TÉCNICO LISBOA

View Integration of Business Process Models

João Pedro António Colaço

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisor: Prof. Pedro Manuel Moreira Vaz Antunes de Sousa

Examination Committee

Chairperson: Prof. Daniel Jorge Viegas Gonçalves

Supervisor: Prof. Pedro Manuel Moreira Vaz Antunes de Sousa

Member of the Committee: Prof. José Luís Brinquete Borbinha

November 2017

Acknowledgments

This dissertation marks the closure of the best five years of my life as a student of this university. I want to thank all the brothers, sisters, family, friends and colleagues who were part of this journey.

I want to express my gratitude to my supervisor, Professor Pedro Sousa, who was able to push me to work further towards the objective, by discussing and clarifying ideas regarding this project. I also want to thank my colleague Diogo Cardoso for helping me in several matters related with this work.

To the brothers and sisters with whom I shared the most beautiful moments throughout this journey. Bernardo Figueiredo, Catarina Gonçalves, Alexandre Candeias, Ismael Belchior, André Alves and Jonathan Calvillo, I am privileged to have met you. Even when distance separates us, each one of us knows that we can always count on each other.

A special word of esteem to all the members of the A2 floor family, spread around the world, for making this journey funnier and full of unforgettable moments. I must mention a few: João Rodrigues for the priceless advice and feedback regarding my scientific writing; Rui Jaulino for teaching the kids about life and for the inspiring conversations; Tiago Barros for his perseverance and carisma; Helena Contreiras and Joana Duarte for trying to keep the kitchen conversations on track; Hugo Ribeiro for the portuguese grammar and spelling lessons, for his crucial influence on my decisions and for the long interesting conversations; Clara Pereira, Alexandre Moita and Luís Alves for their friendship. I must also thank Marielena De Jesús and Andrea Valentina Chacon for the crazy moments and laughs that we shared.

To the friends of always, José Sousa, Henrique Carvalho, João Fernandes and Bruna Pires. Thank you for being part of my life.

I want to thank Instituto Superior Técnico and Núcleo de Alojamentos for making the residence Eng.º Duarte Pacheco available every day.

A final word of appraisal and love to my parents, for making this journey possible, for supporting all my decisions and for providing me the perfect environment for finishing this dissertation. Thank you for being an example on how to break the walls of your starting environment by always aiming for better, doing your best, every day.

Abstract

Stakeholders have distinct concerns regarding the same business process. Therefore, organizations tend to produce multiple process models in order to address the various stakeholder concerns. These models often lack consistency, which poses challenges to organizations regarding the management of such inconsistencies among the multiple business process models. Furthermore, the current business process modeling languages lack a way for the stakeholders to express their concerns towards a given process.

To cope with these issues, it is proposed a view integration process to obtain a consolidated model, with the support of a repository and an organizational taxonomy. The proposed solution is based on a multi-dimensional definition of business process and applies view integration to business process modeling through an incremental approach.

The impact of such proposal was demonstrated by applying the solution artifact to two distinct scenarios. Its practical use was demonstrated by developing a process modeling tool prototype and applying it to one of the scenarios. The outcome of the proposed view integration process is the generation of a consolidated model, which in turn enables the generation of different views. The proposed solution artifact was evaluated regarding its efficacy and utility.

Keywords: Business process modeling, View integration, Repository, Enterprise architecture

Resumo

Os vários *stakeholders* têm interesses distintos relativamente ao mesmo processo de negócio. Assim, as organizações tendem a produzir múltiplos modelos, com o objetivo de corresponder aos interesses distintos dos *stakeholders*. Estes modelos frequentemente apresentam inconsistências, o que coloca desafios às organizações relativamente à gestão das mesmas entre os vários modelos de processos de negócio. Além disso, as linguagens de modelação de processos de negócio atuais não possuem uma forma de os *stakeholders* expressarem os seus interesses em relação a um determinado processo.

Para lidar com estes problemas, é proposto um processo de integração de vistas para a obtenção de um modelo consolidado, com a ajuda de um repositório e de uma taxonomia organizacional. A solução proposta basea-se numa definição multidimensional de processo de negócio, onde a integração de vistas é aplicada na modelação de processos de negócios através de uma abordagem incremental.

O impacto da proposta foi demonstrada aplicando a solução a dois cenários distintos. A sua utilização prática foi demonstrada desenvolvendo um protótipo de uma ferramenta de modelação de processos e aplicando-o a um dos cenários. O processo de integração de vistas proposto tem como resultado a geração de um modelo consolidado, que por sua vez permite a geração de diferentes vistas. O artefacto da solução proposta foi avaliado quanto à sua eficácia e utilidade.

Keywords: Modelação de processos de negócio, Integração de vistas, Repositório, Arquitectura empresarial

Contents

1 Introduction **1**
- 1.1 Motivation . . . 1
- 1.2 Problem Description . . . 2
- 1.3 Research Methodology . . . 5
- 1.4 Document Structure . . . 5

2 Background **7**
- 2.1 Business Process Management . . . 7
  - 2.1.1 Historical Overview . . . 7
  - 2.1.2 The BPM Lifecycle . . . 8
  - 2.1.3 Process Architecture . . . 9
- 2.2 Business Process Modeling . . . 10
- 2.3 Enterprise Architecture . . . 13
- 2.4 Enterprise Architecture Frameworks . . . 14
  - 2.4.1 The Zachman Framework . . . 14
  - 2.4.2 The Open Group Architectural Framework (TOGAF) . . . 15
  - 2.4.3 Model-Driven Architecture (MDA) . . . 17

3 Related Work **19**
- 3.1 View Integration . . . 19
- 3.2 The Zachman Framework Dimensions and Business Process Modeling . . . 22
- 3.3 Analysis . . . 24

4 Proposed Solution **27**
- 4.1 Repository Metamodel . . . 27
- 4.2 View Integration Process . . . 29
- 4.3 View Elements Classification . . . 30

5 Demonstration **35**
- 5.1 Scenarios of Application . . . 35
  - 5.1.1 Scenario 1 . . . 35
  - 5.1.2 Scenario 2 . . . 37
- 5.2 Tool Support Example . . . 41
- 5.3 Discussion . . . 42
- 5.4 DSR Evaluation . . . 45

6 Conclusion **49**
- 6.1 Contributions . . . 49

6.2 Limitations 50
6.3 Future Work 50

Bibliography **51**

List of Figures

1.1 Logistics and HR teams' concerns regarding the car repair process. 2
1.2 Logistics team car repair process view. 3
1.3 Human Resources (HR) team car repair process view. 3
1.4 Consolidated model that meets the concerns of both teams. 4
1.5 DSRM process steps mapped to the phases of this research work (adapted from [27]). . . 6

2.1 The BPM lifecycle (from [11, p. 21]). 8
2.2 Levels of detail in a process architecture (from [11, p.42]). 10
2.3 Elements of the Petri Net notation (a) and an example of a Petri net (b). 11
2.4 Elements of the EPC notation (a) and an example of EPC (b). 12
2.5 Overview of the BPMN core elements (adapted from [15]). 13
2.6 The enterprise architecture as a tool for enterprise management (from [18]). 14
2.7 The Zachman Framework (from [18]). 15
2.8 TOGAF 9.1 (from [18]). 16
2.9 TOGAF Architecture Development Cycle (from [35]). 17
2.10 Model-Driven Architecture framework (from [18]). 18

3.1 Conceptual Framework for Database Design (adapted from [22, 23]). 20
3.2 View Integration Model (from [22]). 21
3.3 Taxonomy example of the *who* dimension (from [29]). 23
3.4 Definition of business process used in this thesis (from [30]). 24

4.1 Business process repository metamodel. 28
4.2 BPMN diagram representing the view integration process. 30

5.1 State of the repository after uploading and classifying the logistics team view. 36
5.2 State of the repository after uploading and classifying the HR team view. 37
5.3 The Departures Manager view of the passengers departure process. 38
5.4 The Passenger Support Manager view of the passengers departure process. 39
5.5 Taxonomy resultant from applying the view integration process to the Departures Manager view. 40
5.6 Consolidated taxonomy after classifying the Passenger Support Manager view. 40
5.7 The consolidated model, specified in Business Process Model and Notation (BPMN) 2.0, resultant from the integration of the departure process views. 41
5.8 Initial state of the taxonomy (a) and the modeling interface (b). 42
5.9 Choosing the process (a) and pool classification (b). 43
5.10 Lanes (a) and composite activity (b) classification. 44
5.11 Classification of the tasks that decompose the composite activity. 45

5.12 Classification of the last task (a) and blueprint with the final state of the taxonomy (b). . . 46

List of Tables

2.1 Dimensions of the Zachman Framework (from [33]). 15

3.1 Criteria for activity decomposition (from [33]). 22
3.2 Mapping between business process modeling notations and the Zachman Framework dimensions (adapted from [29]). 23

4.1 Mapping between the possible instances of the *LaneType* class and the corresponding dimension. 31

Acronyms

BPM Business Process Management. 1, 5, 7–9

BPMN Business Process Model and Notation. xi, 4, 5, 11, 12, 25, 27–30, 35, 41–43, 45, 47, 49, 50

BPR Business Process Redesign. 8

DSR Design Science Research. 5, 35, 47

DSRM Design Science Research Methodology. 5, 7, 19, 27, 35, 45

EA Enterprise Architecture. 5, 7, 13, 14

EMCIS European Mediterranean & Middle Eastern Conference on Information Systems. 49

EPC Event-driven Process Chain. 11, 21, 25

ER Entity-Relationship. 19, 20

HR Human Resources. xi, 1–4, 35–37

IS Information Systems. 5, 46, 47

IT Information Technology. 5, 8, 17

MDA Model Driven Architecture. 14, 17

OMG Object Management Group. 17

REST Representational State Transfer. 41

SPNet Semantic Process Net. 21, 22

TOGAF The Open Group Architecture Framework. 14–16

UML Unified Modeling Language. 17, 28

Chapter 1

Introduction

In business process design, the production of as-is and to-be process models is a crucial task in the Business Process Management (BPM) lifecycle. However, this task often lacks the necessary clarity and rigor and is performed without a defined method [28, 30, 31]. Moreover, stakeholders have distinct concerns regarding the same business process. As a result, organizations tend to create separate models for the same business process, causing consistency problems or imposing challenges for keeping the multiple models consistent [6].

The purpose of this dissertation is to contribute with a solution to aid process designers, stakeholders and organizations address the aforementioned issues. In the remaining of this chapter it is provided a description in further detail of the research motivation, followed by the problem definition, the research methodology and the structure of the present document.

1.1 Motivation

Organizations often have to manage multiple and possibly conflicting process diagrams that represent the same business process. This issue arises due to two main reasons [28–31, 33]. On the one hand, business processes usually cross multiple organizational units and also tend to cross inter-organizational boundaries. Therefore, processes are often shared among different stakeholders, which have contrasting concerns and focus on distinct perspectives of the same business process, such as performance, auditing, information systems, people or compliance. On the other hand, a business process model is a representation of the modeler's perspective regarding a given process. Thus, a business process model depends on the team that designed it, and different teams will most likely obtain different specifications for the same process.

The aforementioned issues can lead to several inconsistencies in process models, such as heterogeneous schemes for naming activities and entities, usage of different modeling styles and process hierarchies with arbitrary depth and level of detail [5]. For instance, the processing of job applications can be captured from the perspective of the HR unit, where the process model guides the work of HR employees and clarifies hand-overs; or from the perspective of the IT department, where the purpose of the model is to emphasize the interactions of HR employees with an IT-system [38].

A recent survey [17], shows significant research efforts in the past decade regarding business process variability modeling. These research works address the problem of managing families of process vari-

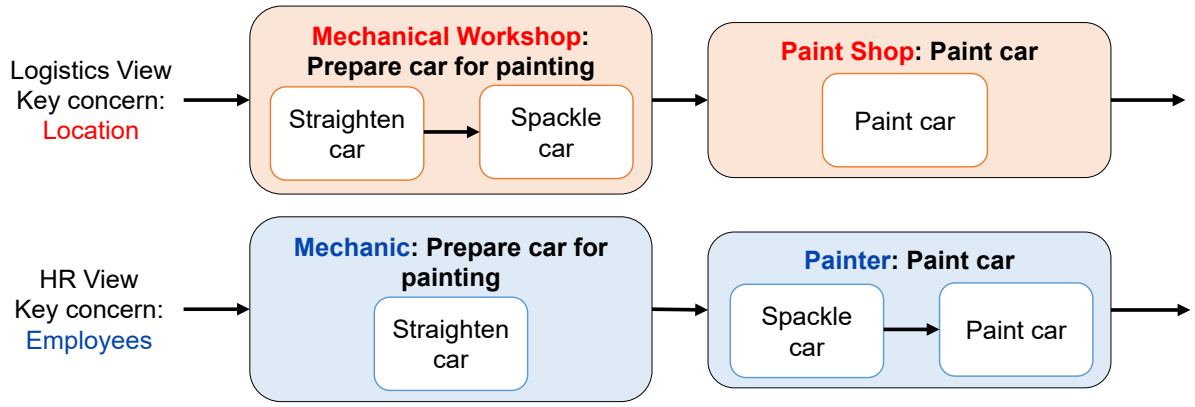


Figure 1.1: Logistics and HR teams' concerns regarding the car repair process.

ants when it is clear that the process variations exist in the "real world" (e.g. multiple sales processes for different products or multiple bookkeeping processes for different countries). However, for the aforementioned reasons, this work addresses a different problem since it is considered that variations and inconsistencies between process models are inherent to the task of business process modeling.

The problem of managing multiple diagrams of the same process should be given more importance in terms of research contributions. Furthermore, the task of business process modeling must cope with the multiple views and goals of the different organizational stakeholders, while capturing the complex relationships between information, people, goals and systems, as well as the underlying control and data flows [5].

1.2 Problem Description

The research problem will be described by presenting the illustrative example of the Auto Repair Company, with the purpose of facilitating the readers' understanding of the problem.

The Auto Repair Company example is described as the following:

> Considering the Auto Repair Company, part of the process of repairing a car body is performed by two participants: a painter and a mechanic. The repairing process starts when the damaged car arrives at the mechanical workshop. At this point, the mechanic is responsible for straightening the damaged car body panels. Once the straightening is finished, the car body panels are spackled by the painter and the car is moved to the paint shop for painting. The HR team and the logistics team are interested in modeling the car repair process. The main concern of the HR team is regarding the different process participants - *who* dimension - in order to measure their performance. The main concern of the logistics team is regarding the different locations where the activities are performed - *where* dimension - in order to measure the time participants spend changing location, during the process execution.

The concerns of both departments are illustrated in Fig. 1.1. Considering the logistics team scenario, the activity "*Prepare car for painting*" contains the tasks performed at the mechanical workshop ("*Straighten car*" and "*Spackle car*") and the "*Paint car*" activity corresponds to the task performed at the paint shop ("*Paint car*"). Considering the HR team scenario, the "*Prepare car for painting*" activity corresponds to the task performed by the mechanic ("*Straighten car*") and the "*Paint car*" activity contains the tasks performed by the painter ("*Spackle car*" and "*Paint car*").

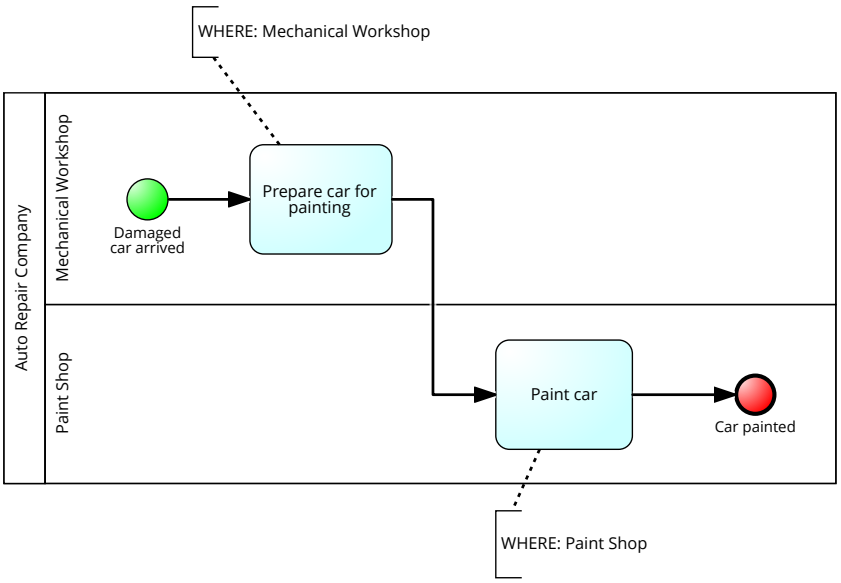


Figure 1.2: Logistics team car repair process view.

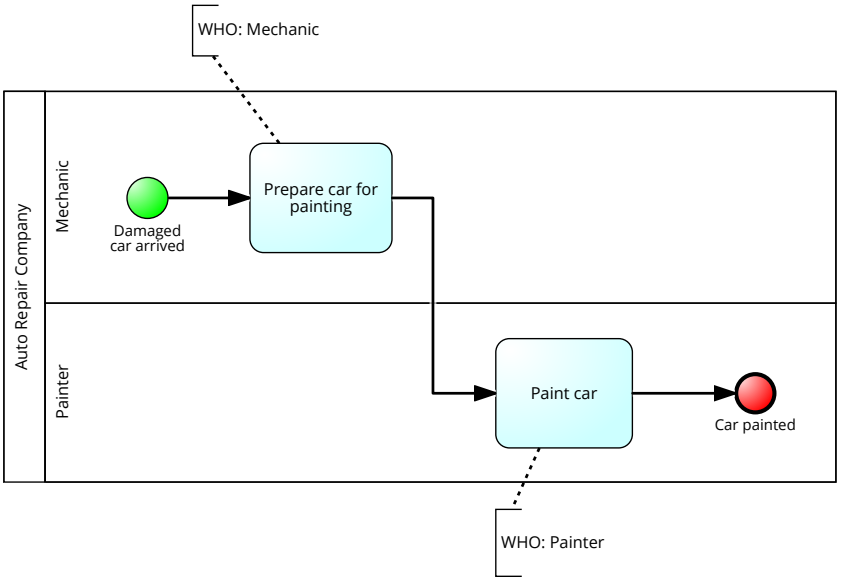


Figure 1.3: HR team car repair process view.

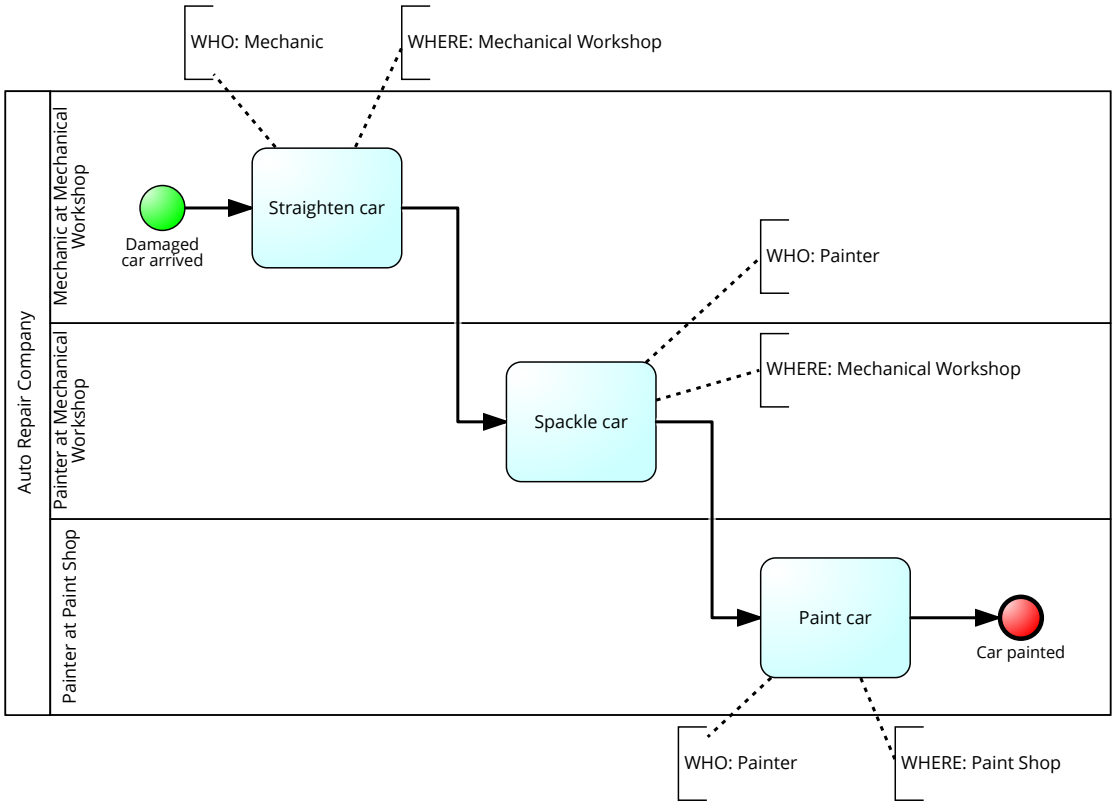


Figure 1.4: Consolidated model that meets the concerns of both teams.

The diagrams specified in BPMN 2.0 representing the views of the logistics and the HR teams are illustrated in Figs. 1.2 and 1.3, respectively. In Figure 1.2, since the *where* dimension is the main concern of the logistics department, it is used as a split criterion of the activities. In Figure 1.3, since the *who* dimension is the main concern of the HR department, it is used as a split criterion of the activities.

Considering the aforementioned process views, the question that now arises is: "How to integrate these views into a single business process model?". This integration must be achieved knowing that the resultant model must address the concerns of both teams. Thus, the resulting model must consider both splitting criteria, regarding participants (*who* dimension) and locations (*where* dimension). The goal of this proposal is to produce a consolidated model similar to the one presented in Fig. 1.4. For graphical representation and understanding reasons, the semantics of the *Lane* element ([15]) is broadened for this solution to be valid. This integration was achieved without any technique due to the simplicity of this specific example. Figure 1.4 explicitly shows that the splitting criterion of the process activities addresses the concerns of both stakeholders, since the lanes contain formal information regarding the process participants and their locations. Both *who* and *where* dimensions were merged and represented in each lane to consolidate the concerns of both stakeholders.

Given the previous example, the problem that this thesis aims to address can be defined as:

> ***The existence of heterogeneous concerns among the process stakeholders and the lack of a systematic way for representing their concerns in the task of business process modeling, lead to multiple and sometimes conflicting process models for the same business process.***

Given the problem definition, the solution objectives are the following:

1. ***To support the stakeholders in the task of business process modeling by providing a con-***

sistent and systematic way for representing their concerns.

2. ***To provide a method for the integration of distinct business process views specified in BPMN 2.0.***

1.3 Research Methodology

This thesis follows the principles and guidelines of Design Science Research (DSR) in Information Systems (IS). DSR is a problem-solving paradigm which consists of iterating over the creation and evaluation of an innovative Information Technology (IT) artifact, called design artifact, to address an important organizational problem [14]. Design artifacts are defined as constructs (vocabulary and symbols), models (abstractions and representations), methods (algorithms and practices) and instantiations (implemented and prototype systems).

This iterative approach aims to improve both the quality of the design artifact and the design process, due to the feedback provided by the evaluation step. Design-science research products must seek truth (justified theory) and utility (effective artifacts) and be evaluated regarding its practical implications [14]. Hence, a design artifact must be evaluated regarding its utility, quality and efficacy [14].

Peffers et al. proposed a methodology - the Design Science Research Methodology (DSRM) - for the production and presentation of DSR in IS, by specifying a process which is composed of six activities [27]:

- Activity 1: Problem identification and motivation. Define the problem and justify the importance of a solution.
- Activity 2: Define the objectives for a solution. Derive the objectives from the problem definition, taking into consideration what is possible and feasible.
- Activity 3: Design and development. Create the artifact after determining its desired functionality and architecture.
- Activity 4: Demonstration. Demonstrate how to use the artifact to solve one or more instances of the problem. This can be achieved through experimentation, simulation, case study, proof or other appropriate activity.
- Activity 5: Evaluation Observe and measure how well the artifact supports a solution to the problem. This involves comparing the solution's objectives with the demonstration results.
- Activity 6: Communication Communicate the problem and its importance, the artifact, its utility and novelty, the rigor of its design and its effectiveness to relevant audiences.

The mapping between the process steps and the structure of this work is presented in Fig. 1.5.

1.4 Document Structure

The remaining of this dissertation is structured as follows:

- **Chapter 2 - Background** - Provides a brief review of the state-of-the-art regarding the disciplines of BPM, Business Process Modeling, Enterprise Architecture (EA) and EA Frameworks.

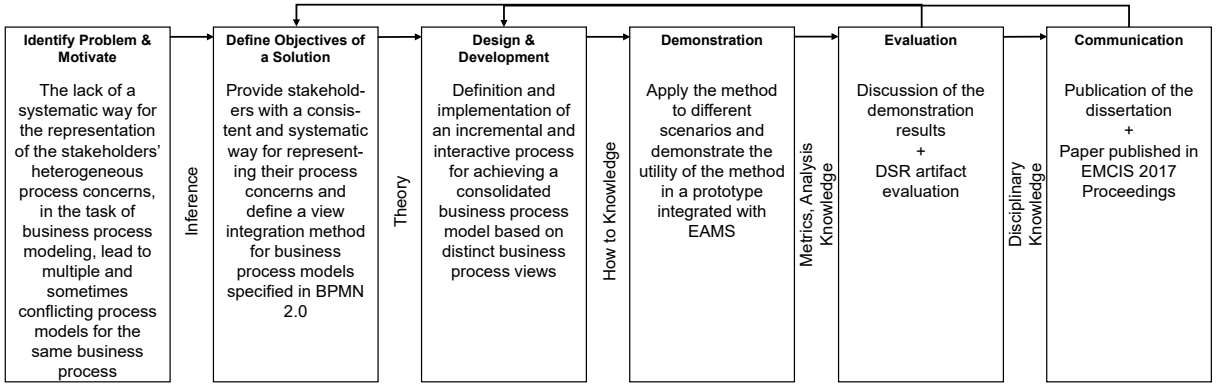


Figure 1.5: DSRM process steps mapped to the phases of this research work (adapted from [27]).

- **Chapter 3 - Related Work** - Presents an analysis of the research contributions which provide a theoretical base for the proposed solution.
- **Chapter 4 - Research Proposal** - Describes the proposed solution artifact and the elements that support it.
- **Chapter 5 - Demonstration** - Demonstrates the usage of the solution artifact in distinct scenarios and presents the discussion of the results.
- **Chapter 6 - Conclusion** - Presents an overview of the present research work, its limitations and future work.

Chapter 2

Background

This chapter supports the DSRM steps of problem identification and motivation, as well as the solution objectives definition, by presenting background knowledge for the understanding of the subsequent chapters. It starts by introducing the fields of BPM and business process modeling, then it provides an overview of EA and finishes with the description of three EA frameworks.

2.1 Business Process Management

BPM is defined by Dumas et al. as the "*art and science of overseeing how work is performed in an organization to ensure consistent outcomes and to take advantage of improvement opportunities*" ([11, p. 1]). These opportunities depend on the objectives of the organization, such as reducing costs, execution times or error rates. BPM is about managing sequences of events, activities and decisions, called *processes*, with the purpose of adding value to the organization, its customers and stakeholders [11, p. 1].

This section presents a brief historical introduction regarding BPM, followed by a description of the BPM lifecycle and the definition of process architecture.

2.1.1 Historical Overview

Regarding its historical origin, BPM did not originate in a single breakthrough event in history but emerged from the evolution over time of several ideas and improvements in organization of work. Adam Smith (1723-1790) showed the advantages of the division of labor and Frederik Taylor (1856-1915) introduced a set of principles known as scientific management, with an extreme focus in labor division [11, 36]. Taylor, by studying labor activities, developed very specific work instructions for workers, so they would only be responsible for performing one of the many steps in the production process. As a result, the concept of division of labor became the most dominant form of organizing work, since it was applied not only to the industry, but also in administrative settings such as government organizations. The advantage of this development was that workers became specialists and would be concerned with only a single part of a given business process. Taylor's ideas lead to the emergence of a new class of professionals, the *managers*, because someone needed to supervise the productivity of workers responsible for the same part of a production process [11, p. 9].

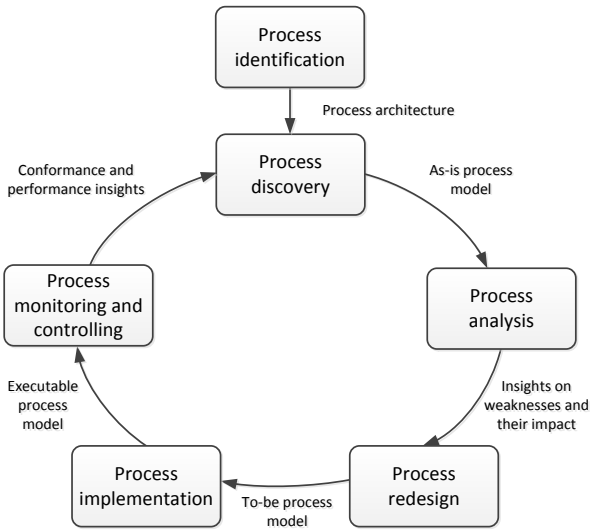


Figure 2.1: The BPM lifecycle (from [11, p. 21]).

Process thinking started with the Ford's acquisition of a big financial stake in Mazda during the 1980s. This was one of the breakthrough events for the development of BPM. This event resulted into a famous case study that deals with Ford's purchasing process. A crucial element in this case study is the approach considering the entire purchasing process to a problematic performance issue - an excessive amount of time and resources spent on checking documents in the accounts payable department [11, pp. 10-12].

The work by Davenport and Short [8], among the works of others, triggered the widespread adoption of the management concept referred as Business Process Redesign (BPR) or Business Process Reengineering. However, by the late 1990s the enthusiasm for BPR faded down, with many companies stopping the support of further BPR initiatives and projects [11, p.12]. Afterwards, two major events revived some of the BPR ideas, leading to the appearance of BPM. First, some empirical studies showed that organizations that were process-oriented did better than non-process-oriented organizations, giving renewed credibility to process thinking. Second, the technological development resulted in the emergence of different types of IT systems, like Enterprise Resource Planning systems and Workflow Management Systems. Even though BPM can be historically seen as a revival of BPR, the latter should be seen as a subset of techniques that can be used in the scope of BPM [11, pp. 12-15].

2.1.2 The BPM Lifecycle

The BPM discipline is composed of concepts, methods, techniques and tools, which are represented through the BPM lifecycle. The BPM lifecycle, illustrated in Fig. 2.1, provides a structured view of how a given process can be managed, and comprises the following phases [11, pp. 21-23]:

- *Process identification*. In this phase, a business problem is posed, processes relevant to the problem being addressed are identified, delimited and related to each other. The outcome of process identification is a new or updated process architecture that provides an overall view of the processes in an organization and their relationships (detailed in the next section). In some cases, process identification is done in parallel with performance measure identification.
- *Process discovery* or *as-is process modeling*. Here, the current state of each of the relevant processes is documented in one or several as-is process models.

- *Process analysis*. The issues related with the as-is process are identified, documented and quantified using performance measures, whenever possible. The output of this phase is a structured collection of issues, typically prioritized by impact.
- *Process redesign* or *process improvement*. The objective of this phase is to identify changes to the process that would help to address the issues identified in the previous phase. To meet the performance objectives of the organization, multiple change options are analyzed and compared in terms of the chosen performance measures. As a result, process analysis and redesign work together considering that, as new change options are proposed, they are also analyzed through process analysis techniques. By combining the most promising change options, a redesigned process is achieved. The output of this phase is a to-be process model.
- *Process implementation*. This phase consists in preparing and performing the changes required to move from the as-is process to the to-be process. Process implementation covers two aspects: organizational change management and process automation. Organizational change management refers to the set of activities required to change the way of working of all process participants. Process automation refers to the development and deployment of IT systems (or enhanced versions of existing IT systems) that support the to-be process.
- *Process monitoring and controlling*. Once the redesigned process is implemented and in execution, relevant data are collected and analyzed to determine the process performance regarding its performance measures and objectives. In this phase, eventual bottlenecks, recurrent errors or behavior deviations are identified and corrective actions are undertaken. In the future, new issues may arise, in the same or in other processes, requiring the cycle to be repeated on a continuous basis.

2.1.3 Process Architecture

When an organization starts moving towards the adoption of BPM methodologies, namely modeling its business processes, the complexity of the business process models' landscape rapidly increases.

Hence, several questions arise, such as: how many and which business processes exist in the organization, what are their boundaries, what are the relationships between them, what are their triggers, among others [9, 12].

To address these issues, an organization must define its business process architecture. A business process architecture is a structured overview of the processes that exist in an organization and makes their relationships explicit. These relationships can be of different types [9] [11, p.42]:

- *Composition*. One business process is composed of other business processes - called subprocesses. With this type of relationship one can define different levels of detail in a process architecture (see example in Fig. 2.2).
- *Consumer-producer*. Information or business/data objects flow from one business process to another, where one business process provides its output for another process to take as input.

A business process architecture can be seen as a tool to design a structure for the organizational processes before their detailed design. As a result, it helps organizations understand which processes exist and where one process ends and the next starts. That is why the process architecture is typically designed before the organizational processes [10].

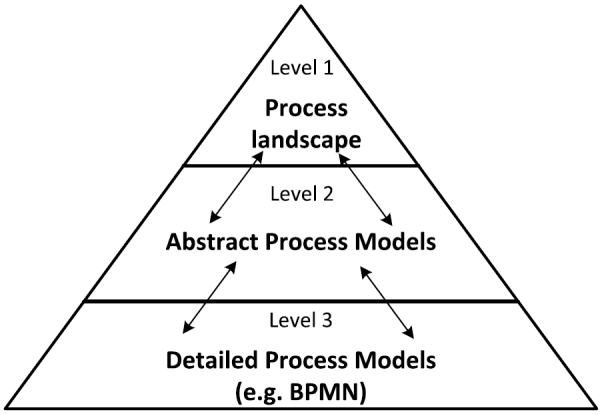


Figure 2.2: Levels of detail in a process architecture (from [11, p.42]).

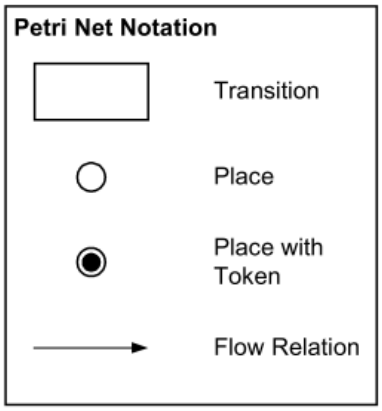
A process architecture can be divided in three levels: level 1 corresponds to the process landscape, which represents the main processes of the organization on a very abstract level and each of its elements points to a more concrete business process on level 2; level 2 represents processes with an higher level of detail but are still quite abstract and each element points to a process model on level 3. At this third level of detail the business processes show control flow, data inputs and outputs as well as assignment of participants [11, p.42].

2.2 Business Process Modeling

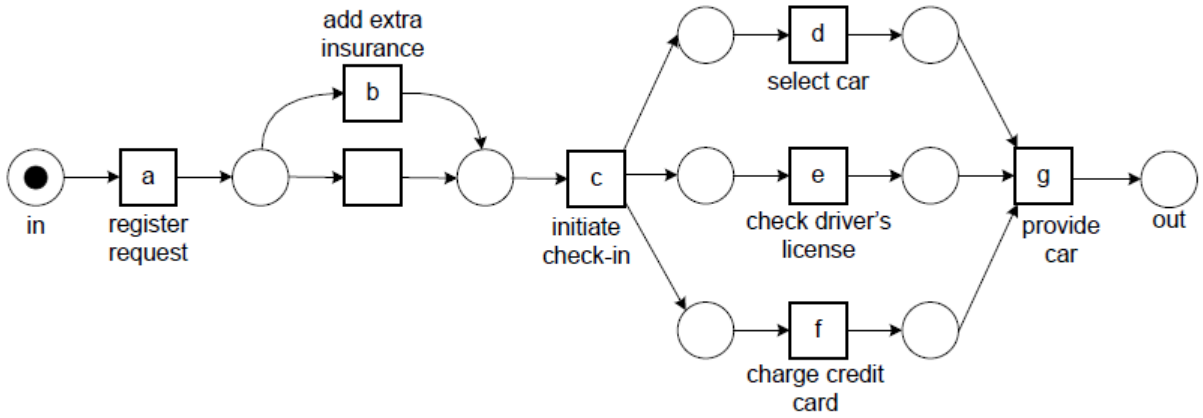
Business process modeling, as defined by Mendling, is "*the human activity of creating a business process model*" ([19]). A business process model is the result of specifying a business process and business process modeling languages offer a predefined set of elements and relationships for guiding the task of business process modeling [19]. The most relevant process modeling languages will be introduced next.

Petri Nets

Introduced by Carl Adam Petri in 1962, Petri nets were one of the first languages used for the specification of business processes and formed a base for the development of other business process modeling languages. Petri nets is a formal language that can be defined as a graph with two different types of nodes. The transitions, which not only can represent activities in a business process but can also be used to capture process routing conditions. The other node type is called places, which define the pre and post-conditions of each transition. Since places and transitions have to alternate accordingly, directed arcs are used to connect them, forming the Petri net graph [19]. According to Mendling, the "*behavioral semantics of Petri nets relate to the concept of a marking (or state)*" ([19]) and each marking assigns none, one or multiple tokens to each place. The behavior of a Petri net (when together with a marking is called system), is defined by the rule: "*a marking of a system enables a transition if and only if all its predecessor places have at least one token assigned*" ([19]). The Petri net notation is presented in Fig. 2.3.



(a) The Petri Net notation (from [19]).



(b) Example of a Petri net (from [37]).

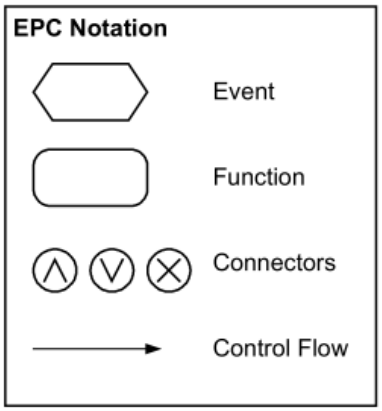
Figure 2.3: Elements of the Petri Net notation (a) and an example of a Petri net (b).

Event-driven Process Chains

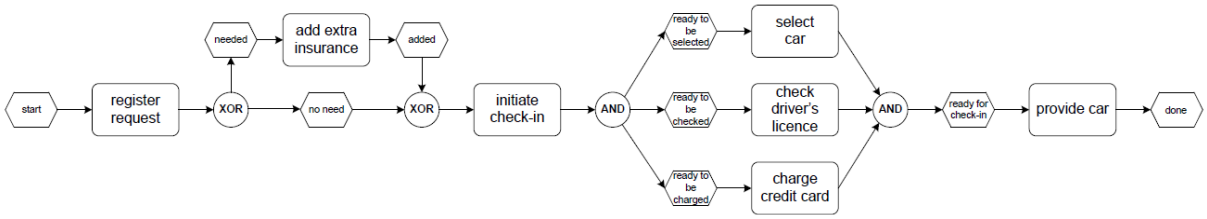
The Event-driven Process Chain (EPC), as defined by Mendling, is a "*business process modeling language for the representation of temporal and logical dependencies of activities in a business process*" ([19]). Identically to Petri nets, the EPC has two types of elements: the functions and the events. The function type elements are used to capture process activities. The event type elements describe the pre and post-conditions of functions. For the definition of the routing rules there are three kinds of connector types: AND, OR and XOR. Connectors can join or split the control flow. Syntactically, functions and events have to alternate, either directly or indirectly when they are linked via one or more connectors. Moreover, OR- and XOR-splits cannot be placed after events, since the latter cannot make decisions. The control flow arcs are used to link all the mentioned elements [19]. The elements that compose the EPC notation are illustrated in Fig. 2.4.

BPMN

Most process modeling languages, including BPMN, take a transformational approach called *input-process-output*. In this approach, processes are divided into activities, which may be divided further into sub-activities. Each activity takes inputs, which are then transformed into outputs. Input and output



(a) Elements of the EPC notation (from [19]).



(b) EPC example (from [37]).

Figure 2.4: Elements of the EPC notation (a) and an example of EPC (b).

relations thus define the sequence flow of work.

One key goal of BPMN was to provide a notation that is understandable by all business users, namely [1]: the business analysts that design the processes, the technical developers responsible for implementing the supporting technology to execute those processes and the business managers who monitor those processes. Moreover, the separation between business process models for business people and the process representations required by the systems designed to implement and execute those processes, lead to the need of translating the original process models to execution models. To address this, other key goal in the development of BPMN was to create a bridge from a visual notation to execution languages [1].

BPMN 2.0 consists of three diagrams: the business process diagram, which will be the focus of this section, conversation diagram, and choreography diagram. Business process diagrams specified in BPMN can include *private business processes* and *public processes*. The *private business processes* are internal to a specific organization and its purpose is to detail the workflow. The *public processes* represent interactions between two or more business entities [1]. The BPMN language is grouped into four basic element categories [1] (see Fig. 2.5):

- *Flow objects* - contain events, activities and gateways. Activities denote the work that is done within a company. Events can be *start*, *intermediate* or *end* and are defined as something that happens in a process and how the process responds to (catching event) or generates a signal (throwing event). Gateways are used for determining branching, forking, merging, or joining of paths within the process.
- *Connecting objects* - used for connecting the flow objects. Sequence flow defines execution order of the activities within a process, while message flow indicates a flow of messages between entities or roles. Associations are used to associate both text and graphical non-flow objects.

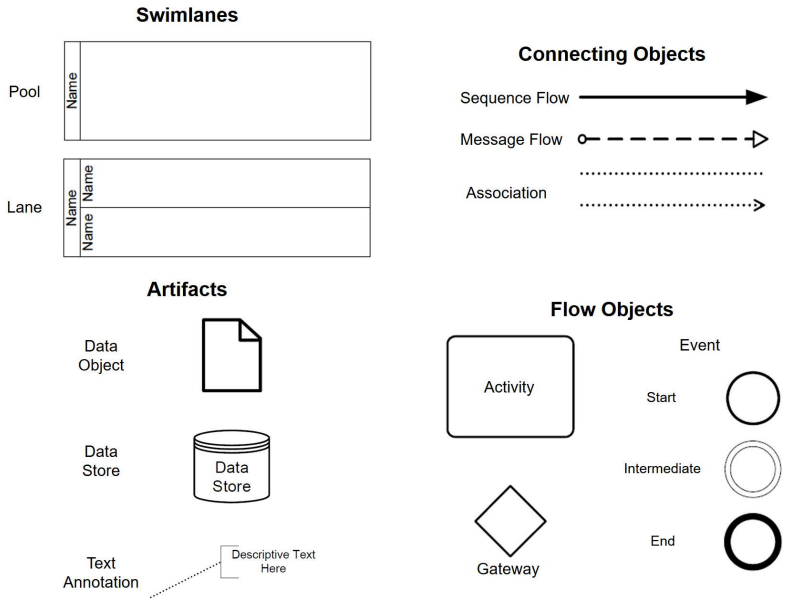


Figure 2.5: Overview of the BPMN core elements (adapted from [15]).

- *Swimlanes* - used to denote a participant or organizational unit in a process and acts as a graphical container for a set of activities taken on by that participant. By dividing *Pools* into *Lanes* (thus creating sub-partitioning), activities can be organized and categorized according to the part of the organizations performing them.
- *Artifacts* - are data objects, data stores, groups and annotations. Data objects represent information resources required or produced by activities. The group element is a visual aid for documentation or analysis purposes while the text annotation is used to add additional information to the model.

2.3 Enterprise Architecture

This section defines the key EA concepts that will be used throughout this document. The definitions of these concepts are based on the International Standard ISO/IEC/IEEE 42010:2011 [16], which specifies the manner in which architecture descriptions of systems are organized and expressed. The terms relevant for the understanding of the remaining of this document are the following:

- **Architecture**: "*fundamental concepts or properties of a system in its environment embodied in its elements, relationships and in the principles of its design and evolution*" ([16, p. 2]).
- **Architecture description**: a "*work product used to express an architecture*" ([16, p. 2]).
- **System**: refers to "*entities whose architectures are of interest*" ([16, p. 3]). The term system-of-interest (or simply, system) "*refers to the system whose architecture is under consideration in the preparation of an architecture description*" ([16, p. 4]).
- **Stakeholder**: an "*individual, team, organization, or classes thereof, having an interest in a system*" ([16, p. 2]).
- **Concern**: an "*interest in a system relevant to one or more of its stakeholders*" ([16, p. 2]).

- **Architecture view**: a "*work product expressing the architecture of a system from the perspective of specific system concerns*" ([16, p. 2]). The term **view** will be used in the remaining of the document as referring to architecture view.
- **Architecture viewpoint**: a "*work product establishing the conventions for the construction, interpretation and use of architecture views to frame specific system concerns*" ([16, p. 2]).
- **Enterprise**: "*any collection of organizations that has a common set of goals*" ([35]).

Given the definitions of the concepts above, EA is defined as "*a coherent whole of principles, methods, and models that are used in the design and realization of an enterprise's organizational structure, business processes, information systems, and infrastructure*" ([18]). The purpose of EA is to capture the essentials of the business, IT and its evolution, while providing a holistic view of the enterprise. Moreover, EA can assist the enterprise in coping with the complexity of ensuring the alignment between business and IT. As a result, EA can be positioned in the context of managing the enterprise, as illustrated in Fig. 2.6. Here, the pyramid represents the layers of enterprise management which contribute to the continuous development of the existing processes, IT systems and other assets, as well as helping to identify necessary changes [18].

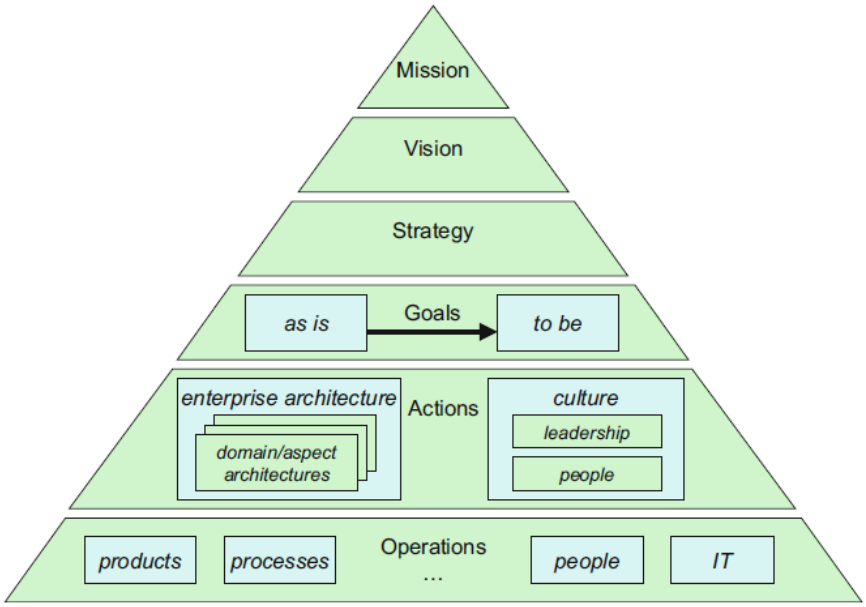


Figure 2.6: The enterprise architecture as a tool for enterprise management (from [18]).

2.4 Enterprise Architecture Frameworks

This section introduces three relevant frameworks which enable the development of EA in organizations, namely: the Zachman Framework, The Open Group Architecture Framework (TOGAF) and Model Driven Architecture (MDA).

2.4.1 The Zachman Framework

John Zachman introduced a framework for enterprise architecture which proposes a matrix-like structure to classify and organize descriptive representations of an enterprise (see Fig. 2.7). Each representation

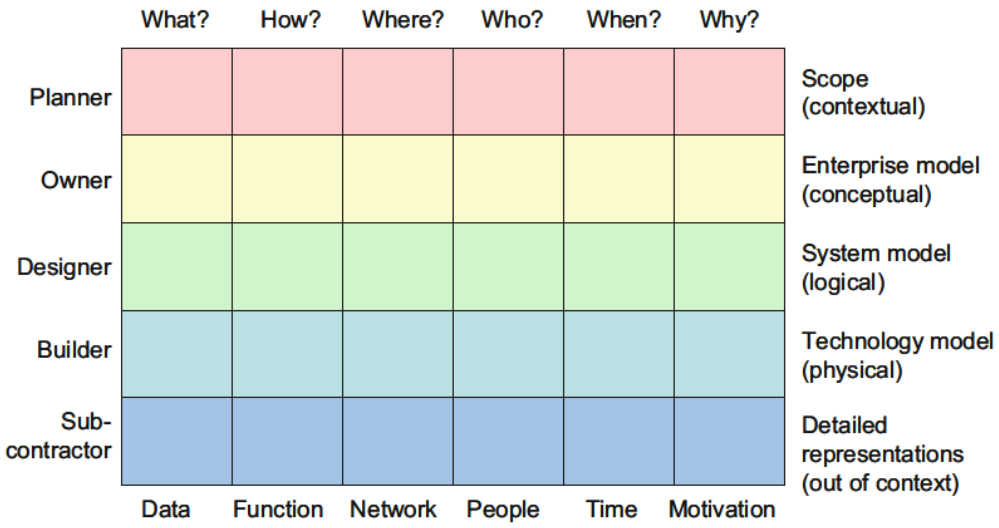


Figure 2.7: The Zachman Framework (from [18]).

corresponds to a cell in the matrix, illustrating the products in the intersection between the stakeholder perspectives and the product abstractions. The stakeholder perspectives regarding the design process are *planner*, *owner*, *designer*, *builder* and *implementer*. The product abstractions depict *who* works on *what* products, *how* they are processed, *where* they are located/related, *when* the process occurs and *why* are certain choices made [34, 39]. Moreover, the product abstractions can be seen as dimensions of the enterprise, each with a distinct focus and purpose. A summary of the dimensions is presented in Table 2.1. The dimensions described here will be revisited in Section 3.2.

Table 2.1: Dimensions of the Zachman Framework (from [33]).

Dimension	Focus	Purpose
What	Data	The enterprise's data and how it is used.
How	Function	The process of translating the mission of the organization into its business and into successive definitions of its operations.
Where	Network	The geographical distribution of the organization's activities and artifacts.
Who	People	Who is related with the major artifacts of the organization: business processes, information and IT. Higher level cells refer to organizational units; lower level refers to system users.
When	Time	How each artifact relates and evolves with time.
Why	Motivation	The translation of goals into actions and objectives.

2.4.2 The Open Group Architectural Framework (TOGAF)

The TOGAF framework is a detailed method and a set of supporting tools for developing an enterprise architecture. This framework is designed to support the four architecture domains that commonly constitute an enterprise architecture [35]:

- The Business Architecture defines the business strategy, governance, organization, and key business processes.

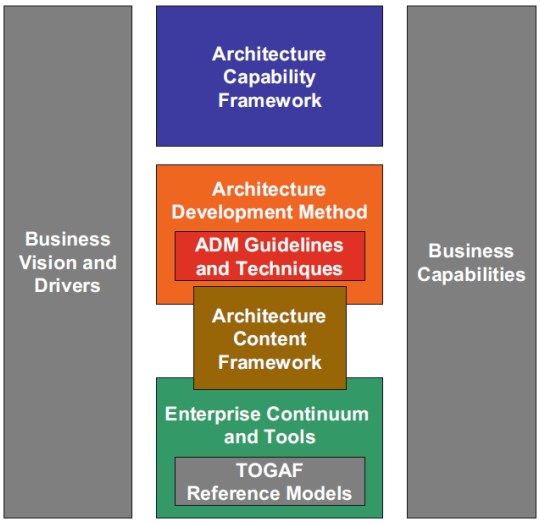


Figure 2.8: TOGAF 9.1 (from [18]).

- The Data Architecture describes the structure of an organization's logical and physical data assets and data management resources.
- The Application Architecture provides a blueprint for the individual applications to be deployed, their interactions, and their relationships to the core business processes of the organization.
- The Technology Architecture describes the logical software and hardware capabilities that are required to support the deployment of business, data, and application services.

TOGAF is a generic framework and intended to be used in various environments. As a result, it may be used alone with the generic deliverables that it describes or these deliverables may be replaced or extended by more specific/appropriate ones defined in any other framework (such as the Zachman Framework). TOGAF 9.1 is composed of the following main elements (see Fig. 2.8) [35]:

- An Architecture Capability Framework, which addresses the organization, processes, skills, roles, and responsibilities required to establish and operate an architecture function within an enterprise.
- Architecture Development Method (ADM) which describes a step-by-step approach to developing an enterprise architecture.
- Architecture Content Framework, which describes the TOGAF content framework, including a structured meta-model for architectural artifacts, the use of re-usable architecture building blocks, and an overview of typical architecture deliverables.
- The Enterprise Continuum, which discusses appropriate taxonomies and tools to categorize and store the outputs of architecture activity within an enterprise.

TOGAF's ADM (Fig. 2.9) describes an iterative method for developing and managing the lifecycle of an enterprise architecture, forming the core of TOGAF. There must be a frequent validation of results against the original expectations, both for the whole ADM cycle and for each particular phase [35]. The Requirements Management is a continuous phase to ensure that any changes to requirements are properly handled with defined governance processes, reflecting these changes in all other phases [35].

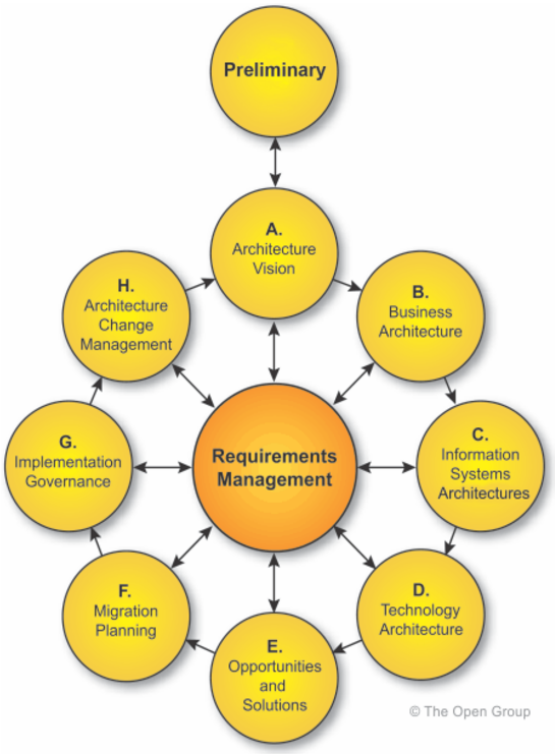


Figure 2.9: TOGAF Architecture Development Cycle (from [35]).

2.4.3 Model-Driven Architecture (MDA)

The MDA, created by the Object Management Group (OMG), supports the full life cycle of physical, organizational and IT systems, representing everything from requirements to business modeling to technology implementations [25]. MDA defines the structure, semantics and notations of models based on OMG's standards: the Unified Modeling Language (UML), the Meta Object Facility (MOF), and the Common Wharehouse Meta-model (CWM) [25]. The purpose of MDA is to represent systems at different levels of abstraction, while maintaining the traceability from enterprise architectures to technology implementations.

MDA comprises three abstraction levels [18, 25]:

1. Business or domain models, used to specify system requirements, historically called Computation-Independent Model (CIM).
2. The Platform-Independent Model (PIM), which describes the interactions between system components while hiding the details necessary for a specific platform.
3. A Platform-Specific Model (PSM) which combines the specifications in the PIM with the implementation details of a particular platform.

To "navigate" between each of the abstraction levels, a set of rules and techniques, called mapping, exists to modify one model to get another model. The mappings between the abstraction levels are illustrated in Fig. 2.10. The next chapter will present the analysis of the related work.

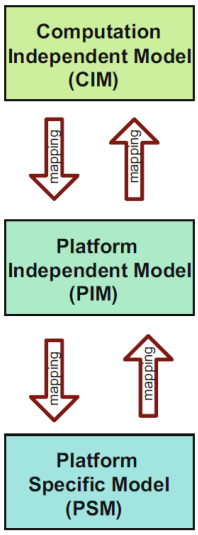


Figure 2.10: Model-Driven Architecture framework (from [18]).

Chapter 3

Related Work

The contents of this chapter support the first three steps of the DSRM process, namely "*Identify Problem & Motivate*", "*Define Objectives of a Solution*" and "*Design & Development*". This chapter presents a literature review of contributions on the topics of view integration and the Zachman Framework dimensions applied to business process modeling. The chapter ends with a brief analysis of the related work regarding its relevance to the proposal of the present work.

3.1 View Integration

This section defines view integration and describes the research works that applied view integration in the fields of database and business process design, respectively.

Conceptual Database Design

View integration first emerged in the field of conceptual database design, where there is extensive work not only on view integration, but also on schema integration. Navathe and Schkolnick in [23] propose a framework for conceptual database design (see Fig. 3.1) based on the Entity-Relationship (ER) Model notation, in which view integration is considered one of the four main phases of conceptual database design:

1. *View Modeling* - Extract and represent each user's view of the real word using requirement specifications as input.
2. *View Integration* - Each of the possibly conflicting user views must be integrated into one data model that represents a global view of the required data.
3. *Model (Schema) Analysis and Mapping* - Analysis of the model produced in the previous step and mapping to the logical structure of an existing database management system.
4. *Physical Schema Design and Optimization* - The logical structure is analyzed regarding the physical design alternatives, resulting in the construction of an "optimal" physical schema.

Navathe and Gadgil continue the previously mentioned work by proposing a methodology for view integration in [22]. Here, the issues regarding the view integration phase of database design are examined

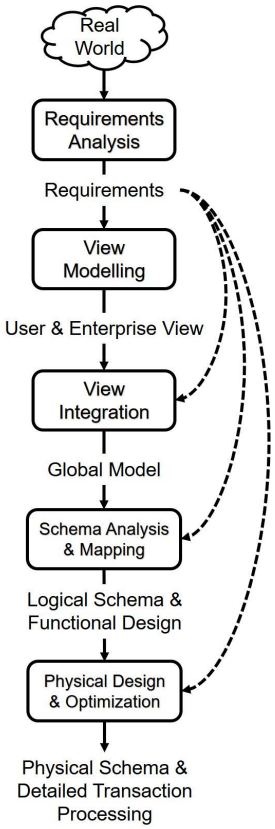


Figure 3.1: Conceptual Framework for Database Design (adapted from [22, 23]).

in detail. Moreover, the authors define a model for the view integration process (see Fig. 3.2), together with the description of an approach for its automation, including operations for merging views [22].

The view integration model shown in Fig. 3.2 represents the inputs and outputs of the view integration process. The inputs are the following: the *Enterprise View* corresponds to a representation of the knowledge of the organization; the *Local Views* correspond to models that represent the user perspectives and may be supplemented by *Intra-view Assertions*; the *Integration Policy* is a set of rules which dictate how the local views must be integrated into an *Enterprise View* (how certain views are preferred over others); the *Inter-view Assertions* specify the equivalent views and the data correspondences between them [22]. The outputs are: the *Statement of Conflicts* presents the conditions which could not be resolved during integration; the *Global View(s)* correspond to the integrated *Local* and *Enterprise* views; the *Modified Assertions* or *Mapping Rules* state the access paths on how data specified in the *Local Views* may be obtained from the *Global View* [22]. As part of the matching process in view integration, a set of rules is presented to define object similarity, which is determined by comparing object names and attributes.

Batini and Lenzerini, following the same research direction, present a similar methodology for view integration using as conceptual model an extension of the ER Model, by adding *subset* and *generalization hierarchies* between entities [3]. One of the distinctive features of this methodology, when compared with the one proposed by [22], is the incremental nature of the view integration process (*n-ary* schema integration) [3].

Batini et al. in [4] present a comparative analysis of methodologies for database schema integration. Here, the purpose is to provide a unifying framework for the problem of schema integration while carrying out a comparative review of the research produced in this area.

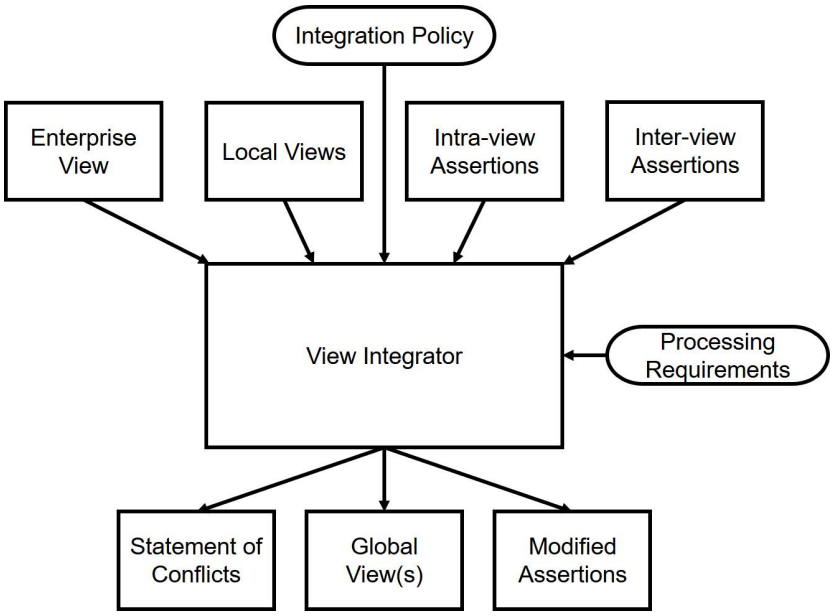


Figure 3.2: View Integration Model (from [22]).

Continuing in the same research direction, Navathe et al. in [24] provide a different approach to view integration. In contrast with the previous approaches that aimed for a fully automated view integration method, the authors argue that view integration can only be accomplished with interactive design tools in a continuous dialogue with the designer. Moreover, they consider that view integration is subjective, therefore, the designer must play a key role in helping to resolve semantic conflicts [24].

Business Process Design

Mendling and Simon in [20] propose a method for business process design by view integration using EPCs. Their method consists of three major steps:

1. Identifying semantic relationships between functions and events of the different models, namely equivalence and sequence.
2. Defining a merge operator for EPCs that also takes two business process views as input to build an integrated model.
3. Defining a set of restructuring rules to remove unnecessary structure from the integrated model.

Morrison et al. in [21] provide a framework for process integration based on process structural and semantic descriptions, using Semantic Process Nets (SPNets) [13]. The SPNet integration methodology comprises the following steps [21]:

1. *Business Process Matching* - May involve the use of three techniques, such as evaluating the labels of business processes and activities to help determine semantic similarity, evaluating the functionality (outcome) of a business process or activity or detecting part-whole relations applying the aforementioned techniques, given that processes may be represented at varying levels of abstraction.
2. *Business Process Integration Goals* - Determine a goal (set of criteria) for the integrated business process, by merging the effect scenarios of each business process to be integrated.

3. *Business Process Integration* - Using a proximity metric associated with each SPNet, search for the most proximally efficient integration through a space of possible integration options.

3.2 The Zachman Framework Dimensions and Business Process Modeling

This section describes the contributions which applied the Zachman Framework dimensions to support business process modeling.

Sousa et al. in [33] provide an analysis of the six Zachman Framework dimensions (introduced in Section 2.4.1) for defining a rule for activity decomposition. This decomposition rule can be defined as the following: "*An activity α can be decomposed into two or more distinct discrete activities if and only if one of the conditions stated is satisfied*" ([33]). These conditions are presented in Table 3.1. Here, the authors consider that a "*business process can be defined as a set of connected activities with inputs and outputs, which interact with people, contribute to achieving business goals, take place in a specific location and occur during a period of time*" ([33]). With this definition of business process, the authors state that since processes are referred as a set of activities, both concepts are interchangeable. As a result, each activity is determined by the values of the six aforementioned dimensions [33].

Table 3.1: Criteria for activity decomposition (from [33]).

Dimension	Criteria
What	α is composed by two or more activities which receive/create different data entities.
How	α is composed by two or more activities which are processed using different applications.
Where	α is composed by two or more activities which occur in different locations.
Who	α is composed by two or more activities which are managed by different business actors.
When	α is composed by two or more activities which are performed in distinct periods of time.
Why	α is composed by two or more activities which exist to serve different purposes.

Pereira and Sousa in [29] use the aforementioned rule to define business process equivalence through activity equivalence. This work is structured as the following:

1. It is presented a correspondence between a set of business process modeling notations and the six Zachman Framework dimensions (see Table 3.2). Each notation element is mapped into one of the referred dimensions.
2. For each dimension, it is defined a hierarchical tree of concepts. An example of hierarchical tree for the *who* dimension is illustrated in Fig. 3.3.
3. Based on the previously introduced definition of business process and on the aforementioned activity decomposition rule, the authors define dimensional activity equivalence as the following: "*An activity (A) is dimensional equivalent to another (A') when they have no different when, what, where, who and why*" ([29]). For example, considering that the activities of a given process are only characterized by the *who* and *what* dimensions, if the activity *Driving* is carried out by *John*

(*who*) using a *Truck* (*what*) and the activity *Riding* which is carried out by *John* (*who*) using also a *Truck* (*what*), the activities labeled as *Driving* and *Riding* are dimensional equivalent [29].

4. Using this definition of activity equivalence and considering that the activities are children of a given process, a process (P) is dimensional equivalent to another (P') if all their children are dimensional equivalent, for the dimensions that are used.

Table 3.2: Mapping between business process modeling notations and the Zachman Framework dimensions (adapted from [29]).

Notation	Element	Dimension
BPMN	Event	When
	Activity	How
	Gateway	Why
	Swimlane	Where (Who)
	Data Object	What
EPC	Event	When
	Function	How
	Organizational Unit	Where (Who)
	Information, material or resource object	What
	Logical connector	Why

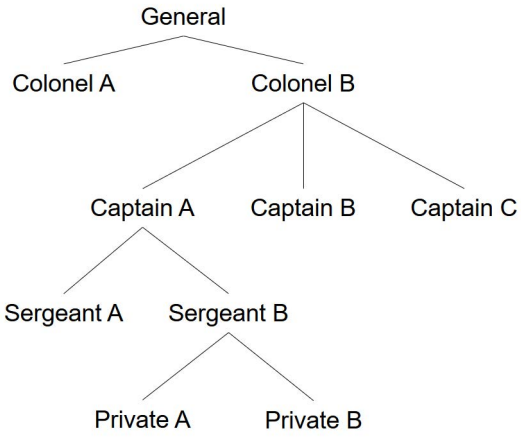


Figure 3.3: Taxonomy example of the *who* dimension (from [29]).

Pereira et al. state that a "*business process can be functionally decomposed into a set of individual tasks*" ([30]). The result of decomposing a business process into a set of individual tasks is formalized into a hierarchical structure defined as "*organizational taxonomy*" ([30]). The organizational taxonomy is composed of a set of dimensions (the six Zachman Framework dimensions) and each dimension contains a hierarchy of business process concept instances. Pereira in [28] proposes the usage of a process repository to support this organizational taxonomy. The business process definition proposed by Pereira et al. is presented in Fig. 3.4. According to several authors, a business process is defined as the following [6, 28, 30, 31]:

How: A business process is defined as a set of connected activities which consumes (inputs) and produces (outputs) tangible or intangible information/business entities, is performed by someone, contributes to achieve business goals, takes place in a specific location and occurs during a given

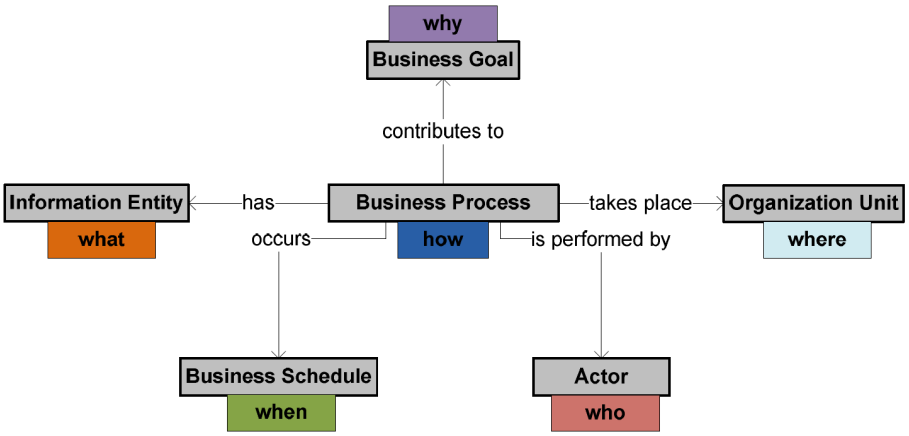


Figure 3.4: Definition of business process used in this thesis (from [30]).

period of time. A business process can be functionally decomposed in as many levels of detail as required.

Who: An actor plays one or more roles. Actors may represent people or systems. An actor may play multiple roles and the same role may be played by different actors.

What: An information entity represents the information about an artifact, person, place, concept, or event that has meaning in the business context.

Where: An organizational unit defines the organizational structure. It may describe a logical unit (e.g. department) or a physical or geographical location.

When: Corresponds to events that are relevant to an organization and its business.

Why: A business goal is an objective that may be decomposed and each one can be associated with the business processes that contributes to its achievement.

A recent work by Awadid ([2]), addresses multi-perspective business process modeling but focused on supporting inter-model consistency, which refers to the extent to which the information contained in each model does not contradict each other. Consistency can be of two types: syntactic and semantic. Here, the proposed solution includes a set of syntactic and semantic consistency rules which help ensuring inter-model consistency.

The following section presents an analysis of the aforementioned contributions, highlighting their relevance to the proposed solution.

3.3 Analysis

Considering view integration in the field of conceptual database design, the definition of the framework for database design proposed in [23] (see Fig. 3.1), namely the *View Modeling* and *View Integration* phases, including the introduction of the global view (or global model) concept, provide a base for the view integration process described in Section 4.2.

The view integration model (see Fig. 3.2) provides an organizational context to view integration. Furthermore, the operations for merging views proposed in [22], including the rules that determine object similarity, constitute a key role in the view integration process because they enable the integration of

objects from distinct views. Object similarity is determined by comparing object names and attributes. A similar approach is followed in the definition of activity equivalence presented in Section 4.2. Besides, the view integration process described later on in this document (Section 4.2) is aligned with the incremental approach used in [3].

Navathe et al. state that "*the best mode of operation is an interactive design tool environment in which the system is strictly an aid*" ([24]). This statement hints that view integration methods should include interactions with the stakeholders, namely by accepting user input during the process. This recommendation is followed in the proposal presented in Chapter 4.

Mendling and Simon in [20] are the first to propose a view integration method applied to business process design and supporting the EPC notation. This work was one of the few in the literature whose approach consists in applying view integration to business process design. This topic was not discussed enough in the research community, therefore this thesis fills a gap in knowledge by defining a novel view integration method applied to business process modeling and supporting BPMN 2.0.

The proposed solution described in Section 4.2 is aligned with the *Business Process Matching* phase of the methodology described by Morrison et al. in [21], namely regarding the detection of part-whole relations aiming to resolve abstraction conflicts.

The Zachman Framework dimensions are applied to business process modeling in [6, 28–31, 33]. These works provide rules for activity decomposition, activity equivalence and business process equivalence as well as a new business process definition. These contributions will play a crucial role in the proposed view integration process described in Section 4.2. Moreover, Pereira and Sousa refer a "*master process repository*" ([29]) which is aligned with the proposed solution, given that it is supported by a business process repository.

Awadid in [2] identifies two ways of managing consistency: consistency checking and inconsistency prevention. The former handles consistency after the models are created while the latter handles consistency even before the creation of models. Even though inter-model consistency is a topic out of the scope of this thesis, it is inevitably present in the view integration process described later on in Section 4.2. The proposed view integration process follows an inconsistency prevention approach, regarding the creation of the consolidated model.

The next chapter introduces the proposed solution to the research problem.

Chapter 4

Proposed Solution

This chapter corresponds to the DSRM step of "*Design & Development*" and it describes the proposal to address the research problem. The research problem and the solution objectives shall be recalled. This thesis aims contribute to address the following problem:

> ***The existence of heterogeneous concerns among the process stakeholders and the lack of a systematic way for representing their concerns in the task of business process modeling, lead to multiple and sometimes conflicting process models for the same business process.***

Based on the analysis of the related work and on the research problem, the following objectives were established for the proposed solution:

1. ***To support the stakeholders in the task of business process modeling by providing a consistent and systematic way for representing their concerns.***
2. ***To provide a method for the integration of distinct business process views specified in BPMN 2.0.***

The remaining of this chapter is structured as the following: first, it is explained the metamodel of the repository that will support the proposed view integration process, which is detailed afterwards, and finishes with the description of the view elements classification.

4.1 Repository Metamodel

This solution proposal uses a business process repository for supporting the view integration process, which will be detailed further in Section 4.2. Such repository incrementally receives a set of business process views as input, together with the stakeholders' classification of the process elements and stores a consolidated representation of the gathered information. Moreover, this proposal is based on the definition of business process described in Section 3.2 (see Fig. 3.4). The classification of the process elements into the different dimensions is based on the BPMN row of Table 3.2. The aforementioned business process definition is extended by considering that organizations, organizational units and departments can be classified simultaneously in the *where* and *who* dimensions (in the context of each specific business process). This slight variation is the result of considering organizations as a system composed of other systems such as organizational units, departments and people. Hence, in this proposal it is considered that the "*Organization Unit*" and the "*Actor*" concepts presented in Fig. 3.4 are

interchangeable for classification purposes. For instance, a pool representing an organization which has two lanes representing actors, it is considered reasonable to classify such organization into the *who* dimension (which is also valid for organizational units and departments).

The metamodel of the repository that will support the process of view integration is presented as an UML class diagram in Fig. 4.1.

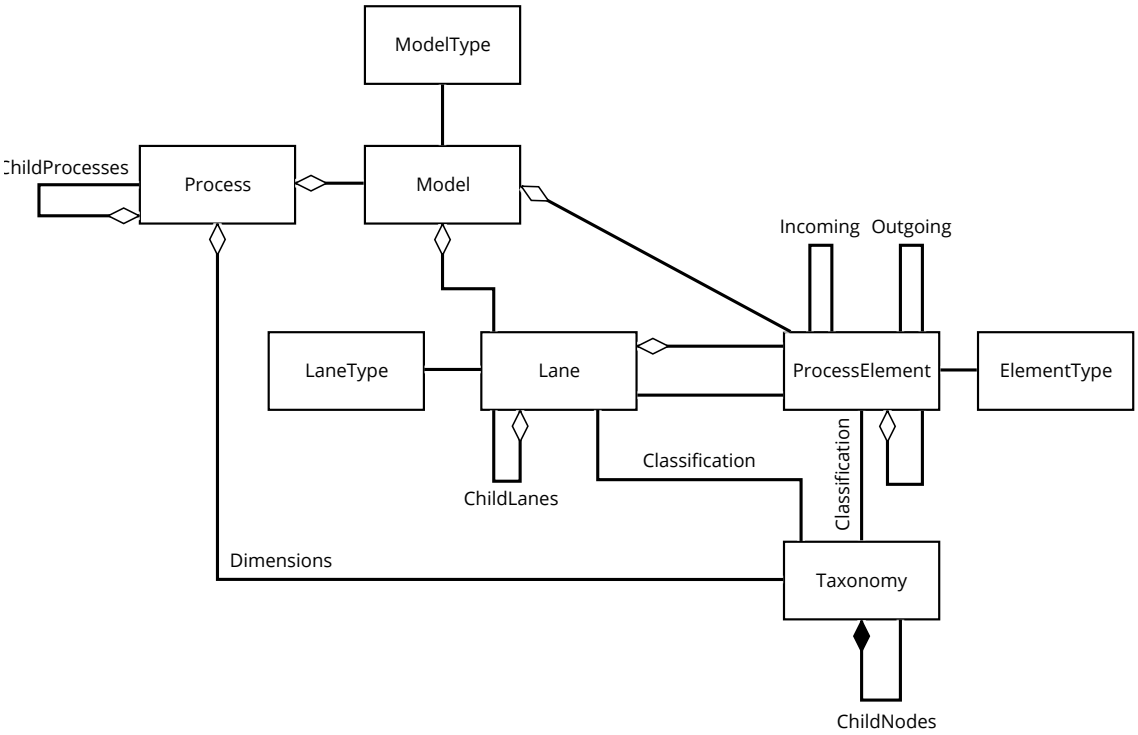


Figure 4.1: Business process repository metamodel.

This metamodel was designed with the objective of being as abstract as possible, since the element types are delegated into other classes. This conceptual design choice increases flexibility and simplicity. Flexibility in extending the metamodel with more types of the existing classes. Simplicity because if an inheritance pattern was chosen, the number of classes in the metamodel would increase, resulting in a more complex model.

The *Process* class represents a business process which is part of the process architecture of the organization. A process can have a parent process and child processes, since organizations can adopt different ranges of abstraction levels in their process architecture. Moreover, a process can be related with instances of the classes *Model* and *Taxonomy*.

The *Model* class has a type given by the *ModelType* class instances. In this proposal three model types are considered: *Input*, *Consolidated* and *Output*. As the model type names suggest, the *Input* type corresponds to the input business process views, the *Consolidated* type corresponds to a consolidated business process model and the *Output* type represents possible output views based on the consolidated model. Each model aggregates instances of the classes *ProcessElement* and *Lane*, where the former represents BPMN data objects and flow objects and the latter represents BPMN swimlanes.

The *Taxonomy* class represents the organizational taxonomy associated with each process. Each process aggregates *Taxonomy* class instances which represent the process dimensions. These process dimensions correspond to the roots of the taxonomy. Below the dimensions in the taxonomy hierarchy,

there are other *Taxonomy* class instances referred as taxonomy nodes. Each taxonomy node represents and classifies instances of the *Lane* and *ProcessElement* classes, which correspond to the elements contained in a business process model view. A taxonomy node has a parent node and can have multiple child nodes. The organizational taxonomy is the structure used for storing the consolidated representation of the input models.

The *Lane* class represents the swimlanes in a BPMN model, namely pool and lane elements. It is considered that the BPMN pool element as a lane without children. Each instance of the *Lane* class can have a parent lane and multiple child lanes. Each lane aggregates instances of the *ProcessElement* class, which correspond to process elements placed on that lane. The *LaneType* class instances represent the types of lanes considered for classification. The types of lanes considered in the repository meta-model are: **System**, **Actor**, **Role**, **Application**, **Organization**, **Organizational Unit** and **Location**. If a lane is classified as an organization or organizational unit, as mentioned before in this section, it is classified as belonging simultaneously to the *who* and *where* dimensions.

The *ProcessElement* class represents BPMN data objects and flow objects, namely activities, gateways and events. Each instance of the *ProcessElement* class can have incoming and outgoing elements. By using these associations, the need for representing connecting objects in this metamodel is abstracted. Each process element has a reference to the container lane and also supports the representation of subprocesses through the self-aggregation association. Moreover, the *ProcessElement* class is associated with the *ElementType* class which defines the type of the *ProcessElement* class instances in a business process model, such as data object, activity, gateway, among others.

4.2 View Integration Process

The view integration process described in this section assumes that the stakeholders and the organization know which processes exist - by defining the process architecture of the organization - and which must be specified in a diagram. Moreover, a preliminary analysis must be performed regarding the dimensions needed to characterize a given business process. Once these dimensions are identified, the taxonomy is composed of the root nodes corresponding to each chosen dimension.

To achieve the integration of business process views in a systematic way, it is proposed the process represented in Fig. 4.2. Here, the lanes *Stakeholder A*, *Stakeholder B* and *Stakeholder C* can represent any organizational unit, modeling team or stakeholder. The *Modeling Tool* and the *Repository* data stores represent, respectively, a modeling tool that supports the designers in modeling and classifying their process views and a repository that supports the process by storing the relevant information captured in the modeling tool. The view integration process starts with the identification of a modeling need by a given stakeholder, who is represented by *Stakeholder A*. This modeling need is associated with the process that must be designed, therefore, it is assumed that the process is already chosen. Then, the stakeholder designs his view of the process using the modeling tool and uploads the model to the repository. Subsequently, the view must be classified and the taxonomy updated in the repository. At this point, the stakeholder performs the classification of the view elements, by providing input information to the modeling tool. Using the gathered information, the modeling tool generates the taxonomy nodes and updates the taxonomy existing in the repository. These steps will be detailed further in the next section.

The aforementioned steps are repeated in the context of another view, represented by the activities in the *Stakeholder B* lane. Once the "*Update Taxonomy*" activity is completed for the second process view, the taxonomy already contains the consolidated information of both views. At this point, another stakeholder

(or one of the previously mentioned) is responsible for exporting the consolidated representation to a process model specified in BPMN, given that the hierarchical representation of the taxonomy is not sufficient for his needs. The generation of a process model from the information stored in the repository is out of the scope of this thesis. However, it is considered that the taxonomy resultant from the view integration process, together with the information stored relative to the different views, is able to provide the necessary information for producing a consolidated model in BPMN 2.0. The generation of business process views from an organizational taxonomy was already addressed in previous works [6, 28].

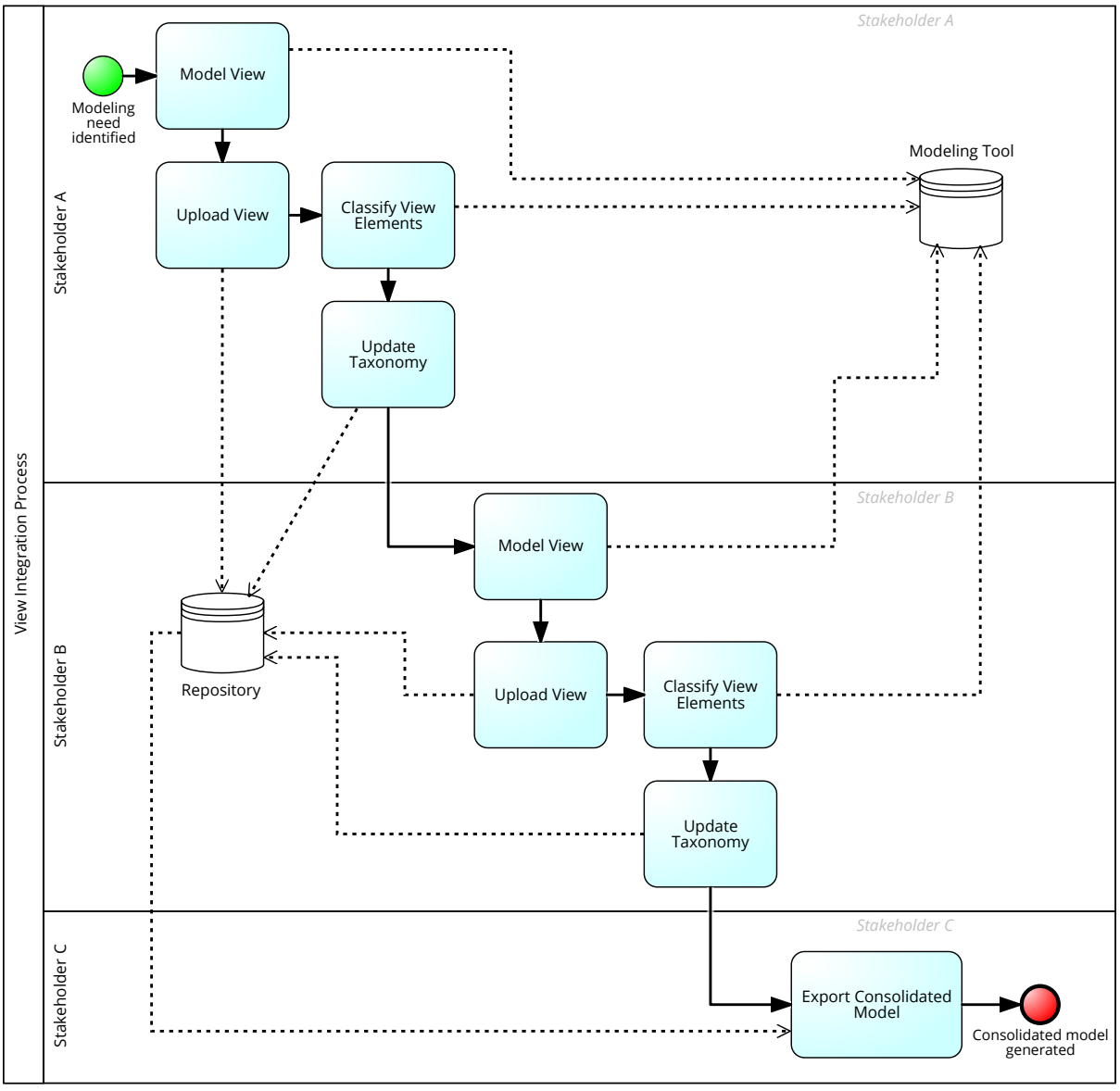


Figure 4.2: BPMN diagram representing the view integration process.

4.3 View Elements Classification

This section will detail further the "*Classify View Elements*" activity of the view integration process. This activity involves the following steps:

1. Classify the events as start, intermediate or end and add the corresponding nodes to the *when*

dimension taxonomy, in case this dimension is used. Classify data objects in the *what* dimension. These steps are executed automatically since there is no need for user input.

2. Classify the pool and the lanes by choosing one of the possible values of the *LaneType* class and add each node to the corresponding dimension taxonomy, according to Table 4.1. The node corresponding to the pool is the parent node of the lane nodes in the taxonomy. In this proposal it is assumed that there is only one pool with multiple lanes.

3. Classify the model activities by assigning values to the dimensions that characterize the process activities. Two sets of dimensions are considered: the process dimensions and the activity dimensions. Regarding the former, the *how* dimension is implicitly associated with the process because it represents the process activities. The set of dimensions that characterizes the activities can be a subset of the process dimensions. For instance, the set of process dimensions can be given by P=$\{$*how*, *where*, *when*, *who*$\}$ and the set of dimensions used to characterize the process activities can be A=$\{$*who*, *where*$\}$. In this case, only the values for the *who* and *where* dimensions must be considered to characterize the activities.

Table 4.1: Mapping between the possible instances of the *LaneType* class and the corresponding dimension.

LaneType	Dimension
Actor Role Organization Organizational Unit	Who
Location Organization Organizational Unit	Where
Application System	What

The classification also implies the creation of the taxonomy nodes to update the taxonomy stored in the repository. The creation of these nodes demands the definition of hierarchical relations between composite activities and tasks. The distinction between composite activity and task is given by the following: a task is an atomic activity which has only one value for each of the dimensions that characterize it; composite activities have more than one value for at least one of the dimensions that characterize it. For example, considering the activity "Cook pizza": $\{$*Who*: "John"; *Where*: "Kitchen", "Oven Room"$\}$, it can be detailed further into two distinct tasks: "Prepare pizza": $\{$*Who*: "John"; *Where*: "Kitchen"$\}$ and "Bake pizza": $\{$*Who*: "John"; *Where*: "Oven Room"$\}$. Therefore, if the stakeholder classifies an activity by adding more than one value to a given dimension, he must add detail to his model and drill down the composite activity into two or more tasks.

Starting from the business process view, to create the taxonomy nodes and the hierarchical relations between them, Algorithm 1 is proposed. Here, lines 1-3 correspond to the preliminary steps that gather information regarding the process that must be designed and the dimensions considered for that process. The algorithm begins with the automatic classification of elements, such as events and data objects. Then, it proceeds to the classification of the pool and the lanes existing in the model. This classification consists in asking the user to choose among the lane type options, which results in adding the respective nodes to the taxonomy in the corresponding dimension. Then, the classification of each activity begins by getting the values for the dimensions from the user input (line 14). At this point, the

current activity node can satisfy one of the following conditions:

1. The current node already exists in the taxonomy, so the user is prompted to confirm the already existing information.
2. The current node refers to a composite activity (line 19). A reference for the composite node is stored, the composite node is added to the taxonomy and the user is alerted to add more detail to the model, by splitting this activity into distinct tasks. Once the user updated the model, the activities classification loop is restarted.
3. The current node refers to a task which is part of the composite activity. This is detected by checking if the current node is dimensional equivalent to the composite activity node (line 24). They are dimensional equivalent if the composite activity node contains a superset of the dimension values found in the current node. As a result, the current node is added in the taxonomy as a child of the composite node (line 25).
4. The current node refers to a task which does not belong to the taxonomy and is not related with any composite node. If this node has any dimensional equivalent nodes in the first hierarchical level of the *how* taxonomy, the current node is added as a child of the equivalent nodes (line 30). If there are no dimensional equivalent nodes in the taxonomy, the current node is added to the taxonomy (line 32).

Regarding Algorithm 1, the conditions based on dimensional equivalence between nodes are only valid if it is considered a reasonable set of activity dimensions. The activity dimensions must be chosen after analyzing the process to be designed. For instance, considering a process that is performed by the same actor in the same location. It is not reasonable to characterize the activities of this process by only taking into account the *who* and *where* dimensions. If we do so, it is not possible to differentiate the process activities using only these two dimensions. If the stakeholders come across a process with such characteristics, they must find differentiating factors in the remaining dimensions.

Algorithm 1: View elements classification algorithm.

Data: An array E containing all the model elements
Result: Process taxonomy

```
1  process ⟵ chooseProcess()
2  dimensions ⟵ getDimensions(process)
3  taxonomy.addNodes(dimensions)
4  begin
5      taxonomy.classifyEvents(E)
6      taxonomy.classifyDataObjects(E)
7      poolNode ⟵ classifyPool(E)
8      taxonomy.addNode(poolNode)
9      laneNodes ⟵ classifyLanes(E)
10     taxonomy.addNodes(laneNodes)
       // beginning of activities classification
11     A ⟵ E.filterActivities()
12     compActivityNode ⟵ null
13     while i < A.length do
14         actNode ⟵ classifyActivity(A[i], dimensions)
15         if actNode ∈ taxonomy then
16             confirm()
17             i ⟵ i + 1
18         else
19             if actNode.isComposite() then
20                 compActivityNode ⟵ actNode
21                 taxonomy.addNode(compActivityNode)
22                 alertUser()
23                 i ⟵ 0
24             else if actNode.dimensionalEquivalent(compActivityNode) then
25                 taxonomy.addChild(compActivityNode, actNode)
26                 i ⟵ i + 1
27             else
                   /* returns an array containing dimensional equivalent activity nodes from the first
                      hierarchical level of the how taxonomy                                          */
28                 eqNodes ⟵ taxonomy.getDimensionalEquivalentNodes(actNode)
29                 if eqNodes ≠ ∅ then
30                     taxonomy.addChild(eqNodes, actNode)
31                 else
32                     taxonomy.addNode(actNode)
33                 i ⟵ i + 1
```

Chapter 5

Demonstration

This chapter corresponds to the "*Demonstration*" and "*Evaluation*" steps of the DSRM, in which the usage of the proposed view integration process is demonstrated in two fictitious scenarios (Section 5.1). Here, the results are shown regarding the implementation of a prototype tool (Section 5.2), which supports each step of the view integration process. Afterwards, are presented the discussion of the demonstration results and then the assessment the proposed solution using DSR evaluation methods already established in the literature.

5.1 Scenarios of Application

5.1.1 Scenario 1

This scenario recalls the Auto Repair Company example introduced in Section 1.2. The proposed view integration process will be applied to the HR and the logistics views of the car body repair process. The steps of the view integration process will be described with a focus on the view elements classification by following Algorithm 1. As mentioned before, the generation of the consolidated model in BPMN 2.0, as part of the "*Export Consolidated Model*" activity, will not be detailed in this thesis.

It is assumed that the repository already contains an instance that represents the car body repair process, including the process dimensions - *how*, *when*, *who* and *where* - and the dimensions to characterize the process activities - *who* and *where*. At this point, the taxonomy existing in the repository only contains the root nodes corresponding to each of the process dimensions. Moreover, it is considered that the car body repair process was chosen and that the logistics team view presented in Fig. 1.2 was already designed and uploaded. Hence, the classification of its elements in the repository taxonomy is performed as follows:

1. **Events: Paint Shop & Mechanical Workshop**. The start and end events are automatically assigned as nodes of the *when* taxonomy.
2. **Pool: Auto Repair Company**. The question "From the following list, how do you classify this pool?" is answered by the stakeholder. The stakeholder must choose one of the aforementioned lane types from the list. The answer should be "Organization" and a new taxonomy node is created in both dimensions *who* and *where*.

3. **Lanes: Paint Shop & Mechanical Workshop**. The question "From the following list, how do you classify these lanes?" is answered regarding both lanes. Again, the stakeholder must choose one of the lane types from the list. The answer is the same for both, but in this case there are two valid answers: "Location" and "Organizational Unit". Here, "Location" should be considered, resulting in the assignment of both lanes to the *where* dimension. If the alternative answer was chosen, the final consolidated taxonomy would be the same. As a result, two child nodes having the Auto Repair Company lane node as a parent are created in the *where* taxonomy.

4. **Activities: Prepare car for painting & Paint car**. Here, the stakeholder is asked to classify each activity. The "*Prepare car for painting*" activity is a composite activity, as a result, two values are given as input for the *who* dimension: "*Mechanic*" and "*Painter*". The composite activity node is added to the taxonomy and the stakeholder is prompted to add more detail to the model by splitting the composite activity into atomic tasks. The "*Prepare car for painting*" activity is replaced with the "*Straighten car*" and "*Spackle car*" tasks in the model. These tasks are classified with the values $\{$*Where*: "Mechanical Workshop", *Who*: "Mechanic"$\}$ and $\{$*Where*: "Mechanical Workshop", *Who*: "Painter"$\}$, respectively. Both tasks are added as child nodes of the "*Prepare car for painting*" activity in the *how* taxonomy. The "*Paint car*" activity is classified with the values $\{$*Where*: "Paint Shop", *Who*: "Painter"$\}$.

The state of the repository after uploading and classifying the elements of the logistics team view is presented in Fig. 5.1. At this point, the repository already has information regarding the logistics team

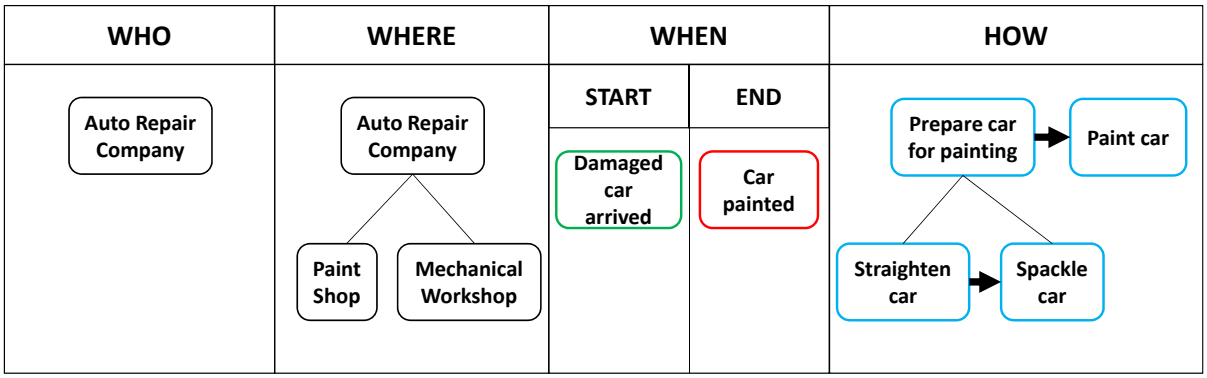


Figure 5.1: State of the repository after uploading and classifying the logistics team view.

view. The activity responsible for the classification of the elements belonging to the HR team view will be described now. The steps for classifying these elements in the taxonomy are the following:

1. **Events: Paint Shop & Mechanical Workshop**. The start and end event nodes already exist in the *when* taxonomy.

2. **Pool: Auto Repair Company**. Now that the taxonomy contains the element nodes from the logistics view, it is suggested to the stakeholder that the pool "*Auto Repair Company*" already exists in the taxonomy. Thus, considering that the stakeholder agrees with the classification of this lane as "Organization", the stakeholder confirms the already existing nodes for this pool in the *who* and *where* dimensions.

3. **Lanes: Painter & Mechanic**. This step is analogous to the one described for the classification of the logistics team view. The difference lies in the classification given to the lanes, which in this case is "Role". The result of this step is adding both lanes as child nodes of the "*Auto Repair Company*" node in the *who* dimension.

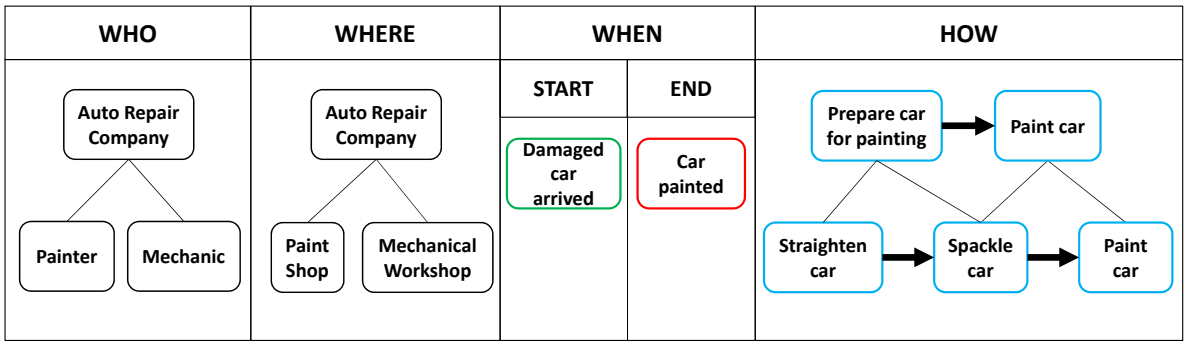


Figure 5.2: State of the repository after uploading and classifying the HR team view.

4. **Activities: Prepare car for painting & Paint car**. In this step, as previously described, the stakeholder proceeds to the activities classification. Regarding the "*Prepare car for painting*" activity, this node already exists in the taxonomy, therefore it is confirmed by the stakeholder. The "*Paint car*" activity is a composite activity, as a result, two values are given as input for the *where* dimension: "Mechanical Workshop" and "Paint Shop". At this point, when adding the composite activity node to the taxonomy an insertion conflict arises, since a dimensional equivalent node with the same name already exists. Both nodes refer to an activity with the same name and they are dimensional equivalent, however the nodes are not the same because the existing one is an atomic task and the current is a composite activity. To resolve the conflict, the atomic task node is replaced with the composite node to guarantee there is no loss of information. Once the replacement occurred, the stakeholder is prompted to add more detail to the model by splitting the composite activity into atomic tasks. The "*Paint car*" activity is replaced with the "*Spackle car*" and "*Paint car*" tasks in the model. These tasks are classified with the values $\{$*Where*: "Mechanical Workshop", *Who*: "Painter"$\}$ and $\{$*Where*: "Paint Shop", *Who*: "Painter"$\}$, respectively. The "*Spackle car*" task node is selected in the taxonomy while the "*Paint car*" task node is created and both are added as children of the "*Paint car*" composite activity in the *how* taxonomy.

After uploading and classifying the elements of the HR team view, the final state of the repository is reached (see Fig. 5.2). At this point, with the information present in the consolidated taxonomy, it is possible to generate the consolidated model presented in Fig. 1.4.

5.1.2 Scenario 2

In this section it is presented the example of an airport, where two distinct stakeholders design the process of flight departures from the passenger point of view. This example is adapted from [11, p.69] and it is described as the following:

> When a passenger intends to take a departure flight, he must first go to the airport check-in counter to perform the check-in. Once the check-in finishes, the passenger receives the boarding pass and proceeds to the location of the security checkpoint. The passenger passes through this checkpoint by getting a security screening of himself and his luggage. After passing security, the passenger proceeds to the boarding gate and waits for the gate opening to board the plane.

Considering the airport management company, the Departures Manager wants to understand the typical itinerary of a passenger between arriving at the check-in counter and reaching the boarding gate, to measure the time. Thus, he obtains the process model shown in Fig. 5.3.

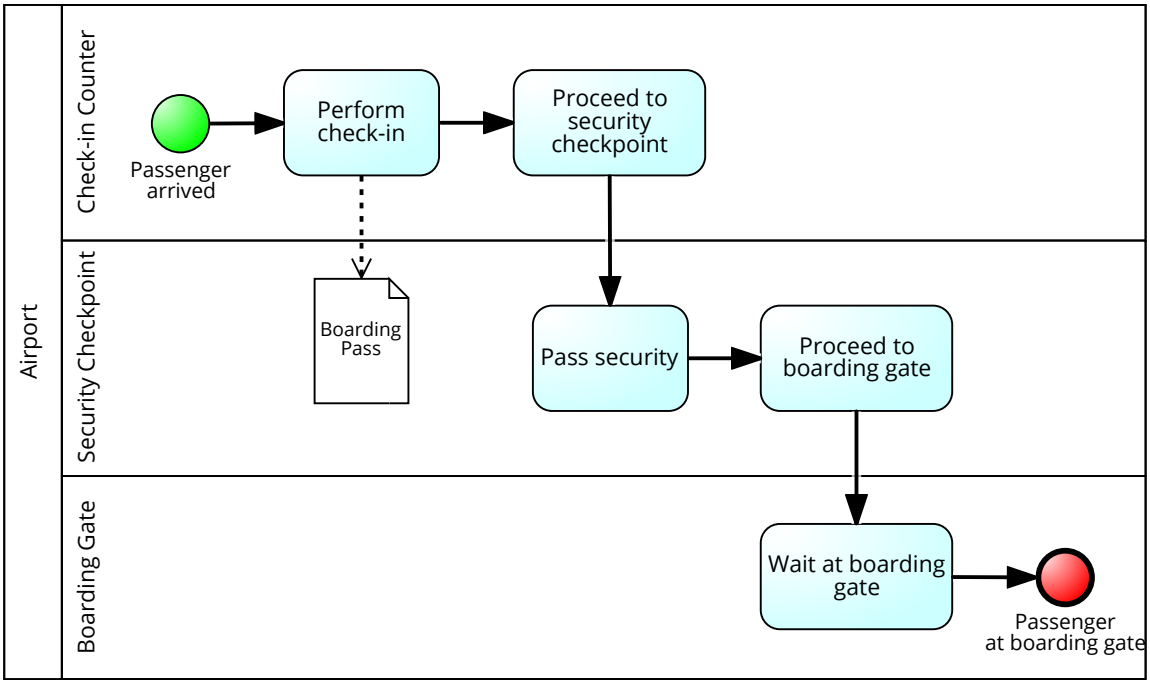


Figure 5.3: The Departures Manager view of the passengers departure process.

In the same company, the Passenger Support Manager - responsible for ensuring the passengers satisfaction while they are in the airport - wants to measure the average time spent in each location between a passenger arriving at the check-in counter and entering the boarding gate. The process model obtained by the Passenger Support Manager is presented in Fig. 5.4.

Now the proposed view integration process will be applied to this example. It is assumed that the repository already contains an instance that represents the passengers departure process, including the process dimensions - *how*, *where*, *what*, *why* - and the dimensions to characterize the process activities - *where*, *what*, *why*. The *who* dimension was omitted because the values for this dimension are constant throughout the process and the *when* dimension was also omitted because the classification of the events was already addressed in the previous example. Thus, the steps for classifying the Departures Manager view elements, according to Algorithm 1, are the following:

1. **Data Object: Boarding Pass**. The data object is automatically assigned as a node of the *what* taxonomy.

2. **Pool: Airport**. The question “How do you classify this pool?” is answered by the stakeholder. The stakeholder must choose one of the aforementioned lane types from a list. The answer should be “Location” and a new taxonomy node is created in the *where* dimension.

3. **Lanes: Check-in Counter, Security Checkpoint & Mechanical Workshop**. The question “How do you classify these lanes?” is answered regarding the present lanes. Again, the stakeholder must choose one of the lane types from a list. The answer should be “Location”, resulting in the assignment of the lanes as child nodes of the Airport lane node in the *where* dimension.

4. **Activity: Perform check-in**. At this point, the stakeholder is asked to classify each activity. The “*Perform check-in*” activity is an atomic task, therefore, it is classified with the values $\{$*Where*: “Check-in Counter”, *What*: “Boarding Pass”, *Why*: “Receive boarding pass” $\}$. The corresponding

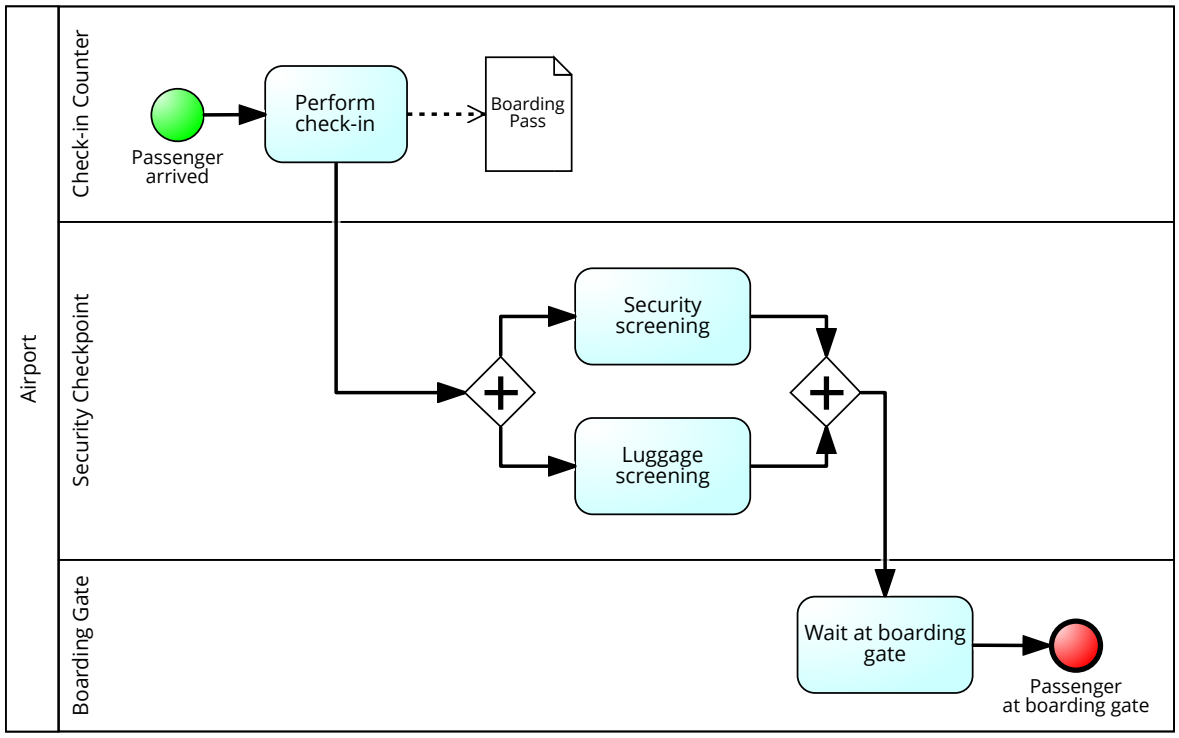


Figure 5.4: The Passenger Support Manager view of the passengers departure process.

activity node is added to the taxonomy in the *how* dimension and the activity *why* value is added as a node of the *why* dimension.

5. **Activity: Proceed to security checkpoint**. The "*Proceed to security checkpoint*" activity is an atomic task but it can be only classified with the values $\{$*Where*: "Check-in Counter" and *Why*: "Getting to security checkpoint"$\}$. The corresponding activity node is added to the *how* dimension and the *why* value is added to the coincident dimension.

6. **Activity: Pass security**. The "*Pass security*" activity is an atomic task and it is classified with the values $\{$*Where*: "Security Checkpoint" and *Why*: "Flight security regulations"$\}$. This task node is added to the *how* dimension.

7. **Activity: Proceed to boarding gate**. The "*Proceed to boarding gate*" activity is an atomic task and must be classified with the values $\{$*Where*: "Security Checkpoint" and *Why*: "Boarding the flight"$\}$. The node is added to the taxonomy in the *how* dimension and the value "Boarding the flight" is added as a node to the *why* dimension.

8. **Activity: Wait at boarding gate**. The "*Wait at boarding gate*" activity is an atomic task and it is classified with the values $\{$*Where*: "Boarding gate" and *Why*: "Boarding the flight"$\}$. This task node is added to the *how* dimension and the rest of the dimensions remain unaltered.

The taxonomy resulting from classifying the elements of the Departures Manager view is illustrated in Fig. 5.5. Adding the *why* activity dimension values to the taxonomy is not explicit in Algorithm 1 because it is part of the `classifyActivity` function.

The classification of the Passenger Support Manager view elements consists of the following steps:

1. **Data Object: Boarding Pass**. The data object node already exists in the *what* taxonomy.

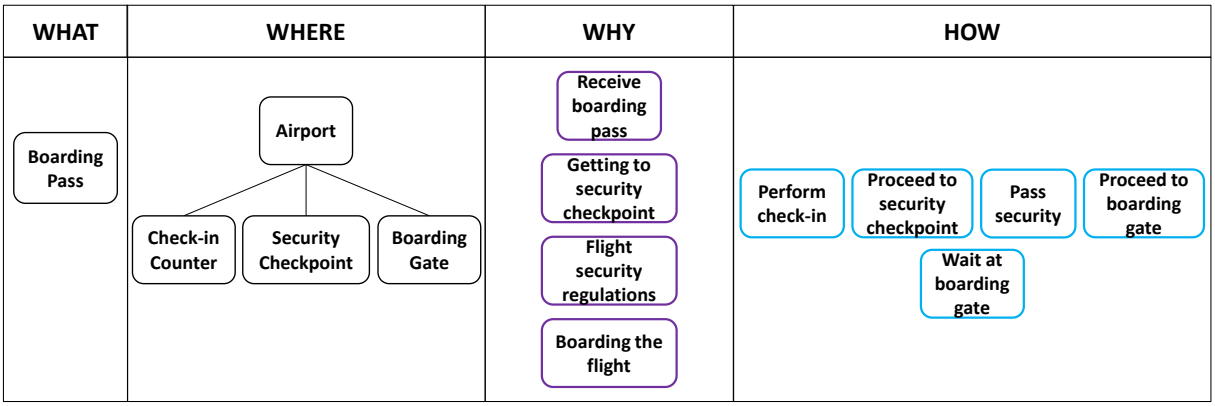


Figure 5.5: Taxonomy resultant from applying the view integration process to the Departures Manager view.

2. **Pool: Airport**. The node corresponding to this pool already exists in the taxonomy so its classification is suggested to the stakeholder. Agreeing with the suggested classification, the stakeholder confirms the already existing node.

3. **Lanes: Check-in Counter, Security Checkpoint & Mechanical Workshop**. Similarly, the nodes corresponding to these lanes already exist in the taxonomy, therefore, the stakeholder just confirms their classification.

4. **Activity: Perform check-in**. At this point, the activities classification step begins. The "*Perform check-in*" activity already exists in the taxonomy, therefore it remains unaltered upon confirmation.

5. **Activities: Security screening & Luggage screening**. These activities are atomic tasks and are classified with the values $\{$*Where*: "Security Checkpoint" and *Why*: "Flight security regulations"$\}$. These nodes are not in the taxonomy and they are dimensional equivalent to the "*Pass security*" node. As a result, they are added as children of the "*Pass security*" activity node in the *how* dimension. The *why* taxonomy does not suffer any change because the "Flight security regulations" value already exists.

6. **Activity: Wait at boarding gate**. The "*Wait at boarding gate*" activity already exists in the taxonomy, therefore it remains unaltered upon confirmation.

The final state of the taxonomy which represents the integration of both views is shown in Fig. 5.6. Given the information present in the consolidated taxonomy, it is possible to generate the consolidated model presented in Fig. 5.6.

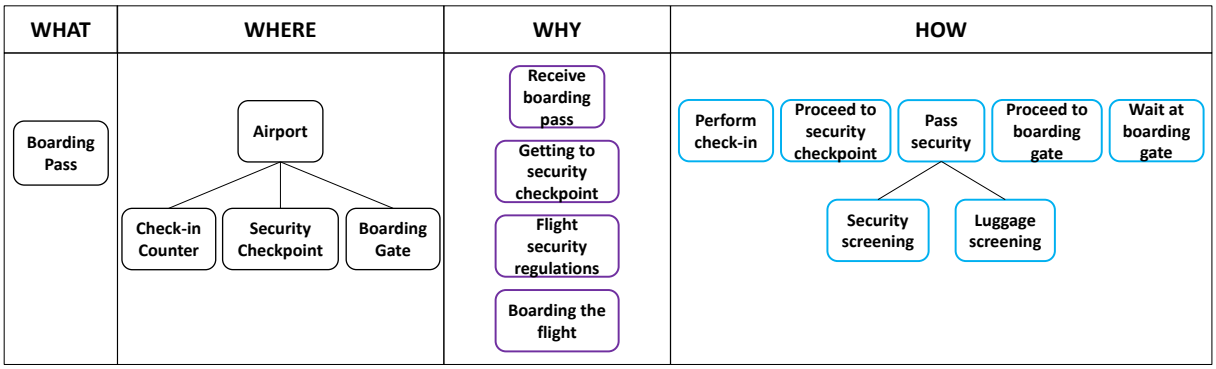


Figure 5.6: Consolidated taxonomy after classifying the Passenger Support Manager view.

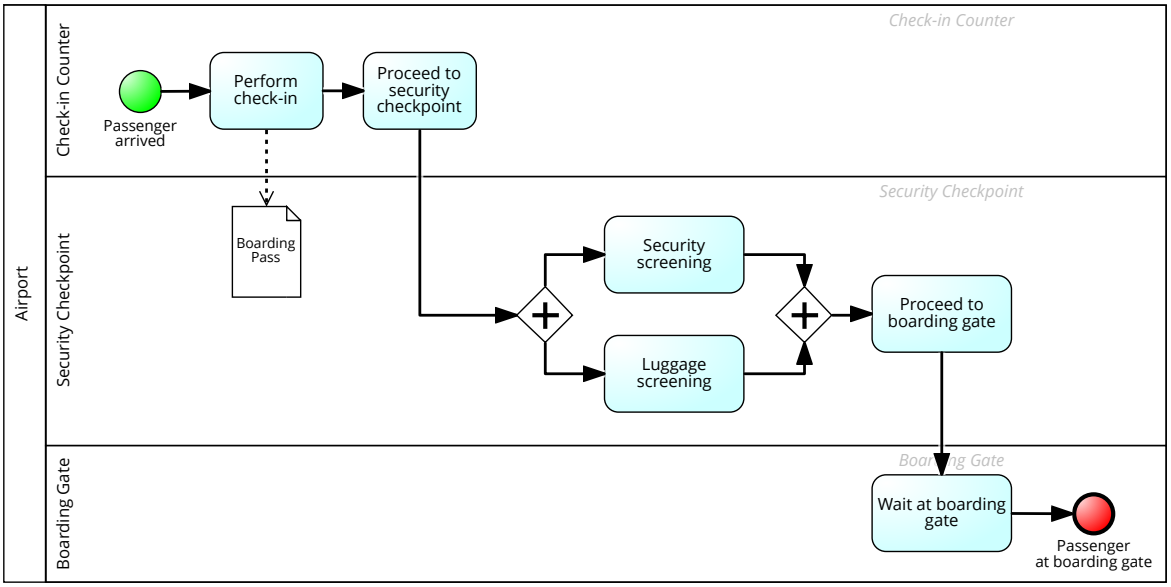


Figure 5.7: The consolidated model, specified in BPMN 2.0, resultant from the integration of the departure process views.

5.2 Tool Support Example

This section presents an implementation example of the proposed algorithm, using the EAMS[1] repository and the BPMN.io[2] JavaScript library. The communication between the EAMS repository and the BPMN modeling tool was performed through Representational State Transfer (REST) web services. The modeling tool prototype was developed by extending the modeler example of the BPMN.io library.

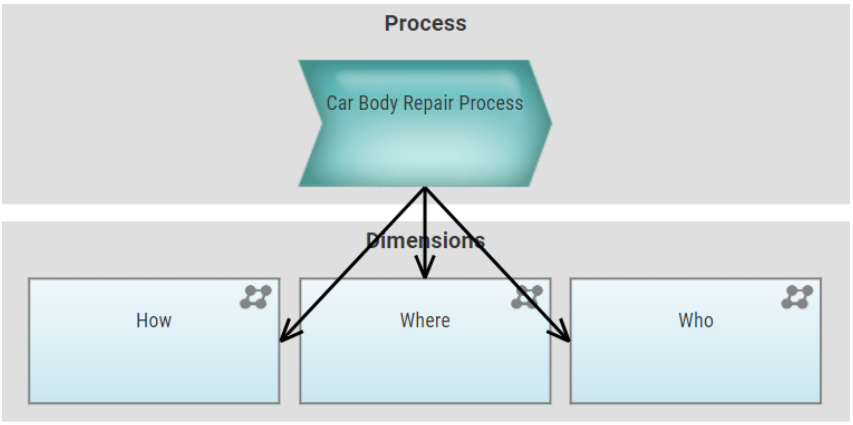
Figure 5.8 shows the initial state of the repository with the manually loaded information, namely the process and the relevant dimensions. Moreover, the modeling prototype interface shows a business process view already specified. This figure corresponds represents the starting conditions of the algorithm: the process to be modeled, the relevant dimensions and the process view specified.

Figure 5.9(a) shows how the user can interact with the prototype to choose the process while Fig. 5.9(b) represents the first step of the algorithm, which is classifying the pool of the process view. The latter illustrates how the user can choose the pool type from a drop-down list.

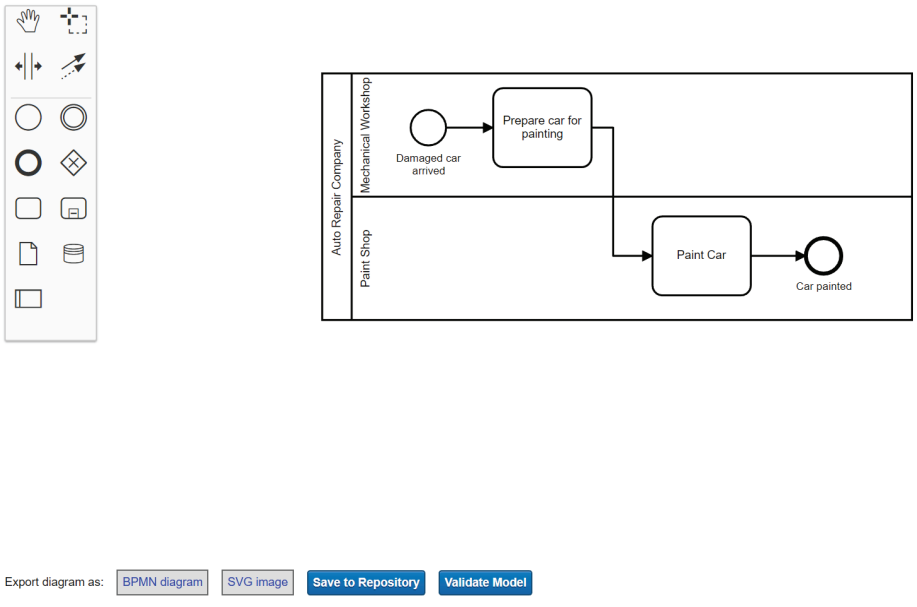
Figure 5.10(a) shows the classify lanes step, including the drop-down list with the possible lane types. Figure 5.10(b) illustrates another step of the algorithm, namely the classification of the first activity, which in this example is also a composite activity. Furthermore, this figure shows the alert displayed for the user to add detail to the model.

Figure 5.11 shows the classification of the tasks resultant from the decomposition of the composite activity. Figure 5.12(a) illustrates the classification of the last model task. At this point, the activities classification step finishes and the taxonomy resulting from the classification of the first process view is uploaded into the repository. Analogously, the second view is designed and classified, as described before in Section 5.1.1. Figure 5.12(b) illustrates the taxonomy representation resultant of the view integration process applied to the example described in Section 5.1.1.

[1] http://www.linkconsulting.com/eams/
[2] https://bpmn.io/



(a)



(b)

Figure 5.8: Initial state of the taxonomy (a) and the modeling interface (b).

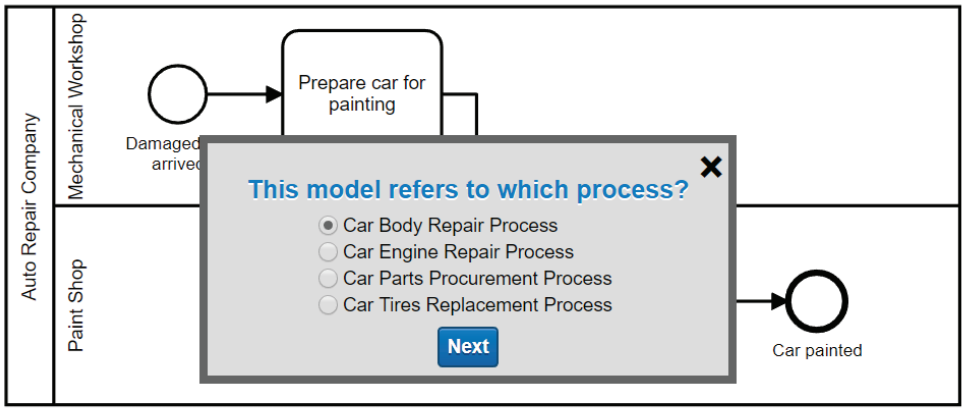
The next chapter will present the analysis of the demonstration results and evaluate the proposed solution artifact.

5.3 Discussion

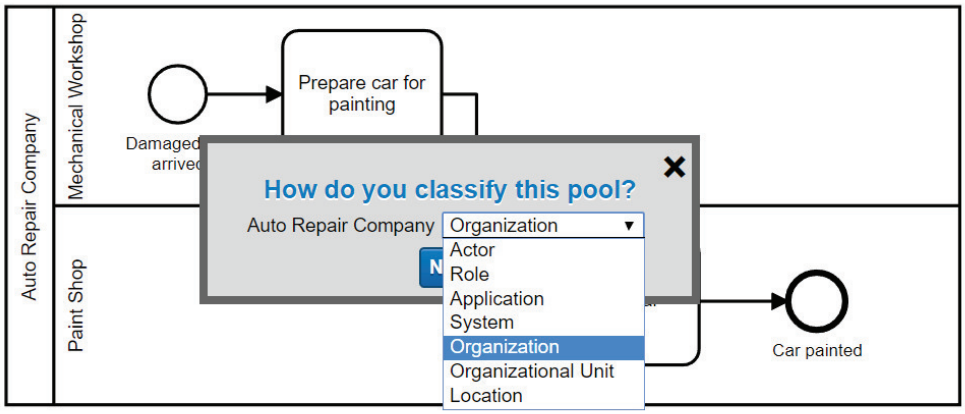
In this section are discussed the demonstration results presented in Chapter 5, starting with the results from the scenarios of application and ending with the results of the modeling tool prototype.

At this point, the established solution objectives shall be recalled:

1. ***To support the stakeholders in the task of business process modeling by providing a consistent and systematic way for representing their concerns.***
2. ***To provide a method for the integration of distinct business process views specified in BPMN 2.0.***



(a)



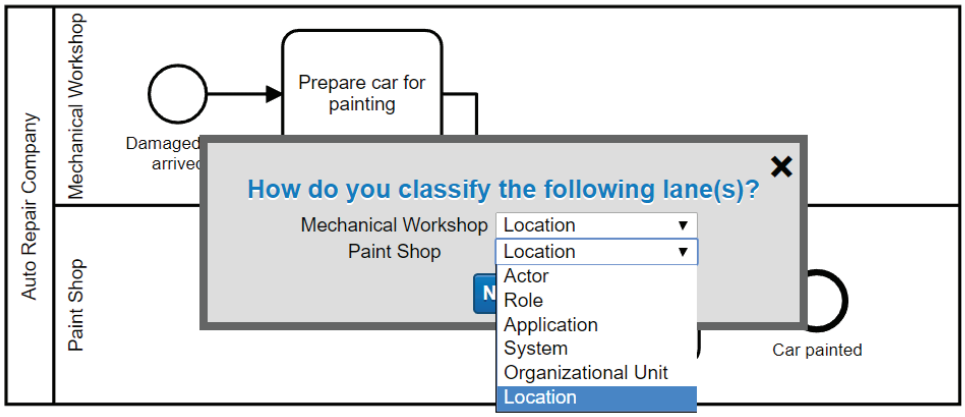
(b)

Figure 5.9: Choosing the process (a) and pool classification (b).

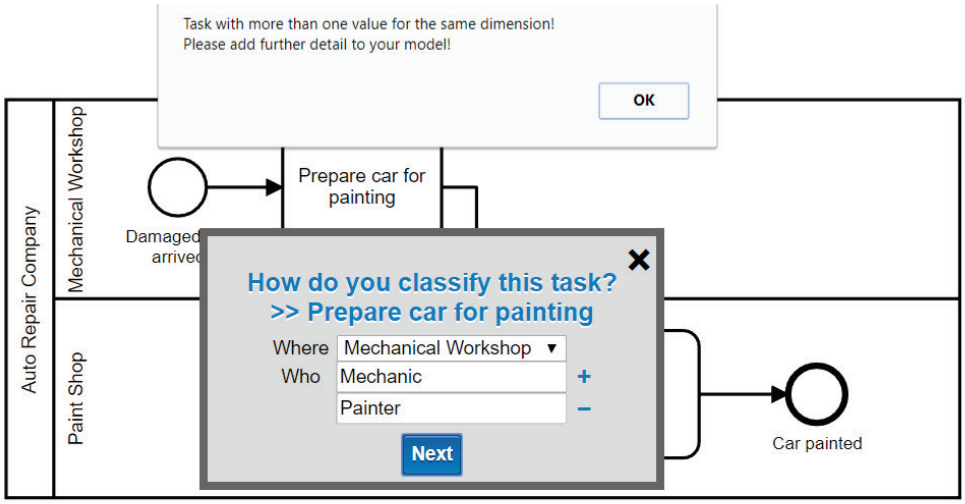
Regarding the aforementioned objectives, it must be pointed out that this thesis does not aim to propose or assess the “best” way for concerns representation, neither it states that the chosen dimensions are the adequate mean to do so.

To attain the established objectives, it is proposed a process for achieving the view integration of business process models. Using the process dimensions and the *LaneType* class instances in the classification step of the view integration process, it is possible for the stakeholders to express their concerns in a systematic way. On the one hand, the stakeholders provide their input values to the activity dimensions and on the other hand, they classify the model pools and lanes by choosing the adequate type among the list of *LaneType* class instances. The view integration process, together with the repository metamodel (Fig. 4.1) and the view elements classification algorithm (Algorithm 1), compose a method able to achieve the integration of business process views specified in BPMN 2.0. Examples of consolidated representations (Figs. 5.2, 5.6 and 5.7) resultant from the view integration process demonstrate that the established objectives were achieved with success.

Section 5.1.1 demonstrates the application of the proposed view integration process to the Auto Repair Company scenario. The results show how the classification of the different views can be used to address the distinct perspectives regarding the abstraction level of activities. The taxonomy resultant from the



(a)

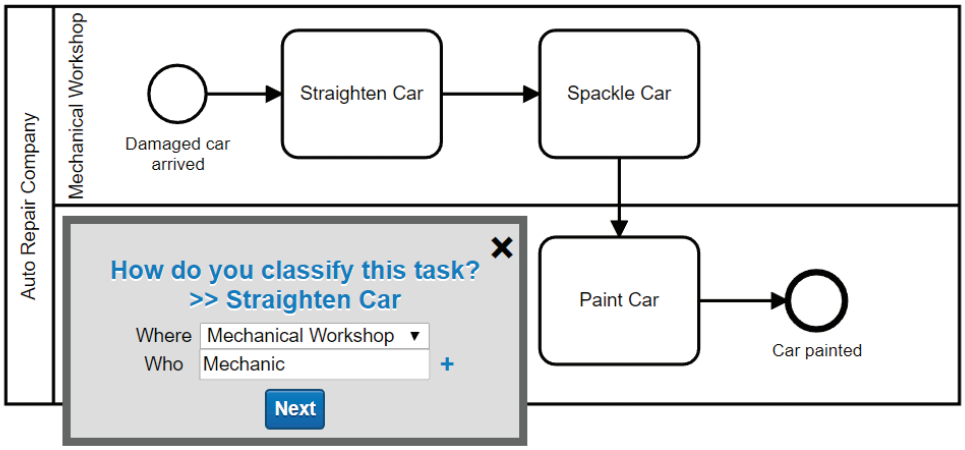


(b)

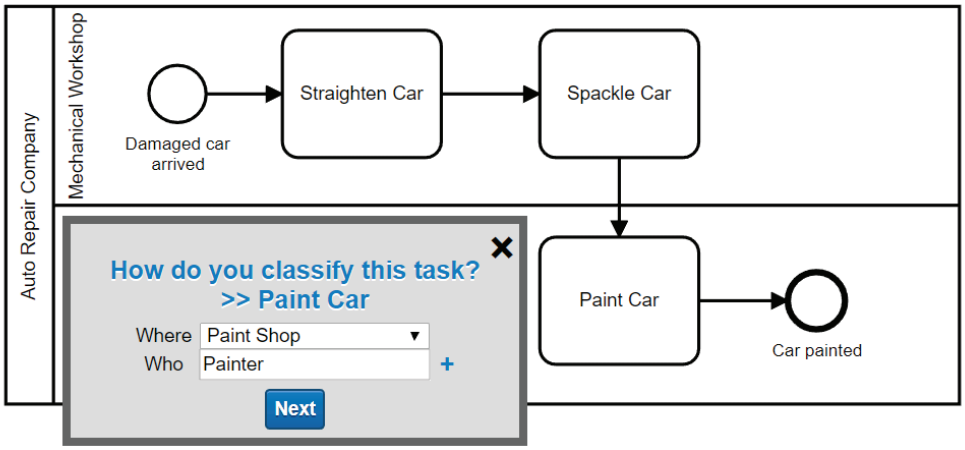
Figure 5.10: Lanes (a) and composite activity (b) classification.

classification of both views (Fig. 5.2) shows the activities in the first level of detail - with the composite activities "*Prepare car for painting*" and "*Paint car*" - and the atomic tasks in the second level. Hence, it was possible to integrate activities from different views and with distinct levels of abstraction. This was achieved by "forcing" the stakeholders to detail their activities until they end up with atomic tasks. Using the activity dimensions values, the activities of different views were drilled down to the same level of detail without losing information.

Section 5.1.2 shows how the proposed view integration process can be applied to the Airport scenario. This scenario is more complex, not only because of the parallel gateway present in Fig. 5.4, but also due to the inclusion of the *what* and *why* dimensions. The results show how the *why* dimension can be used to achieve the integration of activities, which otherwise could be perceived as unrelated. However, in this scenario there is no information to complete the sequence flow between the activities. The information regarding the split and join of the sequence flow through the usage of parallel gateways is not present in the consolidated taxonomy, even though this information could be accessed through other class instances of the repository metamodel (*ProcessElement* and *Model*). Hence, for the generation of



(a)



(b)

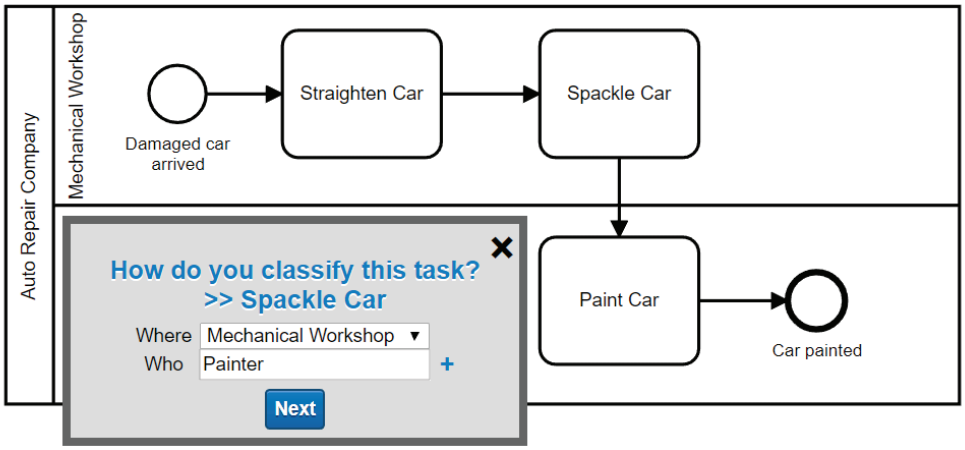
Figure 5.11: Classification of the tasks that decompose the composite activity.

a correct BPMN model, additional information must be captured the modeling tool, for instance to handle parallel control flow splits.

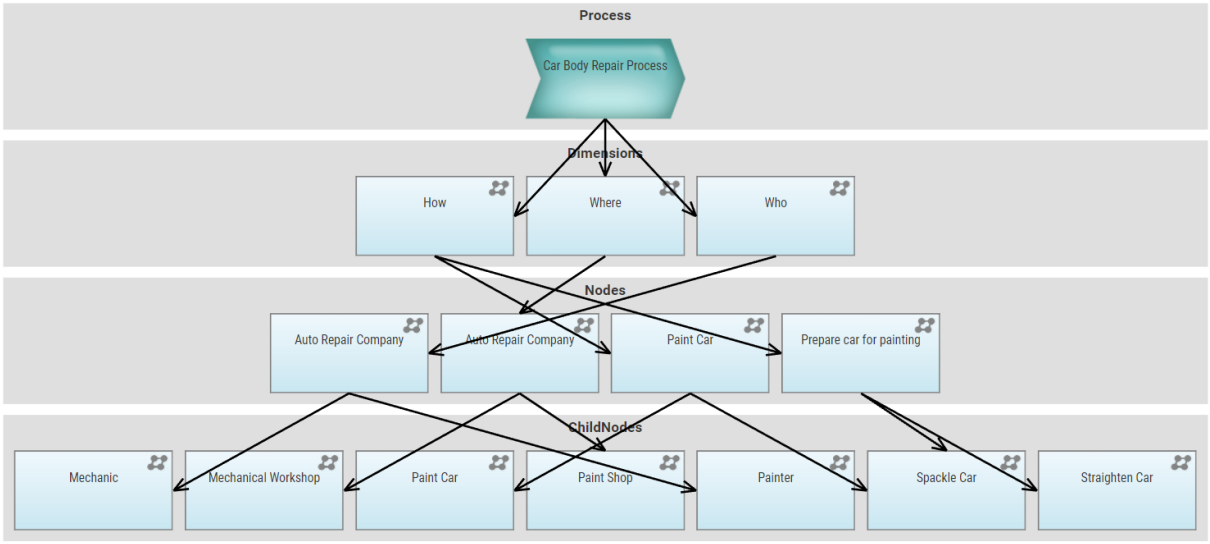
Section 5.2 shows how the proposed view integration process could be implemented in a modeling tool. Here, the modeling tool prototype, together with the EAMS repository, support the view integration process end-to-end. The EAMS tool provides each class instance of the repository metamodel with its own life cycle, enabling a timeline of changes and a gap analysis. Here, the “time factor” is embedded into the business processes and the corresponding models, which supports organizations managing their multiple changes over time. Moreover, it is demonstrated how the solution artifact could be instantiated for practical use.

5.4 DSR Evaluation

To evaluate the design artifact resultant from the DSRM process, the artifact evaluation approach proposed by Prat et al. will be used together with the four research principles introduced by Österle et al.([26, 32]).



(a)



(b)

Figure 5.12: Classification of the last task (a) and blueprint with the final state of the taxonomy (b).

Prat et al. proposed a hierarchy of evaluation criteria for IS artifacts and a model of generic evaluation methods. An evaluation criterion can be decomposed into levels, forming a hierarchy, and each criterion is assessed by a generic evaluation method [32]. In the proposed hierarchy, the evaluation criteria are organized along the five dimensions of systems: goal, environment, structure, activity, and evolution. Each dimension contains a set of criteria, which can be further detailed in sub-criteria. To assess the proposed solution artifact the following criteria were chosen: *Goal* - ***Efficacy***, and *Environment* - *Consistency with people* - ***Utility***. These criteria were considered the most adequate for the evaluation of the proposed artifact, since efficacy and utility are recognized as relevant criteria by several authors [14, 27, 32].

Applying these evaluation criteria to the solution artifact, the following results were obtained:

- *Goal* - ***Efficacy***: The proposed solution provides a view integration process capable of achieving the integration of distinct business process views. The taxonomy represents the consolidated information gathered from the different business process models.

- *Environment - Consistency with people* - ***Utility***: The proposed solution was demonstrated as being useful since it provides automatic assistance to the stakeholders for obtaining consolidated information from multiple views. Moreover, it provides the stakeholders with a systematic way for representing their concerns in the elements classification.

At this point, the solution artifact will be assessed against the four principles introduced by Österle et al.. According to the authors, DSR in IS must comply with four basic principles [26]:

- **Abstraction**: *Each artifact must be applicable to a class of problems* - The proposed solution is applicable to any business process model specified in BPMN 2.0.
- **Originality**: *Each artifact must substantially contribute to the advancement of the body of knowledge* - To our knowledge, there is a lack of scientific contributions regarding the application of view integration in the field of business process design.
- **Justification**: *Each artifact must be justified in a comprehensible manner and must allow for its validation* - The proposed solution is grounded in previously published work. The solution emerged from the research motivation and problem, from the solution objectives and relying on the literature review. Every step of this research is described and justified throughout the present document.
- **Benefit**: *Each artifact must yield benefit, either immediately or in the future, for the respective stakeholder groups* - The proposed solution artifact provides a structured approach for achieving the view integration of business process models and for the representation of the stakeholder concerns.

Chapter 6

Conclusion

In this research work, the solution artifact proposed to solve the problem addressed in this thesis corresponds to a process for the view integration of business process models. The solution artifact comprises a repository metamodel and an algorithm for the classification of the business process elements, which support the proposed view integration process.

The proposed view integration process aims to help process designers, stakeholders and organizations managing the multiple specifications often found for the same process. Moreover, it was meant to provide the possibility for the stakeholders to express their concerns in a structured manner. This was achieved by proposing an interactive process, which uses user input to perform the classification of the view elements to obtain the consolidated taxonomy.

To demonstrate the practical usage of the proposed solution artifact, an already existing modeling tool based on the BPMN.io JavaScript library was extended. The developed prototype is integrated with the EAMS repository which instantiates the proposed metamodel. The prototype was used to support the view integration process steps of the Auto Repair Company scenario.

6.1 Contributions

The main contribution of this research work was the proposed process for the view integration of business process models specified in BPMN 2.0. The contributions that support the proposed solution are the following:

- A metamodel for structuring information regarding business process models specified in BPMN 2.0.
- A mapping between the set of lane types and the corresponding dimensions.
- An algorithm for the classification of the model elements.
- A prototype of a modeling tool, developed in JavaScript, to demonstrate the practical use of the proposed solution.

The present work was communicated to the research community through a paper published in the European Mediterranean & Middle Eastern Conference on Information Systems (EMCIS) 2017 Proceedings, which was also presented in the same conference. At the conference, the paper "*View Integration of Business Process Models*" [7] won the Best Practical Paper Award.

6.2 Limitations

The main limitations identified in the proposed solution are the following:

1. It is assumed that the upload step of the integration process implies an approval workflow before the changes persist in the repository.
2. It is assumed that the different views do not contain contradictory information when they go through the classification step, which can be assured in the approval workflow.
3. It is assumed that, for each task, the values for the dimensions are clearly defined. However, in some tasks it is challenging to define its location (*where*) or the motivation (*why*) behind it. For instance, the task "Call customer" can take place all around the world. In certain cases, the values for the activity dimensions may not be clear and are subject to the interpretation of the user.

6.3 Future Work

Based on the research proposal and on the aforementioned limitations it is considered that future contributions to this problem should try to answer the following questions:

- *Are the 5W1H dimensions adequate for capturing the stakeholder concerns in business processes?*
- *Considering the view integration process, how could be defined a conflicts resolution step?*
- *How to automatically check consistency between the views of the same process?*
- *Is there an algorithm for generating business process model views in BPMN 2.0 based on the proposed taxonomy and repository metamodel?*
- *Can the proposed solution be used in organizations with complex process models? What is missing?*

The demonstration of the proposed solution artifact in a real organization regarding its utility would be of great value and could bring more interesting questions to be addressed in the scope of this problem. Moreover, it would also be interesting to define approval workflows and the corresponding roles inherent to the steps of uploading views and generating the consolidated model.

Bibliography

[1] G. Aagesen and J. Krogstie, "BPMN 2.0 for Modeling Business Processes," in *Handbook on Business Process Management 1: Introduction, Methods, and Information Systems*, J. vom Brocke and M. Rosemann, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 219–250.

[2] A. Awadid, "Supporting the consistency in multi-perspective Business Process Modeling: A mapping approach," in *2017 11th International Conference on Research Challenges in Information Science (RCIS)*, S. Assar, O. Pastor, and H. Mouratidis, Eds. Brighton, UK: IEEE, 2017, pp. 414–419.

[3] C. Batini and M. Lenzerini, "A Methodology for Data Schema Integration in the Entity Relationship Model," *IEEE Transactions on Software Engineering*, vol. SE-10, no. 6, pp. 650–664, 1984.

[4] C. Batini, M. Lenzerini, and S. B. Navathe, "A Comparative Analysis of Methodologies for Database Schema Integration," *ACM Comput. Surv.*, vol. 18, no. 4, pp. 323–364, 1986.

[5] A. Caetano, A. R. Silva, and J. Tribolet, "Business Process Decomposition – An Approach Based on the Principle of Separation of Concerns," *Enterprise Modelling and Information Systems Architectures - An International Journal*, vol. 5, no. 1, pp. 44–57, 2010.

[6] A. Caetano, C. Pereira, and P. Sousa, "Generation of Business Process Model Views," *Procedia Technology*, vol. 5, pp. 378–387, 2012.

[7] J. Colaço and P. Sousa, "View Integration of Business Process Models," in *Information Systems: 14th European, Mediterranean, and Middle Eastern Conference, EMCIS 2017, Coimbra, Portugal, September 7-8, 2017, Proceedings*, M. Themistocleous and V. Morabito, Eds. Cham: Springer International Publishing, 2017, pp. 619–632.

[8] T. Davenport and J. Short, "The New Industrial Engineering: Information Technology And Business Process Redesign," *Sloan Management Review*, vol. 31, no. 4, pp. 11–27, 1990.

[9] R. Dijkman, I. Vanderfeesten, and H. a. Reijers, "The Road to a Business Process Architecture: An Overview of Approaches and their Use," *Architecture*, vol. 350, no. July, pp. 1–17, 2011.

[10] ——, "Business process architectures: overview, comparison and framework," *Enterprise Information Systems*, vol. 7575, no. June 2015, pp. 1–30, 2014.

[11] M. Dumas, M. La Rosa, J. Mendling, and H. Reijers, *Fundamentals of Business Process Management*. Springer-Verlag Berlin Heidelberg, 2013.

[12] R. H. Eid-Sabbagh, R. Dijkman, and M. Weske, "Business process architecture: Use and correctness," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7481 LNCS, pp. 65–81, 2012.

[13] A. Ghose and G. Koliadis, “Auditing Business Process Compliance,” in *Service-Oriented Computing – ICSOC 2007: Fifth International Conference, Vienna, Austria, September 17-20, 2007. Proceedings*, B. J. Krämer, K.-J. Lin, and P. Narasimhan, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 169–180.

[14] A. R. Hevner, S. T. March, J. Park, and S. Ram, “Design Science in Information Systems Research 1,” *Design Science in IS Research MIS Quarterly*, vol. 28, no. 1, pp. 75–105, 2004.

[15] International Organization Of Standardization, *ISO/IEC 19510:2013 - Information technology - Object Management Group Business Process Model and Notation*, 2013.

[16] ——, *ISO/IEC/IEEE 42010:2011 - Systems and software engineering - Architecture description*, 2011.

[17] M. La Rosa, W. M. P. van der Aalst, M. Dumas, and F. P. Milani, “Business Process Variability Modeling: A Survey,” *ACM Comput. Surv.*, vol. 50, no. 1, pp. 2:1–2:45, 2017.

[18] M. Lankhorst, *Enterprise Architecture at Work: Modelling, Communication and Analysis*, 4th ed. Springer-Verlag Berlin Heidelberg, 2017.

[19] J. Mendling, “Foundations of Business Process Modeling,” in *Handbook of Research on Modern Systems Analysis and Design Technologies and Applications*, Mahbubur Rahman Syed and Sharifun Nessa Syed, Eds. Hershey, PA: IGI Global, 2008, ch. 15, pp. 189–222.

[20] J. Mendling and C. Simon, “Business Process Design by View Integration,” in *Business Process Management Workshops: BPM 2006 International Workshops, BPD, BPI, ENEI, GPWW, DPM, semantics4ws, Vienna, Austria, September 4-7, 2006. Proceedings*, J. Eder and S. Dustdar, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 55–64.

[21] E. D. Morrison, A. Menzies, G. Koliadis, and A. K. Ghose, “Business Process Integration: Method and Analysis,” in *Sixth Asia-Pacific Conference on Conceptual Modelling (APCCM 2009)*, ser. CRPIT, S. Link and M. Kirchberg, Eds., vol. 96. Wellington, New Zealand: ACS, 2009, pp. 29–37.

[22] S. B. Navathe and S. G. Gadgil, “A Methodology for View Inegration in Logical Database Design,” in *Eigth International Conference on Very Large Data Bases, September 8-10, 1982, Mexico City, Mexico, Proceedings*. Morgan Kaufmann, 1982, pp. 142–164.

[23] S. B. Navathe and M. Schkolnick, “View representation in logical database design,” in *Proceedings of the 1978 ACM SIGMOD international conference on management of data - SIGMOD ’78*. New York, New York, USA: ACM Press, 1978, p. 144.

[24] S. B. Navathe, R. Elmasri, and J. Larson, “Integrating User Views in Database Design,” *Computer*, vol. 19, no. 1, pp. 50–62, 1986.

[25] Object Management Group (OMG), *Model Driven Architecture (MDA) MDA Guide Revision 2.0 (ormsc/2014-06-01)*, 2014, no. June.

[26] H. Österle, J. Becker, U. Frank, T. Hess, D. Karagiannis, H. Krcmar, P. Loos, P. Mertens, A. Oberweis, and E. J. Sinz, “Memorandum on design-oriented information systems research,” *European Journal of Information Systems*, vol. 20, no. 1, pp. 7–10, 2011.

[27] K. Peffers, T. Tuunanen, M. Rothenberger, and S. Chatterjee, “A Design Science Research Methodology for Information Systems Research,” *J. Manage. Inf. Syst.*, vol. 24, no. 3, pp. 45–77, 2007.

[28] C. M. Pereira, “Using an Organizational Taxonomy to Support Business Process Design,” Ph.D. dissertation, Instituto Superior Técnico, 2011.

[29] C. M. Pereira and P. Sousa, “Business Process Modelling Through Equivalence of Activity Properties,” in *Proceedings of the Tenth International Conference on Enterprise Information Systems - Volume 3: ICEIS*, INSTICC. Barcelona, Spain: ScitePress, 2008, pp. 137–146.

[30] C. M. Pereira, A. Caetano, and P. Sousa, “Ontology-Driven Business Process Design,” in *Building the e-World Ecosystem: 11th IFIP WG 6.11 Conference on e-Business, e-Services, and e-Society, I3E 2011, Kaunas, Lithuania, October 12-14, 2011, Revised Selected Papers*, T. Skersys, R. Butleris, L. Nemuraite, and R. Suomi, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 153–162.

[31] ——, “Using a Controlled Vocabulary to Support Business Process Design,” in *Enterprise and Organizational Modeling and Simulation: 7th International Workshop, EOMAS 2011, held at CAiSE 2011, London, UK, June 20-21, 2011. Selected Papers*, J. Barjis, T. Eldabi, and A. Gupta, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 74–84.

[32] N. Prat, I. Comyn-Wattiau, and J. Akoka, “Artifact Evaluation in Information Systems Design Science Research - A Holistic View,” *PACIS 2014 Proceedings*, vol. Paper 23, pp. 1–16, 2014.

[33] P. Sousa, C. Pereira, R. Vendeirinho, A. Caetano, and J. Tribolet, “Applying the Zachman Framework Dimensions to Support Business Process Modeling,” in *Digital Enterprise Technology: Perspectives and Future Challenges*, P. F. Cunha and P. G. Maropoulos, Eds. Boston, MA: Springer US, 2007, pp. 359–366.

[34] J. F. Sowa and J. A. Zachman, “Extending and formalizing the framework for information systems architecture,” *IBM Systems Journal*, vol. 31, no. 3, pp. 590–616, 1992.

[35] The Open Group, *TOGAF Version 9.1*, 10th ed. Reading, UK: Van Haren Publishing, 2011.

[36] W. M. P. van der Aalst, “Business Process Management: A Comprehensive Survey,” *ISRN Software Engineering*, vol. 2013, pp. 1–37, 2013.

[37] ——, “Business Process Simulation Survival Guide,” in *Handbook on Business Process Management 1: Introduction, Methods, and Information Systems*, J. vom Brocke and M. Rosemann, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 337–370.

[38] M. Weidlich and J. Mendling, “Perceived consistency between process models,” *Information Systems*, vol. 37, no. 2, pp. 80–98, 2012.

[39] J. A. Zachman, “A Framework for Information Systems Architecture,” *IBM Systmes Journal*, vol. 26, no. 3, pp. 454–470, 1987.