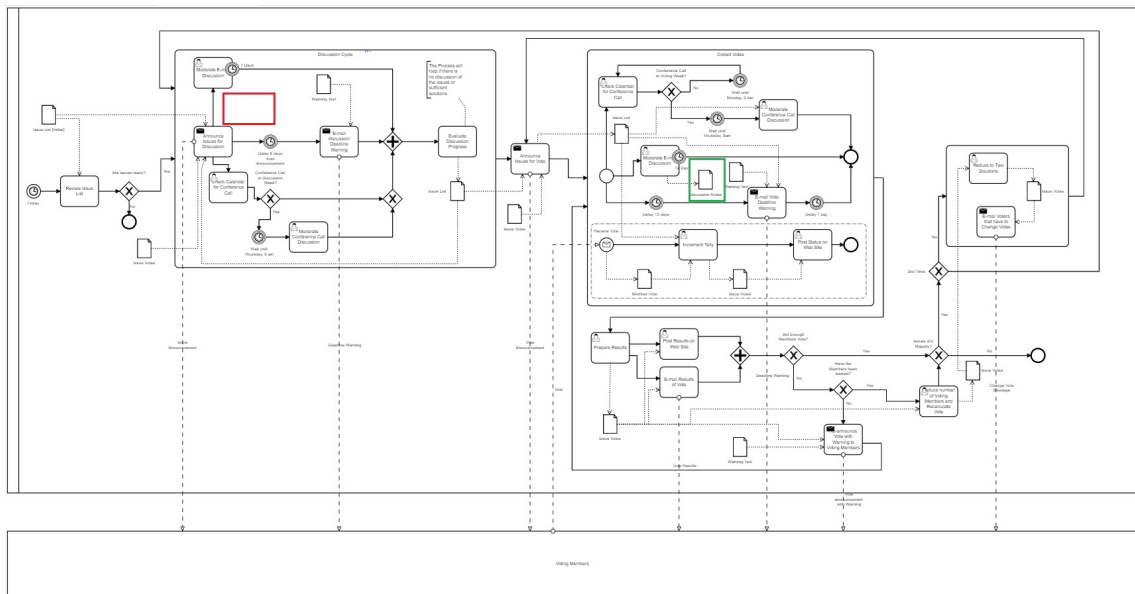


Figure 6.10: Email voting business process updated model

6.3.2 Result

The result of the comparison of the original model with the updated one, returning a list of changed artifacts. The first changed element is the "Discussion Notes" artifact inserted on the updated model. This artifact contains a direct data output association with the activity "Moderate E-mail Discussion". The removed artifact "Issue List" has a data output association from activity "Announce Issues for Discussion" and three data input associations to the followings activities: "Moderate Conference Call Discussion", "E-mail discussion Deadline Warning," and "Evaluate Discussion Progress". Figure 6.11 present the list of changed artifacts.

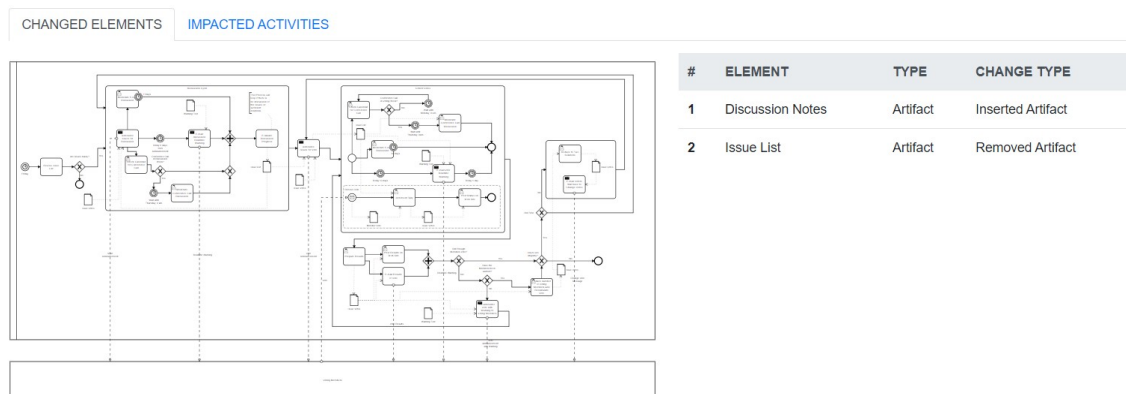


Figure 6.11: Email voting changed elements

The change artifacts originated on the updated model, produce a set of impacted activities, as we can see in figure 6.12.

#	IMPACTED ACTIVITY	REASON
1	Moderate E-mail Discussion	Inserted Artifact (Discussion Notes)
2	Moderate Conference Call Discussion	Removed Artifact (Issue List)
3	Evaluate Discussion Progress	Removed Artifact (Issue List)
4	E-mail discussion Deadline Warning	Removed Artifact (Issue List)
5	Announce Issues for Discussion	Removed Artifact (Issue List)

Figure 6.12: Email voting impacted activities

6.4 Summary

The previous examples serve as the bases to validate the thesis's developed tool framework to analyze BPMN changes' impacts. Each example includes its complexity and a scenario for testing a set of change patterns differences. The first example describes the activities differences change patterns, while the second and third examples present the gateways and artifacts change differences, respectively. The result of each performed test experience is aligned with the expectation. So, we can assume that the tool developed during this thesis' work can analyze and compare BPMN models and show impact analysis results according to the changed elements results.

Chapter 7

Conclusions and Future Work

This chapter addresses the final considerations and concluding of this work and describes the difficulties in taken the thesis and future works related to the change patterns catalog.

7.1 Final Considerations

The business process model and notation define a set of criteria that supports creating an abstract view of a business process. It allows us to store information about a business process, and nowadays, it is easy to find on the web software to create a BPMN model. This thesis's principal goal was to create a tool that allows the automation of the process of change impact analysis based on a business process model and notation.

We start working on this thesis by searching about background concepts and state-of-the-art change impact analysis and business processes. Based on the information and acknowledgment acquired, we develop a meta-model that defines the execution steps taken to perform the change impact analysis based on a BPMN model. The meta-model serves as the implementation tool's base that automates all the steps defined to calculate the impact analysis information. The platform implements some of the catalog patterns that allow us to divide the types of changes in different categories based on elements types and change actions.

In the first moment, we considered implementing just the change patterns related to the activities, gateways, and artifacts. These patterns are not so complicated to implement but with a high level of difficulty that makes the work exciting and challenging. The development of the tool follows an interactive and modular approach during the academic year. This tool allows comparing two BPMN model files, visualizing their changed

elements, and impacting activities information based on the number of N's steps.

In the end, the tool's validation, and consequently, the thesis was done based on different cases of examples. Each of these examples tests a different set of change patterns implemented on the tool. We verified that the platform has the expected behavior for the model examples used during the test, showing the impact analysis information.

7.2 Difficulties

The main challenges faced in the development of this thesis were defining the meta-model of the approach to be followed in change impact analysis. After this definition, find a way to read a BPMN file's content so that we have all the elements, and their relationship with each other takes some time.

In general, this project's realization was challenging; however, we always see it as a goal to be met since the beginning. In this case, to accomplish this objective, we start by researching change impact analysis and business process model topics. Because of the large amount of data available for those topics we do searching for, sometimes we lose our focus, trying to add information that is not essential to this study. In the end, the result of the developed tool was as expected, so that we considered the development of the works done has a positive result.

7.3 Further Work

Apart from work done until now, the tool developed during this thesis's development implements part of the change patterns present in the catalog described in chapter 3, so it needs future work. Given that this catalog corresponds to the first iteration of change patterns catalog related to the process of change impact analysis, future works can be infinite as new iterations will undoubtedly refine existing change patterns and add new ones. The future works can be starting by implementing the authorization differences change pattern. That is the only category in the catalog without any pattern implemented in the tool's current version.

One of the features to be added to the platform in future work is exporting the analysis's result to PDF. In this case, the user can get hands-on results from the change impact

analysis and share it with other users. The way the results are present can also be improved to be more readable and understandable to the end-user. Other components that can be added to the platform are user profile and BPMN files management. By introducing the user profile in the tool, each user will have its workspace where can be used to upload BPMN files and to manage them. The user can also save the result of the change impact analysis for future consultation.

References

- [1] H O Ali, M Z A Rozan, and A M Sharif. Identifying challenges of change impact analysis for software projects. In *2012 International Conference on Innovation Management and Technology Research*, pages 407–411, 2012.
- [2] R S Arnold and S A Bohner. Impact analysis-Towards a framework for comparison. In *1993 Conference on Software Maintenance*, pages 292–301, 1993.
- [3] Robert S Arnold. *Software Change Impact Analysis*. IEEE Computer Society Press, Los Alamitos, CA, USA, 1996.
- [4] L C Briand, Y Labiche, and L O’Sullivan. Impact Analysis and Change Management of UML Models. In *Proceedings of the International Conference on Software Maintenance, ICSM ’03*, pages 256—, Washington, DC, USA, 2003. IEEE Computer Society.
- [5] Lionel Briand, Yvan Labiche, L O’Sullivan, and M M Sówka. Automated impact analysis of UML models. *Journal of Systems and Software*, 79:339–352, 2006.
- [6] Remco Dijkman. A Classification of Differences between Similar BusinessProcesses. In *Proceedings - IEEE International Enterprise Distributed Object Computing Workshop, EDOC*, page 37, 2007.
- [7] Chetna Gupta, Maneesha Srivastav, and Varun Gupta. Software change impact analysis: an approach to differentiate type of change to minimise regression test selection. *International Journal of Computer Applications in Technology*, 51:366, 2015.
- [8] M HAMMER. REENGINEERING WORK - DONT AUTOMATE, OBLITERATE. *HARVARD BUSINESS REVIEW*, 68(4):104–112, 1990.
- [9] Mohd Nazri Kama. Extending change impact analysis approach for change effort estimation in the software development phase. 2013.
- [10] John Krogstie. Perspectives to Process Modeling – A Historical Overview. In Iliia Bider, Terry Halpin, John Krogstie, Selmin Nurcan, Erik Proper, Rainer Schmidt, Pnina Soffer, and Stanisław Wrycza, editors, *Enterprise, Business-Process and Information Systems Modeling*, pages 315–330, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

- [11] Steffen Lehnert. A Taxonomy for Software Change Impact Analysis. In *Proceedings of the 12th International Workshop on Principles of Software Evolution and the 7th Annual ERCIM Workshop on Software Evolution, IWPSE-EVOL '11*, pages 41–50, New York, NY, USA, 2011. ACM.
- [12] Kjetil Moløkken-Østvold and M Jorgensen. A review of surveys on software effort estimation. pages 223–230, 2003.
- [13] OMG. Object Management Group Business, Process Model and Notation.
- [14] Ana C R Paiva, Nuno H Flores, João P Faria, and José M G Marques. End-to-end Automatic Business Process Validation. *Procedia Computer Science*, 130:999–1004, 2018.
- [15] Macedo de Morais Rinaldo. An analysis of BPM lifecycles: from a literature review to a framework proposal. *Business Process Management Journal*, 20(3):412–432, jan 2014.
- [16] Tobias Roeser and Eva-Maria Kern. Surveys in business process management – a literature review. *Business Process Management Journal*, 21:692–718, 2015.
- [17] Stephen A. White. *Business Process Modeling Notation*. 2004.
- [18] Watcharin Uronkarn and Twittie Senivongse. Pattern-driven traceability of business processes. *Lecture Notes in Engineering and Computer Science*, 2209:601–606, 2014.
- [19] Watcharin Uronkarn and Twittie Senivongse. *Change Patterns Detection and Traceability Impact Analysis of Business Process Models*, pages 441–455. 2015.
- [20] W Wang, H Ding, J Dong, and C Ren. A Comparison of Business Process Modeling Methods. In *2006 IEEE International Conference on Service Operations and Logistics, and Informatics*, pages 1136–1141, 2006.
- [21] Yi Wang, Jian Yang, and Weiliang Zhao. Change impact analysis in service-based business processes. In *Service Oriented Computing and Applications*, volume 6, pages 1–8, 2011.
- [22] Piotr Wiśniewski. Decomposition of business process models into reusable sub-diagrams. *ITM Web of Conferences*, 15:1002, 2017.
- [23] Hua Xiao, Jin Guo, and Ying Zou. Supporting Change Impact Analysis for Service Oriented Business Applications. In *Proceedings of the International Workshop on Systems Development in SOA Environments, SDSOA '07*, pages 6—, Washington, DC, USA, 2007. IEEE Computer Society.

Appendix A

BPMN Model Code Snippets

Find bellow the xml code snipptes version of the BPMN models used on example 3 that demonstrate the usage and validation of the tool.

A.1 Hardware retail example - First Model

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <semantic:definitions xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance
   " xmlns:di="http://www.omg.org/spec/DD/20100524/DI" xmlns:bpmndi="http:
   //www.omg.org/spec/BPMN/20100524/DI" xmlns:dc="http://www.omg.org/spec/
   DD/20100524/DC" xmlns:semantic="http://www.omg.org/spec/BPMN/20100524/
   MODEL" id="_1275500364427" targetNamespace="http://www.trisotech.com/
   definitions/_1275500364427" exporter="Camunda Modeler" exporterVersion=
   "3.7.1">
3 <semantic:process id="_6-1" isExecutable="false">
4   <semantic:laneSet id="ls_6-53">
5     <semantic:lane id="_6-65" name="Warehouse Worker">
6       <semantic:flowNodeRef>_6-547</semantic:flowNodeRef>
7       <semantic:flowNodeRef>_6-569</semantic:flowNodeRef>
8       <semantic:flowNodeRef>_6-620</semantic:flowNodeRef>
9       <semantic:flowNodeRef>Activity_0gl0kgm</semantic:flowNodeRef>
10      <semantic:flowNodeRef>_6-153</semantic:flowNodeRef>
11    </semantic:lane>
12    <semantic:lane id="_6-63" name="Clerk">
13      <semantic:flowNodeRef>_6-67</semantic:flowNodeRef>
14      <semantic:flowNodeRef>_6-80</semantic:flowNodeRef>
15      <semantic:flowNodeRef>_6-102</semantic:flowNodeRef>
16      <semantic:flowNodeRef>_6-204</semantic:flowNodeRef>
17      <semantic:flowNodeRef>_6-277</semantic:flowNodeRef>
```

```

18     <semantic:flowNodeRef>_6-328</semantic:flowNodeRef>
19     <semantic:flowNodeRef>_6-379</semantic:flowNodeRef>
20     <semantic:flowNodeRef>_6-430</semantic:flowNodeRef>
21     <semantic:flowNodeRef>_6-503</semantic:flowNodeRef>
22     <semantic:flowNodeRef>_6-525</semantic:flowNodeRef>
23 </semantic:lane>
24 <semantic:lane id="_6-61" name="Logistics Manager">
25     <semantic:flowNodeRef>_6-452</semantic:flowNodeRef>
26 </semantic:lane>
27 </semantic:laneSet>
28 <semantic:task id="_6-452" name="Take out extra insurance">
29     <semantic:incoming>_6-645</semantic:incoming>
30     <semantic:outgoing>_6-651</semantic:outgoing>
31 </semantic:task>
32 <semantic:startEvent id="_6-67" name="Goods to ship">
33     <semantic:outgoing>_6-633</semantic:outgoing>
34 </semantic:startEvent>
35 <semantic:parallelGateway id="_6-80" name="">
36     <semantic:incoming>_6-633</semantic:incoming>
37     <semantic:outgoing>_6-635</semantic:outgoing>
38     <semantic:outgoing>Flow_1wbbjdh</semantic:outgoing>
39 </semantic:parallelGateway>
40 <semantic:task id="_6-102" name="Decide if regular post or particular
    shipment">
41     <semantic:incoming>_6-635</semantic:incoming>
42     <semantic:outgoing>_6-637</semantic:outgoing>
43 </semantic:task>
44 <semantic:exclusiveGateway id="_6-204" name="Mode of delivery">
45     <semantic:incoming>_6-637</semantic:incoming>
46     <semantic:outgoing>_6-639</semantic:outgoing>
47     <semantic:outgoing>_6-641</semantic:outgoing>
48 </semantic:exclusiveGateway>
49 <semantic:task id="_6-277" name="Assign a carrier &#38; prepare
    paperwork">
50     <semantic:incoming>_6-639</semantic:incoming>
51     <semantic:outgoing>_6-657</semantic:outgoing>
52 </semantic:task>
53 <semantic:task id="_6-328" name="Fill in a Post label">
54     <semantic:incoming>_6-647</semantic:incoming>
55     <semantic:outgoing>_6-649</semantic:outgoing>
56 </semantic:task>
57 <semantic:task id="_6-379" name="Check if extra insurance is necessary
    ">
58     <semantic:incoming>_6-641</semantic:incoming>

```

```
59     <semantic:outgoing>_6-643</semantic:outgoing>
60 </semantic:task>
61 <semantic:inclusiveGateway id="_6-430" name="">
62     <semantic:incoming>_6-643</semantic:incoming>
63     <semantic:outgoing>_6-645</semantic:outgoing>
64     <semantic:outgoing>_6-647</semantic:outgoing>
65 </semantic:inclusiveGateway>
66 <semantic:inclusiveGateway id="_6-503" name="">
67     <semantic:incoming>_6-649</semantic:incoming>
68     <semantic:incoming>_6-651</semantic:incoming>
69     <semantic:outgoing>_6-653</semantic:outgoing>
70 </semantic:inclusiveGateway>
71 <semantic:exclusiveGateway id="_6-525" name="">
72     <semantic:incoming>_6-653</semantic:incoming>
73     <semantic:incoming>_6-657</semantic:incoming>
74     <semantic:outgoing>_6-659</semantic:outgoing>
75 </semantic:exclusiveGateway>
76 <semantic:parallelGateway id="_6-547" name="">
77     <semantic:incoming>_6-659</semantic:incoming>
78     <semantic:incoming>_6-661</semantic:incoming>
79     <semantic:outgoing>_6-663</semantic:outgoing>
80 </semantic:parallelGateway>
81 <semantic:task id="_6-569" name="Add paperwork and move package to
      pick area">
82     <semantic:incoming>_6-663</semantic:incoming>
83     <semantic:outgoing>_6-665</semantic:outgoing>
84 </semantic:task>
85 <semantic:endEvent id="_6-620" name="Goods available for pick">
86     <semantic:incoming>_6-665</semantic:incoming>
87 </semantic:endEvent>
88 <semantic:task id="Activity_0gl0kgm" name="Sanitize goods">
89     <semantic:incoming>Flow_1wbbjdh</semantic:incoming>
90     <semantic:outgoing>Flow_0v69pae</semantic:outgoing>
91 </semantic:task>
92 <semantic:task id="_6-153" name="Package goods">
93     <semantic:incoming>Flow_0v69pae</semantic:incoming>
94     <semantic:outgoing>_6-661</semantic:outgoing>
95 </semantic:task>
96 <semantic:sequenceFlow id="_6-665" name="" sourceRef="_6-569"
      targetRef="_6-620" />
97 <semantic:sequenceFlow id="_6-663" name="" sourceRef="_6-547"
      targetRef="_6-569" />
98 <semantic:sequenceFlow id="_6-661" name="" sourceRef="_6-153"
      targetRef="_6-547" />
```

```

99     <semantic:sequenceFlow id="_6-659" name="" sourceRef="_6-525"
100       targetRef="_6-547" />
101     <semantic:sequenceFlow id="_6-657" name="" sourceRef="_6-277"
102       targetRef="_6-525" />
103     <semantic:sequenceFlow id="_6-653" name="" sourceRef="_6-503"
104       targetRef="_6-525" />
105     <semantic:sequenceFlow id="_6-651" name="" sourceRef="_6-452"
106       targetRef="_6-503" />
107     <semantic:sequenceFlow id="_6-649" name="" sourceRef="_6-328"
108       targetRef="_6-503" />
109     <semantic:sequenceFlow id="_6-647" name="Always" sourceRef="_6-430"
110       targetRef="_6-328" />
111     <semantic:sequenceFlow id="_6-645" name="extra insurance required"
112       sourceRef="_6-430" targetRef="_6-452" />
113     <semantic:sequenceFlow id="_6-643" name="" sourceRef="_6-379"
114       targetRef="_6-430" />
115     <semantic:sequenceFlow id="_6-641" name="Regular Post" sourceRef="_6-
116       -204" targetRef="_6-379" />
117     <semantic:sequenceFlow id="_6-639" name="Special Carrier" sourceRef="
118       _6-204" targetRef="_6-277" />
119     <semantic:sequenceFlow id="_6-637" name="" sourceRef="_6-102"
120       targetRef="_6-204" />
121     <semantic:sequenceFlow id="_6-635" name="" sourceRef="_6-80" targetRef
122       ="_6-102" />
123     <semantic:sequenceFlow id="_6-633" name="" sourceRef="_6-67" targetRef
124       ="_6-80" />
125     <semantic:sequenceFlow id="Flow_lwbbjdj" sourceRef="_6-80" targetRef="
126       Activity_0gl0kgm" />
127     <semantic:sequenceFlow id="Flow_0v69pae" sourceRef="Activity_0gl0kgm"
128       targetRef="_6-153" />
129     <semantic:textAnnotation id="_6-670">
130       <semantic:text>Insurance is included in carrier service</
131         semantic:text>
132     </semantic:textAnnotation>
133 </semantic:process>
134 <semantic:collaboration id="C1275500365927">
135   <semantic:participant id="_6-53" name="Hardware Retailer" processRef="
136     _6-1" />
137 </semantic:collaboration>

```

Listing A.1: Hardware retail example - First Model

A.2 Hardware retail example - Update Model

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <semantic:definitions xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance
   " xmlns:di="http://www.omg.org/spec/DD/20100524/DI" xmlns:bpmndi="http://www.omg.org/spec/BPMN/20100524/DI" xmlns:dc="http://www.omg.org/spec/DD/20100524/DC" xmlns:semantic="http://www.omg.org/spec/BPMN/20100524/MODEL" id="_1275500364427" targetNamespace="http://www.trisotech.com/definitions/_1275500364427" exporter="Camunda Modeler" exporterVersion="3.7.1">
3   <semantic:process id="_6-1" isExecutable="false">
4     <semantic:laneSet id="ls_6-53">
5       <semantic:lane id="_6-65" name="Warehouse Worker">
6         <semantic:flowNodeRef>_6-547</semantic:flowNodeRef>
7         <semantic:flowNodeRef>_6-569</semantic:flowNodeRef>
8         <semantic:flowNodeRef>_6-620</semantic:flowNodeRef>
9         <semantic:flowNodeRef>Activity_0gl0kgm</semantic:flowNodeRef>
10        <semantic:flowNodeRef>_6-153</semantic:flowNodeRef>
11      </semantic:lane>
12      <semantic:lane id="_6-63" name="Clerk">
13        <semantic:flowNodeRef>_6-67</semantic:flowNodeRef>
14        <semantic:flowNodeRef>_6-80</semantic:flowNodeRef>
15        <semantic:flowNodeRef>_6-102</semantic:flowNodeRef>
16        <semantic:flowNodeRef>_6-204</semantic:flowNodeRef>
17        <semantic:flowNodeRef>_6-277</semantic:flowNodeRef>
18        <semantic:flowNodeRef>_6-328</semantic:flowNodeRef>
19        <semantic:flowNodeRef>_6-379</semantic:flowNodeRef>
20        <semantic:flowNodeRef>_6-430</semantic:flowNodeRef>
21        <semantic:flowNodeRef>_6-503</semantic:flowNodeRef>
22        <semantic:flowNodeRef>_6-525</semantic:flowNodeRef>
23      </semantic:lane>
24      <semantic:lane id="_6-61" name="Logistics Manager">
25        <semantic:flowNodeRef>_6-452</semantic:flowNodeRef>
26      </semantic:lane>
27    </semantic:laneSet>
28    <semantic:task id="_6-452" name="Take out extra insurance">
29      <semantic:incoming>_6-645</semantic:incoming>
30      <semantic:outgoing>_6-651</semantic:outgoing>
31    </semantic:task>
32    <semantic:startEvent id="_6-67" name="Goods to ship">
33      <semantic:outgoing>_6-633</semantic:outgoing>
34    </semantic:startEvent>
35    <semantic:parallelGateway id="_6-80" name="">
```

```

36     <semantic:incoming>_6-633</semantic:incoming>
37     <semantic:outgoing>_6-635</semantic:outgoing>
38     <semantic:outgoing>Flow_1wbbjdh</semantic:outgoing>
39 </semantic:parallelGateway>
40 <semantic:task id="_6-102" name="Decide if regular post or particular
    shipment">
41     <semantic:incoming>_6-635</semantic:incoming>
42     <semantic:outgoing>_6-637</semantic:outgoing>
43 </semantic:task>
44 <semantic:exclusiveGateway id="_6-204" name="Mode of delivery">
45     <semantic:incoming>_6-637</semantic:incoming>
46     <semantic:outgoing>_6-639</semantic:outgoing>
47     <semantic:outgoing>_6-641</semantic:outgoing>
48 </semantic:exclusiveGateway>
49 <semantic:task id="_6-277" name="Assign a carrier &#38; prepare
    paperwork">
50     <semantic:incoming>_6-639</semantic:incoming>
51     <semantic:outgoing>_6-657</semantic:outgoing>
52 </semantic:task>
53 <semantic:task id="_6-328" name="Fill in a Post label">
54     <semantic:incoming>_6-647</semantic:incoming>
55     <semantic:outgoing>_6-649</semantic:outgoing>
56 </semantic:task>
57 <semantic:task id="_6-379" name="Check if extra insurance is necessary
    ">
58     <semantic:incoming>_6-641</semantic:incoming>
59     <semantic:outgoing>_6-643</semantic:outgoing>
60 </semantic:task>
61 <semantic:inclusiveGateway id="_6-430" name="">
62     <semantic:incoming>_6-643</semantic:incoming>
63     <semantic:outgoing>_6-645</semantic:outgoing>
64     <semantic:outgoing>_6-647</semantic:outgoing>
65 </semantic:inclusiveGateway>
66 <semantic:inclusiveGateway id="_6-503" name="">
67     <semantic:incoming>_6-649</semantic:incoming>
68     <semantic:incoming>_6-651</semantic:incoming>
69     <semantic:outgoing>_6-653</semantic:outgoing>
70 </semantic:inclusiveGateway>
71 <semantic:exclusiveGateway id="_6-525" name="">
72     <semantic:incoming>_6-653</semantic:incoming>
73     <semantic:incoming>_6-657</semantic:incoming>
74     <semantic:outgoing>_6-659</semantic:outgoing>
75 </semantic:exclusiveGateway>
76 <semantic:parallelGateway id="_6-547" name="">

```

```
77     <semantic:incoming>_6-659</semantic:incoming>
78     <semantic:incoming>_6-661</semantic:incoming>
79     <semantic:outgoing>_6-663</semantic:outgoing>
80 </semantic:parallelGateway>
81 <semantic:task id="_6-569" name="Add paperwork and move package to
      pick area">
82     <semantic:incoming>_6-663</semantic:incoming>
83     <semantic:outgoing>_6-665</semantic:outgoing>
84 </semantic:task>
85 <semantic:endEvent id="_6-620" name="Goods available for pick">
86     <semantic:incoming>_6-665</semantic:incoming>
87 </semantic:endEvent>
88 <semantic:task id="Activity_0gl0kgm" name="Sanitize goods">
89     <semantic:incoming>Flow_1wbbjdh</semantic:incoming>
90     <semantic:outgoing>Flow_0v69pae</semantic:outgoing>
91 </semantic:task>
92 <semantic:task id="_6-153" name="Package goods">
93     <semantic:incoming>Flow_0v69pae</semantic:incoming>
94     <semantic:outgoing>_6-661</semantic:outgoing>
95 </semantic:task>
96 <semantic:sequenceFlow id="_6-665" name="" sourceRef="_6-569"
      targetRef="_6-620" />
97 <semantic:sequenceFlow id="_6-663" name="" sourceRef="_6-547"
      targetRef="_6-569" />
98 <semantic:sequenceFlow id="_6-661" name="" sourceRef="_6-153"
      targetRef="_6-547" />
99 <semantic:sequenceFlow id="_6-659" name="" sourceRef="_6-525"
      targetRef="_6-547" />
100 <semantic:sequenceFlow id="_6-657" name="" sourceRef="_6-277"
      targetRef="_6-525" />
101 <semantic:sequenceFlow id="_6-653" name="" sourceRef="_6-503"
      targetRef="_6-525" />
102 <semantic:sequenceFlow id="_6-651" name="" sourceRef="_6-452"
      targetRef="_6-503" />
103 <semantic:sequenceFlow id="_6-649" name="" sourceRef="_6-328"
      targetRef="_6-503" />
104 <semantic:sequenceFlow id="_6-647" name="Always" sourceRef="_6-430"
      targetRef="_6-328" />
105 <semantic:sequenceFlow id="_6-645" name="extra insurance required"
      sourceRef="_6-430" targetRef="_6-452" />
106 <semantic:sequenceFlow id="_6-643" name="" sourceRef="_6-379"
      targetRef="_6-430" />
107 <semantic:sequenceFlow id="_6-641" name="Regular Post" sourceRef="_6
      -204" targetRef="_6-379" />
```



```
108 <semantic:sequenceFlow id="_6-639" name="Special Carrier" sourceRef="
    _6-204" targetRef="_6-277" />
109 <semantic:sequenceFlow id="_6-637" name="" sourceRef="_6-102"
    targetRef="_6-204" />
110 <semantic:sequenceFlow id="_6-635" name="" sourceRef="_6-80" targetRef
    ="_6-102" />
111 <semantic:sequenceFlow id="_6-633" name="" sourceRef="_6-67" targetRef
    ="_6-80" />
112 <semantic:sequenceFlow id="Flow_1wbbjdh" sourceRef="_6-80" targetRef="
    Activity_0gl0kgm" />
113 <semantic:sequenceFlow id="Flow_0v69pae" sourceRef="Activity_0gl0kgm"
    targetRef="_6-153" />
114 <semantic:textAnnotation id="_6-670">
115 <semantic:text>Insurance is included in carrier service</
    semantic:text>
116 </semantic:textAnnotation>
117 </semantic:process>
118 <semantic:collaboration id="C1275500365927">
119 <semantic:participant id="_6-53" name="Hardware Retailer" processRef="
    _6-1" />
120 </semantic:collaboration>
```

Listing A.2: Hardware retail example - Update Model