Pontifical Catholic University of Paraná
Graduate Program in Informatics - PPGIA

Luiz Fernando Puttow Southier

A framework for multifactor process mining

Curitiba - PR, Brazil
2023

Luiz Fernando Puttow Southier

A framework for multifactor process mining

This thesis was presented to the Graduate Program in Informatics - PPGIa (in Portuguese: *Programa de Pós-Graduação em Informática*) of the Pontifical Catholic University of Paraná - PUCPR (in Portuguese: *Pontifícia Universidade Católica do Paraná*) as a partial requirement to obtain the title of Doctor in Informatics.

Pontifical Catholic University of Paraná

Graduate Program in Informatics - PPGIA

Supervisor: Prof. Dr. Edson Emilio Scalabrin

Curitiba - PR, Brazil

2023

Dados da Catalogação na Publicação
Pontifícia Universidade Católica do Paraná
Sistema Integrado de Bibliotecas – SIBI/PUCPR
Biblioteca Central
Luci Eduarda Wielganczuk – CRB 9/1118

Southier, Luiz Fernando Puttow
S727f A framework for multifactor process mining / Luiz Fernando Puttow Southier ;
2023 orientador: Edson Emilio Scalabrin. – 2023.
197 f. : il. ; 30 cm

Tese (doutorado) – Pontifícia Universidade Católica do Paraná, Curitiba, 2022
Bibliografia: f. 137-158

1. Informática. 2. Mineração de processos. 3. Negócios – Processamento de dados – Administração. I. Scalabrin, Edson Emilio. II. Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Informática. III. Título.

CDD. 20. ed. – 004

Pontifícia Universidade Católica do Paraná
Escola Politécnica
Programa de Pós-Graduação em Informática

Curitiba, 30 de março de 2023.

25-2023

DECLARAÇÃO

Declaro para os devidos fins, que **LUIZ FERNANDO PUTTOW SOUTHIER** defendeu a tese intitulada "**A FRAMEWORK FOR MULTIFACTOR PROCESS MINING"**, no dia 28 do mês de março de 2023, o qual foi aprovado.

Por ser verdade firmo a presente declaração.

Prof. Dr. Emerson Cabrera Paraiso
Coordenador do Programa de Pós-Graduação em Informática

Rua Imaculada Conceição,1155 Prado Velho CEP 80215-901 Curitiba Paraná Brasil
Fone: (41) 3271-1669 e-mail: secretaria@ppgia.pucpr.br

Dedico essa tese aos incontáveis esforços e ao apoio
incondicional dos meus pais e familiares.
Sem eles, e sem sua fé em mim, chegar aqui não seria possível.

To the people of science.

Agradecimentos

[1] Inicialmente, deixo meus agradecimentos à minha família: ao meu pai, Antônio, à sua esposa, Marlei, e à minha mãe, Marli, que forneceram todo o suporte necessário para o meu desenvolvimento como pessoa e sempre me apoiaram em minha trajetória. Agradeço também aos meus familiares e amigos: Francisca e Francisco, Glaucya, Greicy, Elza, Willian, Jandira, Gustavo, Marcos, Eduardo, Patrícia, Juliana, Alana, Cíntia, Thaylline, Henrique e Jheison pelas conversas e desabafos que contribuíram para guiar o meu caminho.

Especialmente agradeço ao meu orientador, professor Edson Scalabrin, que sempre compreendeu minhas habilidades e limitações direcionando com maestria os diferentes aspectos do trabalho de pesquisa desenvolvido. Notadamente, agradeço à empresa parceira do projeto de doutorado UpFlux, em especial ao professor Cleiton dos S. Garcia, ao Gilberto e à Esther.

Deixo também meus agradecimentos aos professores do PPGIA, do PPGTS e do PPGEPS, que contribuíram com a interdisciplinaridade desse projeto, especialmente aos professores Eduardo, Deborah, Claudia e Sandro. Agradeço aos meus amigos e colegas do grupo de pesquisa em Mineração de Processos: Denise, Jair, Sheila e Eduardo. Ainda, à equipe do hospital Cajuru.

Agradeço aos meus ex-orientadores professores Donizzeti, Teodora, e Marco e a todo o corpo docente da UTFPR pelo papel ímpar no meu desenvolvimento profissional. Agradeço especialmente ao meu amigo professor Marcelo Teixeira pela colaboração essencial na minha formação como pesquisador e pelas contribuições na validação deste projeto de pesquisa. Ainda, aos professores do IECP e da Penn State, e à equipe da NYU e ao professor Dennis Sasha.

Ainda, agradeço às políticas públicas de inclusão social, difusão do conhecimento, fomento ao ensino, pesquisa e extensão que estiveram presentes em minha trajetória, bem como, aos homens e mulheres da ciência. Agradeço ainda à sociedade brasileira pelo investimento indireto neste trabalho, por meio do governo federal do Brasil. Agradecimentos especiais ao CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) pela bolsa do Programa Doutorado Acadêmico para Inovação.

[1] This page presents the acknowledgments in Portuguese

“Gods molded living creatures from clay, and titans were tasked to distribute qualities to them. Men were left naked and unprotected, unable to defend themselves in this hostile world. Prometheus stole the ***techne****, the fire of creative power, from the workshop of Athena and gave it to humankind. He suffered eternal torment for his transgression to the gods. With the techne, humanity thrived with technology, knowledge, and civilization.* *(RAGGIO, 1958)*

Abstract

Technological advances in the last years have enabled companies, organizations, and people to use digital systems to support their processes. These systems keep records of processes data: actions taken within the process, the time these actions occurred, who executed them, what kind of resources are used to execute them, etc. *Process mining* is a research area that uses these records to analyze in detail the processes. Process mining provides techniques and tools to: create process models (discovery); identify deviations in recorded data (conformance); analyze processes considering other perspectives (enhancement); make predictions and recommendations (operational support). All these techniques and tools have been applied to several areas. Financial costs have a significant role in organizations, and they are constantly seeking cost-effective improvements for their business processes. Cost-aware process mining approaches have been proposed in the literature. However, the heterogeneity of the approaches, and the difference between the systems where they are implemented, hinder the use of cost-aware techniques. This is the first problem addressed by this thesis. Secondly, some types of processes require a *multifactor perspective*, being not only monetary cost of interest but other factors such as the health rate in a healthcare process, grades and attendance in an educational process, and quality or productivity indicators in a manufacturing process. This is the second problem addressed by this thesis. The main goal of this thesis is to present a computational framework that enables multifactor process mining. This framework supports the following tasks: modeling of multifactors based on aspects of the process such as duration, the occurrence of events, traces, and data attributes; annotation of modeled multifactors in the event log; construction of reports based on multifactor information; factor-based prediction, recommendation, conformance checking, and process model enhancement using the multifactor information; and representation of annotated event log with multifactor information as a data frame suitable to other data mining activities. The framework was validated on three case studies in healthcare, educational, and telecommunications domains.

Keywords: Process Mining. Business Process Management. Cost-aware. Multifactor Process Mining.

List of Figures

List of Frames

List of Definitions

List of Abbreviations and Acronyms

KPI	Key Performance Indicators
DAI	Ph.D. Project for Innovation (from Portuguese: *Doutorado Acadêmico para Inovação*)
MCTI	Brazilian Ministry of Science, Technology, and Innovation (from Portuguese: *Ministério da Ciência, Tecnologia e Inovações*)
CNPq	National Council for Scientific and Technological Development (from Portuguese: *Conselho Nacional de Desenvolvimento Científico e Tecnológico*)
PUCPR	Pontifical Catholic University of Paraná (from Portuguese: *Pontifícia Universidade Católica do Paraná*)
PPGIa	Graduate Program in Informatics (from Portuguese: *Programa de Pós-Graduação em Informática*)
ICD	International Classification of Diseases
YAWL	Yet Another Workflow Language
UML	Unified Modeling Language
EPC	Event-driven Process Chain
DFG	Directly-follows Graph
XES	IEEE Standard for eXtensible Event Stream
URI	*Uniform Resource Identifier*
ABC	Activity-Based Costing
TDABC	Time-Driven Activity-Based Costing
DPN-net	Petri net with data
DSRM	Design Science Research Method
CSDSRM	Case Study Design Science Research Method
KDE	Kernel Density Estimate
IG	Information Gain

RCA	Root Cause Analysis
SIGTAP	Brazilian Management System for the Unified Health System Tables (from Portuguese: *Sistema de Gerenciamento da Tabela de Procedimentos, Medicamentos, Órteses, próteses e meios auxiliares de locomoção do Sistema Único de Saúde*)

Contents

1 Introduction

With the advance of technology in the last years, companies, organizations, and people have increasingly migrated to digital approaches. This advance allows electronic devices to be connected to the Internet facilitating the growth of the *Internet of Things* in several environments such as hospital information systems and manufacturing technologies (MADAKAM et al., 2015). *Industry 4.0* is an example of this. Resource efficiency and further improvements to mechanization and automation are features that are intended to be achieved by it (LASI et al., 2014).

This technological advance has enabled companies, organizations, and people, independent of the area, to use digital systems to support their processes (SLYWOTZKY; MORRISON; WEBER, 2001). These systems keep records of process data, representing the behavior of "real-world" processes in the "digital world". These records, named *event logs*, can include the actions taken within the process, the time they occurred, who executed them, what kind of resources are used to execute them, etc. Additionally, organizations need to adjust their business processes along with the changing environments to maintain a competitive advantage (BEEST; MARUSTER, 2007). Therefore, they are challenged with tracking and optimizing organizational processes to support their businesses.

Process mining is an emerging research area that uses the records of process data created by digital systems to analyze the processes in detail. Process mining provides techniques and tools to create processes models by using recorded data (discovery); confront recorded data with reference models to identify deviations (conformance); enhance and analyze processes considering other perspectives (enhancement); detect deviations, make predictions and recommendations in real-time processes (operational support); etc. All these techniques and tools have been applied to several areas, such as: healthcare (ROJAS et al., 2016), information and communications technology (GUPTA; SEREBRENIK; JALOTE, 2017), manufacturing (LORENZ et al., 2021), education (BOGARÍN; CEREZO; ROMERO, 2018), finance (WERNER; WIESE; MAAS, 2021), logistics (BECKER; INTOYOAD, 2017), security (AALST; MEDEIROS, 2005), telecommunication (DAKIC et al., 2018), and other areas (GARCIA et al., 2019).

Even with a range of methods, techniques, and tools for analyzing, managing, and optimizing processes, the process mining initiatives tend to focus on time and resource inefficiencies rather than directly on *cost* inefficiencies (WYNN et al., 2013a). Cost has a major role in organizations, and they are constantly seeking cost-effective improvements for their business processes (AALST, 2013). However, in most organizations, high-level cost-based decisions are made separately from process-related operational decisions be-

cause of the limited system-based support for cost at the process level.(WYNN et al., 2013b; ADAMS et al., 2015). The lack of integration between process model and cost information hampers better decision-making support towards cost reduction(THABET; GHANNOUCHI; GHEZALA, 2018).

According to Thabet, Ghannouchi and Ghezala (2022), in the last years, cost considerations have been increasingly incorporated into process mining techniques, but several challenges are yet to be overcome. For instance, in some areas like the healthcare domain, costs for service and overhead components are commonly recorded at the patient level, hiding activity-level details (LEEMANS et al., 2022).

Another challenge is that only the financial cost perspective may not be enough to analyze a process properly, requiring a *multifactor perspective* to be incorporated in the analysis (HONG, 2016). This thesis assumes that a specialist may want to annotate and analyze the process considering multifactors; that is, cost is one of the factors that may be of interest. This assumption is based on the business process literature in which Key Performance Indicators (KPI) are used to evaluate the success of an organization or of a particular activity or process (HUGHES; BARTLETT, 2002).

Another aspect is that factors are process or context-dependent. For example, Hong (2016) presents a process mining study that considers cost perspective along with quality factors in a manufacturing process. Other areas can have different multifactors. In healthcare processes where treatment options are available, cost is only one factor for analyzing which option is better. Bodenheimer and Sinsky (2014) assumes that along with costs, healthcare analysts should consider the health rate of the treatment, the patient experience rate, and the work life of healthcare providers. Educational processes may consider, along with the cost a student has to the educational system, factors such as grades, attendance, and time spent to graduate (JAMOLIDDINOVICH, 2022; GREEN, 1994). Customer service process analysis may be subject to factors such as the rate of first call resolution or the dropout rate (ABDULLATEEF; MOKHTAR; YUSOFF, 2011).

Considering that cost-aware processes may be analyzed from a multifactor (cost and/or other factors) perspective, the following section describes the studies related to the cost-aware process mining area, detailing what has been proposed in the last years and who are the most frequent authors. Section 1.2 presents the scope and contribution of this thesis and the objectives. Section 1.3 details the broader scope of the doctoral project, which includes other activities and results as part of an innovation project. Section 1.4 presents the document structure and reading scheme.

1.1 Related studies

By performing a literature review, several studies have been identified in cost-aware process mining. Figure 1.1 presents the main related studies. Each node ⬭ represents a related study. The nodes are distributed by year, from 2011 to 2022. The connections between nodes represent a citation from one study to another: Nauta (2011) - 14 citations, Wynn et al. (2013a) - 5 citations, Wynn et al. (2013b) - 7 citations, Wynn et al. (2014) - 13 citations, other studies - less than 5 citations.

On the left of Figure 1.1, the five most frequent authors are itemized, and an indication is drawn next to the years the author has contributed in one or two studies from that year. *M. Wynn* is the most relevant author in the field, with 8 studies spawning from 2011 to 2022. *H. Ghezala*, *S. Ghannouchi*, and *D. Thabet* have published studies together since 2014 (6 studies). *A. ter Hofstede* has contributed from 2013 to 2016 only as a second author, and *W. Low* in the same period.

Nauta (2011) is the first work to deal with cost-aware Process Mining specifically. This work takes into account existing cost reduction techniques within management accounting, such as activity-based costing and time-driven activity-based costing, and extends them to process mining. It presents the XES cost extension and shows how to create a cost model using cost drivers and how to annotate the event log with cost information.

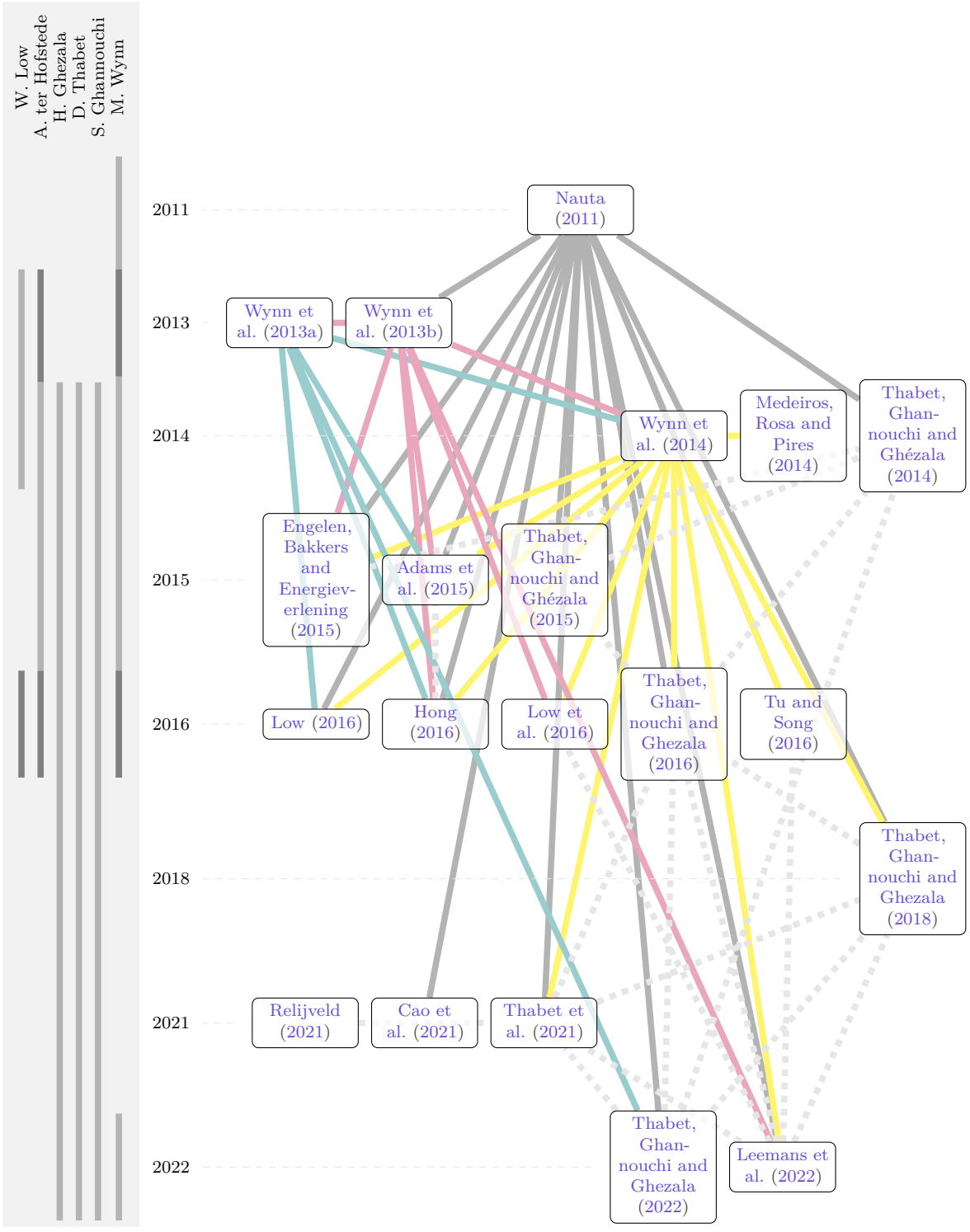
Wynn et al. (2013a) describes a research agenda that proposes an explicit link between cost and processes to all phases of the business process management life cycle. The study discusses some research challenges that must be addressed to realize the goal. Explicitly, the 4 research questions are: How can process-related cost information enrich process design? How can cost-aware business process execution environments be realized? How can cost information enhance operational support? How can business processes be diagnosed from a cost perspective?

Wynn et al. (2013b) describes how cost considerations can guide process-related decisions at the operational level. The paper presents the conceptual framework, data requirements, and technical challenges that need to be addressed to realize cost-informed workflow execution. A prototype was implemented on the YAWL workflow environment.

In Wynn et al. (2014), a framework is proposed to expand the work of Nauta (2011). The framework includes prediction and reporting features. The reports have a style akin to reports in the area of management accounting. Prediction is achieved by merging cost data with historical data from event logs.

Medeiros, Rosa and Pires (2014) proposes a metamodel for expressing costs in service compositions. The proposed metamodel allows designers to express different cost behaviors and to enforce them throughout the service composition lifecycle. Adams et al.

Figure 1.1 – Related studies of cost-aware process mining



(2015) describes the different ways a workflow management system can support process-related decisions guided by cost information. The study defines the criteria a workflow management system should meet to provide such support and discusses an implementation within the YAWL workflow environment.

Engelen, Bakkers and Energieverlening (2015) designs an approach that describes how financial information should be determined and included in BPMN models. To that end, the BPMN modeling language was extended with financial information from the RCA accounting method. The applicability of the approach was validated in a real case. Hong (2016) suggests a framework for performance analysis focusing on cost and quality perspective. Specifically, the contributions of the study are: to suggest a method to extend event log of manufacturing process with cost and quality; to analyze manufacturing information; to predict manufacturing cost; and to enable quality report in manufacturing process.

Low (2016) and Low et al. (2016) investigate a way to identify potential efficiency gains in business processes by considering historical information on time, cost, and resource utilization. The paper proposes some optimization techniques to explore and assess alternative execution scenarios. A hybrid genetic algorithm-based approach had the best result in an experimental evaluation.

Tu and Song (2016) proposes a framework to analyze and predict manufacturing costs by extending existing process mining techniques. The study performs cost prediction based on production volume and time prediction using the working progress of manufacturing processes. Cao et al. (2021) proposes an approach to predict medical expenses based on Process Mining. The authors propose a dynamic-medical-path-net that considers the repetition times of nodes to predict medical expenses. The proposed method found about 25% improvements to the other methods.

Relijveld (2021) uses Process Mining to analyze differences in care provided to colorectal cancer patients and the associated costs. This study used real-world de-identified patient data from patients treated within three Australian hospitals. To investigate differences between care and costs of care, this research uses linked data and evaluates the entire pathway. Reports are presented.

Leemans et al. (2022) introduces a new process model containing trace data that can be used in individual-level or cohort-level decision-analytical model building. Furthermore, it enhances these models with process-based micro-costing estimations. The approach was evaluated by health economics and decision modeling experts.

Thabet, Ghannouchi and Ghézala (2014) proposes an approach to associate Petri Net models with cost information using a process mining extension technique. A test case is performed. Thabet, Ghannouchi and Ghézala (2015) proposes several improvements to the previous study and extensions of the proposed solution to enhance the provided decision-making support. These proposals include cost data structuring, description, and analysis of the recommendations from talks with experts.

Thabet, Ghannouchi and Ghezala (2016) proposes an improved version of Thabet,

Ghannouchi and Ghézala (2015) that includes cost data description and analysis at both activity and business process levels. Also, the study describes the implementation and tests of the improved solution on Petri Nets, Event-driven Process Chain, and Business Process Model and Notation. The cost data analysis was performed through classification algorithms that can be selected by the user. However, the lack of support during this selection may affect the accuracy of the obtained results. Furthermore, the performance of the same classification algorithm may vary from a case to another depending on its context. Considering this, Thabet, Ghannouchi and Ghezala (2018) proposes a context-based cost data analysis allowing to select and apply the classification algorithm the most suited to the case at hand. The approach was validated in a Tunisian clinic.

Thabet et al. (2021) is an improvement of Thabet, Ghannouchi and Ghezala (2018). The authors improve the cost data analysis approach by including the control-flow perspective. In Thabet, Ghannouchi and Ghezala (2022), the authors extend the 2021 work to include cost perspective at the activity level.

Table 1.1 summarizes the topics approached by the articles. Several studies address some level of cost modeling, either along the process model or separated. Only two studies address cost annotation. Only Hong (2016) refers to other factors besides cost because it uses manufacturing quality factors in the study. However, it does not include the annotation of quality factors or the possibility of associating other factors. Report, prediction, conceptual discussion, and decision analysis consider only the cost factor.

Frame 1.1 – Related studies topics

Study	Model	Annotation	Report	Prediction	Discussion	Decision ana.	Other factors
Nauta (2011)	✓	✓					
Wynn et al. (2013a)					✓		
Wynn et al. (2013b)	✓	✓					
Thabet, Ghannouchi and Ghézala (2014)	✓						
Medeiros, Rosa and Pires (2014)	✓						
Wynn et al. (2014)			✓	✓			
Adams et al. (2015)					✓		
Engelen, Bakkers and Energieverlening (2015)	✓						
Thabet, Ghannouchi and Ghézala (2015)						✓	
Hong (2016)			✓	✓			✓
Low et al. (2016)						✓	
Low (2016)						✓	
Thabet, Ghannouchi and Ghezala (2016)	✓					✓	
Tu and Song (2016)				✓			
Thabet, Ghannouchi and Ghezala (2018)	✓					✓	
Cao et al. (2021)				✓			
Relijveld (2021)			✓				
Thabet et al. (2021)	✓					✓	
Leemans et al. (2022)						✓	
Thabet, Ghannouchi and Ghezala (2022)	✓					✓	

1.2 Research contribution and scope

Considering what was described in the last section, the research agenda proposed by Wynn et al. (2013a) is still relevant nowadays. First, as shown in Frame 1.1, no study addresses all cost-related activities: model, annotation, prediction, reporting, and decision analysis. For instance, Thabet, Ghannouchi and Ghézala in their five studies use a specific type of process model with cost information that is not suitable to Wynn et al. (2014) prediction and reporting activities. The heterogeneity of the approaches, and the difference between the systems where they are implemented, hinder the use of cost-aware techniques. This is the first gap in which this thesis presents its scientific contributions.

Secondly, as described before, some types of processes may require a *multifactor perspective*, being not only the financial cost of interest but other factors and indicators. As presented in Frame 1.1, only one study considers other factors besides cost, and in that case, the factor is already annotated in the log. No study presents a multifactor capability of modeling and annotating factors to the log and supporting report, prediction, and other activities with multifactors. This leads to the second gap in which this work presents its contribution.

The research contribution of this work is presented as a goal to answer the following research question:

How to include multifactors when performing process mining activities?

In other words: how to *model* multifactors in a way they are associated with process elements (events, activities, duration, cases, etc.)? How to *annotate* multifactors to the event log? How to *report* multifactors? How to *predict* the value of a factor? How to *recommend* based on multifactors? How to check *conformance* based on factors' values? How to *enhance* process models based on a factor? How to represent multifactor in a way suitable for *decision analysis*?

The above research question is inspired by the two gaps presented previously: the heterogeneity of implementation of the cost-aware techniques; and the absence of multifactors in these techniques. To contribute towards solving these gaps and answering the research question, this thesis proposes a framework that supports multifactor process mining activities. With this framework, the analyst can perform several process mining activities, considering the cost perspective: modeling and annotating cost information, predicting, recommending, and reporting costs. The analyst can also perform these activities considering other factors, along with cost. With this tool, managerial decisions based on costs and other factors can be made along with process-related decisions.

1.2.1 Objective

The objective of this thesis is to present a computational framework that enables multifactor process mining. This framework supports the following tasks: modeling of multifactors based on aspects of the process such as duration, the occurrence of events, traces, and data attributes; annotation of modeled multifactors in the event log; construction of reports based on multifactor information; factor-based prediction, recommendation, conformance checking, and process model enhancement using the multifactor information; and representation of annotated event log with multifactor information as a data frame suitable to other data mining activities. To that end, the following specific objectives are presented:

- Create an event log *extension* that includes multifactors; and *the multifactor model*.
- Design the *annotator* component that uses the multifactor model and the event log to create a multifactor-annotated event log;
- Design the following components: *factor-based coloring* component that enables the enhancement of process models with multifactor information; *factor-based conformance check* component that evaluates the conformance of the log based on multifactor values and exports the conform and not-conform logs; *reporting* component that enables the creation of several types of reports and charts, including multifactors; *prediction/recommendation* component that supports the prediction of a factor's value and activity recommendation towards a certain goal; and *data mining* component that supports the representation of the multifactor-annotated event log in a format suitable to other data mining activities;
- Specify each component's interface (inputs and outputs) and give examples of use;
- Validate the components of the framework in real-world cases.

1.3 Scope of the doctoral project

This thesis is part of Ph.D. Project for Innovation (DAI[1]) subject to a wider scope. The DAI proposal was motivated by the interest in offering doctoral students the opportunity to develop research projects more applied to the reality and needs of the Brazilian industrial sector, as well as offering industries the benefits of research and high-level development. With this, the CNPq intends to: contribute to the formation of human resources for applied research, technological development, and innovation; encourage innovative projects that present technological risk through academic research; stimulate the creation of partnership networks between universities and companies for the execution of innovative research and

[1] from Portuguese: *Doutorado Acadêmico para Inovação*

technology projects; assist companies in the development or improvement of products, processes, and services that favor the advancement of strategic economic sectors; in addition to promoting actions of education, popularization and/or scientific dissemination (CNPQ, 2023).

The DAI project is developed as a partnership between the Brazilian Ministry of Science, Technology, and Innovation (MCTI[2]), represented by the National Council for Scientific and Technological Development (CNPq[3]); the Pontifical Catholic University of Paraná (PUCPR[4]), represented by the Graduate Program in Informatics (PPGIa[5]); and the Process Mining company Upflux (UPFLUX, 2023). In this broader scope, several research contributions were developed intending the improvement of the company's products, processes, and services. The research contribution related to this thesis, the multifactor framework, is one of these. The other research contributions are briefly described below and are presented in the Appendix. *The reader may skip the appendices without prejudice to understand this thesis.*

- Appendix A presents a systematic mapping review on log preparation for Process Mining. Since the start point for Process Mining is using event logs to analyze processes, these event logs need to be extracted from databases and prepared for use. The quality of the event logs used as input is critical to the success of any Process Mining effort. In that sense, a systematic mapping review provides the reader with highlights of the state-of-the-art techniques for event log preparation. Based on the retrieved studies, we identified six main categories of log preparation techniques: extraction, cleaning, repair, non-adequate granularity, quality evaluation, and privacy. The results are explored quantitatively and qualitatively. All results are made available through spreadsheets and charts. The research is a starting point for researchers to identify the studies that would help them prepare event logs for Process Mining.

- Appendix B presents a method to extract healthcare glossing rules from event logs. Based on the services provided to a patient in a healthcare appointment, auditors determine whether that service will be glossed, partially glossed, or not glossed. This determination is made through glossing rules. Using an event log, the method can measure the association between a set of services provided with the glossing performed (total, partial, or none) and create a set of rules.

- Based on the services provided to a patient during an appointment, a given International Classification of Diseases (ICD) is associated with it by a healthcare

[2] from Portuguese: *Ministério da Ciência, Tecnologia e Inovações*
[3] from Portuguese: *Conselho Nacional de Desenvolvimento Científico e Tecnológico*
[4] from Portuguese: *Pontifícia Universidade Católica do Paraná*
[5] from Portuguese: *Programa de Pós-Graduação em Informática*

professional. However, the ICD information is often not entered due to external factors. Appendix C presents research about a system that automatically fills in ICDs for cases where this information is missing. For this, the history of cases in which ICD was informed was used.

- The partner company uses a JSON custom structure for storing process models. Appendix D presents a method for converting process models in this standard to the BPMN standard. The method is configurable to draw activities, gateways, nodes, and subprocesses according to the user's desire.

1.4 Document structure and reading scheme

This thesis is composed of the following parts:

- Part I shows how Process Mining activities are performed and what results the user can obtain from them. This part focuses on the "classical" and "cost-aware" Process Mining frameworks; that is, it does not focus on multifactors. Chapter 2 presents the classical Process Mining Framework. *In case the reader is familiar with process mining activities, this chapter may be skipped.* Chapter 3 presents the relevant cost-related definitions and process mining techniques.

- Part II focus on the Multifactor Process Mining Framework. Chapter 4 describes the research method used and how to create the multifactor model and represent multifactors on the event log. Chapter 5 presents the framework components, with their interface and examples of use.

- Part III shows the case studies performed to validate the framework. Chapter 6 explores the finding of a case study on a real-world Brazilian telecommunication company. Chapter 7 presents a case study on an educational dataset from a Brazilian public university. Chapter 8 presents an ongoing case study on the healthcare domain.

Part I

Process Mining

This part shows how Process Mining activities are performed and what kind of results the user can obtain from them. This part focuses on the "classical" and "cost-aware" Process Mining frameworks; that is, it does not focus on multifactors.

■ Chapter 2 presents the classical Process Mining Framework. It details the theory of each process mining activity: read the recorded data and generate the process model; compare recorded data with reference model; enhance process model; and support operational actions such as prediction, recommendation, and fault detection. For that purpose, mainly the seminal work of Aalst (2016) is used and complemented with up-to-date studies. *In case the reader is familiar with process mining activities: discovery, conformance, enhancement, and operational support, this chapter may be skipped.*

■ Chapter 3 presents the Cost-aware Process Mining techniques based on the related studies in the literature. This chapter shows how: cost models can be created from accountability rules; to estimate costs in a process; associate costs with event data; compare cost data with reference cost models; predict and recommend based on costs; and what cost-related results can be obtained.

2 Process Mining Framework

In 1965, Moore et al. (1965) proposed the famous Moore's law: the number of transistors that would fit on a given area in an integrated circuit would double every year. This rapid growth of computational capabilities allowed the data stored and exchanged electronically to grow spectacularly in the last fifty years (AALST et al., 2011a). This growth of the "digital world" enables the digital recording of several real-world *events*, such as a transfer in a banking account (MOSTAFAEE et al., 2019); a doctor prescribing some medicine for a patient (PARTINGTON et al., 2015); a school teacher applying an online test to their students (BOGARÍN et al., 2014); a person buying some product online (DOGAN; FERNANDEZ-LLATAS; OZTAYSI, 2019); etc.

These events are recorded by *Information Systems* and represent actions taken by real-world entities such as companies, people, machines, and organizations. These events are taken within the context of a certain *process* related to these entities' objective. For instance, the event of a doctor prescribing medicine for a certain patient is an *activity* taken within a treatment process of that patient. This process can involve other events, that is, the patient's admission to the hospital, a certain blood test taken by this patient, the transfer to another hospital, and the patient's discharge (ROJAS et al., 2016).

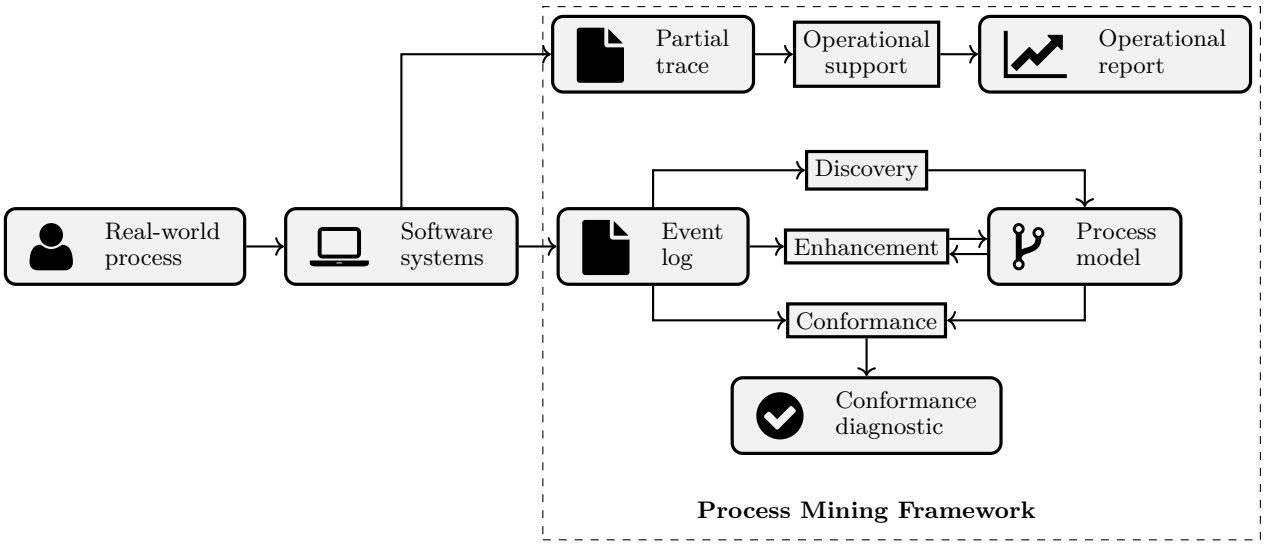
The increasing amount of event data generated by information systems encourages research subjects like *data science*. According to Aalst (2014), data scientists assist organizations in turning data into value, answering various data-driven questions such as: (Reporting) What happened? (Diagnosis) Why did it happen? (Prediction) What will happen? (Recommendation) What is the best that can happen? Unfortunately, the process perspective of real-world entities is absent in many data science techniques. Researchers argue that event data should be used to improve end-to-end processes. Process mining is the research subject that links data science to process modeling, and analysis (AALST et al., 2011a; AALST, 2016).

Therefore, *Process Mining* can turn event data into value, answering process-related questions (MANS et al., 2012). Figure 2.1 shows how process mining interacts with the real and digital world. Actions occur in the real world within processes executed by entities (companies, machines, organizations, people, etc.). These actions are supported by information systems that record them as events in *event logs* (AALST, 2016). Process mining takes the recorded events in event logs, and models/analyzes the real-world processes. For that, it uses *process models*. Process mining can also suggest changes (specifications, implementations, configurations) in the information systems that support those processes.

Figure 2.1 also shows the three main types of process mining: discovery, enhance-

ment, and conformance. *Discovery* takes as input the recorded actions from real-world processes - the event log - and generates a process model. *Enhancement* uses a previously generated process model and an event log to create an enhanced version of this model. *Conformance* compares a created or previously generated process model with an event log, identifying deviations and generating a diagnostic. Furthermore, process mining can be used for *operational support*: event logs are used to predict, recommend and detect violations in ongoing processes.

Figure 2.1 – Positioning of process mining



Source: adapted from Aalst (2016)

This chapter describes process models and event logs detailing the three types of process mining. Section 2.1 presents the model languages and information used to build process models. Section 2.2 explores how activities from real-world processes are recorded as events in event logs. Section 2.3 shows how an event log can be used to discover a process model. Section 2.4 explains how an event log can be compared to a process model to check its conformance. Section 2.5 details how to add perspectives to process models creating a enhanced process model version. Finally, Section 2.6 presents information about operational support in process mining.

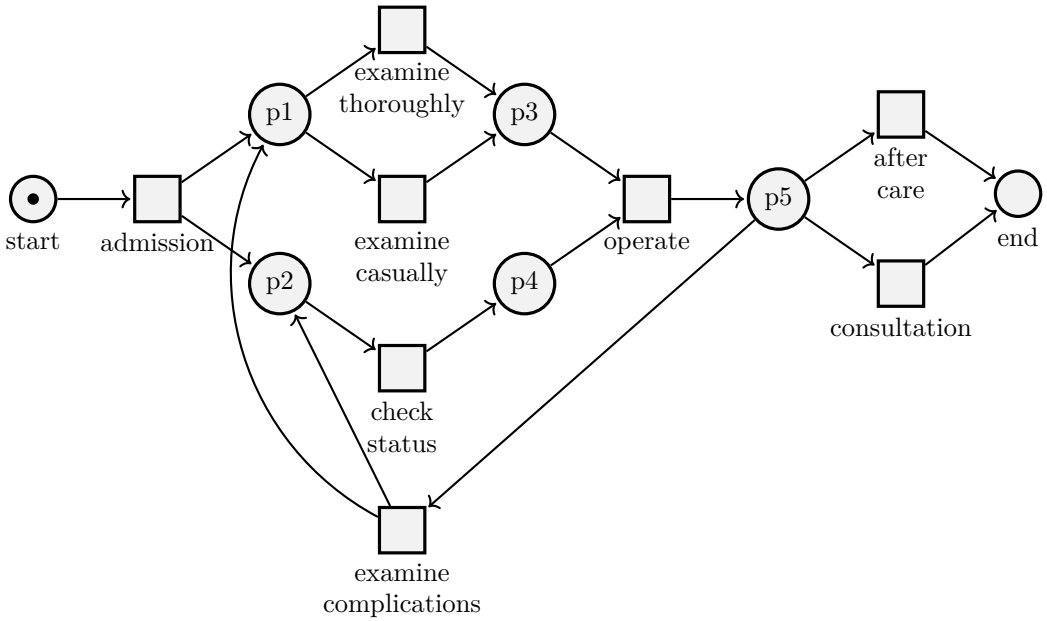
2.1 Process models

Figure 2.1 shows that process mining starts from event data and uses process models in several ways, e.g., process models are discovered from event logs, used as reference models (conformance), or used to enrich the process. Several notations exist in the literature to model the real-world process, such as Transition Systems, Petri nets (JENSEN; KRISTENSEN, 2009), Workflow Nets (AALST, 1998), Yet Another Workflow Language (YAWL) (HOFSTEDE et al., 2009), BPMN (MODEL, 2010), Unified Modeling Language

(UML), Event-driven Process Chain (EPC) (SCHEER, 2012), Causal Nets and Process Trees. All these notations are referred to as *process models*.

Process models describe the process in terms of the control flow, i.e., the ordering of activities and their casual dependencies. They can also include temporal properties, specify the creation and use of data, model decisions, and stipulate how resources interact with the process (AALST, 2016). Figure 2.2 shows a certain healthcare process model by a Petri net. The model is not intended to be realistic and only aims to show the different control-flow constructs in a healthcare setting.

Figure 2.2 – A Petri net modeling a healthcare process



Source: Mans, Aalst and Vanwersch (2015)

Petri nets are a bipartite graph of *transitions* (□) representing activities and *places* (○) representing states. Places can contain a certain amount of *tokens* (●). Transitions are fired, consuming one token from each input place and producing one token for each output place. If the necessary number of tokens in the input places is unavailable, the transition cannot occur, i.e., it is disabled.

The process starts when a patient is admitted. Transition *admission* models this activity. This transition has only one input place (start), and this place initially contains a token representing a patient that needs treatment. Therefore, this transition is enabled to occur. When *admission* is fired, it produces two tokens: one for each output place (*p1* and *p2*). The configuration of tokens over places is referred to as *marking*. Each marking, in this case, represents the state of the patient's treatment. After firing *admission*, three transitions are enabled: the token in place *p2* enables transition *check status* - review

of the medical history of the patient; the token in p1 enables both *examine thoroughly* - executed for patients where complications are expected - and *examine casually* - less problematic cases that only need a casual examination. Firing *examine thoroughly* removes the token from *p1*, disabling *examine casually*, and firing *examine casually* disables *examine thoroughly*. In other words, these two activities are mutually exclusive. Firing *check status* does not disable any other transition (it consumes only the token from *p2*), i.e., it occurs concurrently with *examine thoroughly* or *examine casually*.

Before operating, the patient's medical history needs to be checked (token in place p4), and the casual or thorough examination should have been completed (token in p3). Hence, transition *operate* is only enabled if both input places (p3 and p4) contain a token. It consumes two tokens and produces one token for p5. This shows that there are three possible scenarios: the patient may need an *aftercare*, a *consultation*, or the medical staff may need to *examine complications* of the patient. In the latter case, transition *examine complications* consumes a token from p5 and produces a token for each of its output places (*p1* and *p2*). This was the marking directly following the occurrence of *admission*. Several iterations are possible. The process ends with a token in place *end*. Formally, according to Jensen and Kristensen (2009):

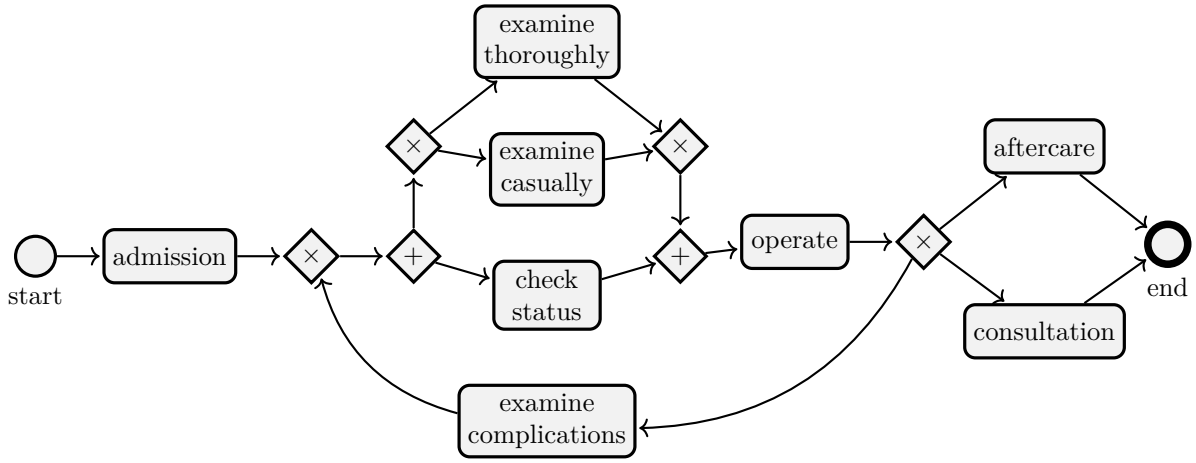
Definition 1: Petri Net

A Petri net is a triplet $N = (P, T, F)$ where P is a finite set of places, T is a finite set of transitions such that $P \cap T = \emptyset$, and $F \in (P \times T) \cup (T \times P)$ is a set of directed arcs, called the *flow relation*.

The Petri net shown in Figure 2.2 can be formalized as follows: $P = \{start, p1, p2, p3, p4, p5, end\}$, $T = \{a, b, c, d, e, f, g, h\}$, and $F = \{(start, a), (a, p1), (a, p2), (p1, b), (p1, c), (p2, d), (b, p3), (c, p3), (d, p4), (p3, e), (p4, e), (e, p5), (p5, f), (f, p1), (f, p2), (p5, g), (p5, h), (g, end), (h, end)\}$. In this case, the transition names were replaced by letters for readability: a = admission, b = examine thoroughly, c = examine casually, d = check status, e = operate, f = examine complications, g = aftercare, and h = consultation.

Figure 2.3 models the same process as a BPMN diagram that uses gateways to model the control-flow logic (MODEL, 2010). The gateways are drawn as diamond shapes in the diagram. The X gateway denotes XOR split/join, and the $+$ gateway denotes AND split/join. In Figure 2.3, the gateway directly following activity *admission* is an XOR-join gateway. It is used to be able to "jump back" after deciding *examine complications*. After this XOR-join gateway, there is an AND-split gateway to model that the *check status* can be done in parallel with the selected examination type (thorough or casual). After *check status*, an AND-join synchronizes the process after examination and checking status. Then, activity *operate* occurs, and an XOR-split separates the process flow into one of the excluding activities: *aftercare*, *consultation*, and *examine complications*.

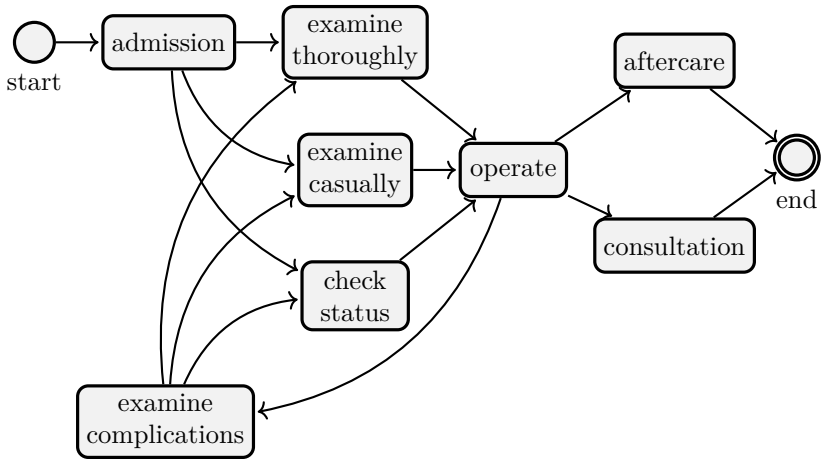
Figure 2.3 – A BPMN diagram modeling a healthcare process



Source: Mans, Aalst and Vanwersch (2015)

Figure 2.4 shows the same process modeled by Figure 2.3 but using a *directly-follows graph*. A Directly-follows Graph (DFG) consists of *nodes* (□) representing activities, the start node (○), and the final node (◎); and *edges*. Each edge denotes that the target activity can occur immediately after the source activity.

Figure 2.4 – A DFG modeling a healthcare process



Source: adapted from Mans, Aalst and Vanwersch (2015)

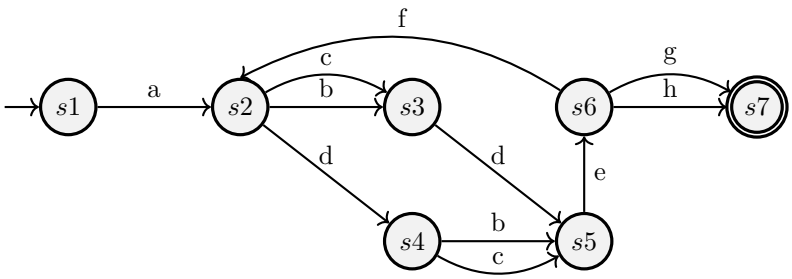
Formally, according to Chartrand (1977):

Definition 2: Direct Follower Graph

A *direct follower graph* is an ordered pair $G = (V, E)$ where V is a set of nodes and E is the set of ordered pairs of nodes, named edges.

Figure 2.5 shows the same process modeled by Figure 2.4 but using a *transition system*. The activity names were replaced by letters for readability[1]. A transition system consists of states (◯) and labeled transitions (arcs). There is one initial state (marked with an arrow) and one final state (◎). Each state has a unique label; each transition connects two states and is labeled with the name of an activity.

Figure 2.5 – A transition system modeling a healthcare process



Source: adapted from Aalst (2016)

Formally, according to Keller (1976):

Definition 3: Transition system

A *transition system* is a tuple (S, A, T) where S is a set of states, A is a set of labels, and T is a relation of labeled transitions (i.e., a subset of $S \times A \times S$). $S_i \subseteq S$ is the set of initial states and $S_f \subseteq S$ is the set of final states.

Considering the example, the set of states is $S = \{s1, s2, s3, s4, s5, s6, s7\}$, the set of activities is $A = \{a, b, c, d, e, f, g, h\}$, the initial state is $S_i = \{s1\}$, and the final state is $S_f = \{s7\}$. The transitions are shown in Figure 2.5.

Figures 2.2, 2.3, 2.4, and 2.5 show the ordering of activities for the process described, i.e., how the activities occur within a certain process in a general way. However, no details about the activities performed by each real-world patient (time, resources, etc.) are described. Most modeling languages offer notations for modeling other perspectives, such as the organizational or resource perspective (ZHAO; ZHAO, 2014) and the time perspective (AALST; SCHONENBERG; SONG, 2011). The next section presents how the event information with such details is stored as event logs.

[1] a = admission, b = examine thoroughly, c = examine casually, d = check status, e = operate, f = examine complications, g = aftercare, and h = consultation.

2.2 Event logs

Figure 2.1 shows that process mining activities start from event data. Event data is generated by the information systems and stored as an event log. Event logs record process instances referred to as *cases*. For example, the process model by Figure 2.3 that describes a certain patient workflow can occur in reality several times, one for each different patient. Each different patient order of activities (case), may differ a little since the process model supports loops and exclusive activities. Event logs describe the details of each different event in each different case. Additional information like time spent, resources consumed, cost information, or any other data attribute can be added. Formally (AALST, 2016):

Definition 4: Event, event attribute

Let A be the set of activities in a process, C the set of cases, and T the time domain. Let E be the set of all events. Let Σ be the set of all names of data attributes an event can have. An event $e \in E$ is a tuple (a, c, t) for $a \in A$, $c \in C$, and $t \in T$. For any event $e \in E$ and name $\sigma \in \Sigma$, $\#_\sigma(e)$ denotes the value of attribute σ for event e.

Each case c (process instance) contains a *trace* (τ) that is a timely ordered sequence of events. Each case can also be characterized by several data *attributes* such as the location from each the event was executed, for instance. Formally (AALST, 2016):

Definition 5: Trace, case attribute, event log

A case $c \in C$ contains a *trace*: a finite sequence of events $\tau \in E^*$ such that each event appears only once. The order of events in a trace should respect their timestamps. Let Γ be the set of all names of data attributes a case can have. For any case $c \in C$ and name $\gamma \in \Gamma$, $\#_\gamma(c)$ denotes the value of attribute γ for case c. An *event log* is a set of cases $L \subseteq C$ such that each event appears at most once in the entire log. $\#_\sigma(L)$ denotes the set of values of attribute σ for all events in log L. $\#_\gamma(L)$ denotes the set of values of attribute γ for all cases in log L.

For example, Frame 2.1 presents a healthcare event log corresponding to the same process model modeled by Figure 2.3. Each line corresponds to a certain event that was registered. Each event corresponds to a specific case, i.e., a patient. Additional information about each event is stored as well: the timestamp - when the event occurred; the resource - who performed such event; and the cost information.

Events (in Frame 2.1) are grouped by *case*. The first event of case *1* has the id *423* and corresponds to the execution of the activity *admission*. It was performed by *Pete*, on *December* 30^{th}*, 2010* at *11:02 AM*, and it costed *50 dollars*. A more compact representation of an event log is represented by its *trace*. A trace records the order in

which each activity occurs within a case. Using the timestamp of each recorded event, it is possible to identify the order of events within a case. Replacing the activity names by letters for readability[2], the last column of Frame 2.1 expresses each case by traces. Next, a standard for event logs is presented.

2.2.1 XES

IEEE Standard for eXtensible Event Stream (XES) (GÜNTHER; VERBEEK, 2009; IEEE, 2016) defines a tag-based language for capturing systems behaviors in event logs. Figure

[2] a = admission, b = examine thoroughly, c = examine casually, d = check status, e = operate, f = examine complications, g = aftercare, and h = consultation.

Frame 2.1 – A fragment of some healthcare event log: each line corresponds to an event

Case id	Event id	Properties					Trace
		Timestamp	Activity	Resource	Cost	...	
	423	30-12-2010:11.02	admission	Pete	50	...	
	424	31-12-2010:10.06	examine thoroughly	Sue	400	...	
1	425	05-01-2011:15.12	check status	Mike	100	...	$\langle a, b, d, e, h \rangle$
	426	06-01-2011:11.18	operate	Sara	200	...	
	427	07-01-2011:14.24	consultation	Pete	200	...	
	483	30-12-2010:11.32	admission	Mike	50	...	
	485	30-12-2010:12.12	check status	Mike	100	...	
2	487	30-12-2010:14.16	examine casually	Pete	400	...	$\langle a, d, c, e, g \rangle$
	488	05-01-2011:11.22	operate	Sara	200	...	
	489	08-01-2011:12.05	aftercare	Ellen	200	...	
	521	30-12-2010:14.32	admission	Pete	50	...	
	522	30-12-2010:15.06	examine casually	Mike	400	...	
	524	30-12-2010:16.34	check status	Ellen	100	...	
	525	06-01-2011:09.18	operate	Sara	200	...	
3	526	06-01-2011:12.18	examine complications	Sara	200	...	$\langle a, c, d, e, f, b, d, e, g \rangle$
	527	06-01-2011:13.06	examine thoroughly	Sean	400	...	
	530	08-01-2011:11.43	check status	Pete	100	...	
	531	09-01-2011:09.55	operate	Sara	200	...	
	533	15-01-2011:10.45	aftercare	Ellen	200	...	
	641	06-01-2011:15.02	admission	Pete	50	...	
	643	07-01-2011:12.06	check status	Mike	100	...	
4	644	08-01-2011:14.43	examine thoroughly	Sean	400	...	$\langle a, d, b, e, h \rangle$
	645	09-01-2011:12.02	operate	Sara	200	...	
	647	12-01-2011:15.44	consultation	Ellen	200	...	
	711	06-01-2011:09.02	admission	Ellen	50	...	
	712	07-01-2011:10.16	examine casually	Mike	400	...	
	714	08-01-2011:11.22	check status	Pete	100	...	
	715	10-01-2011:13.28	operate	Sara	200	...	
	716	11-01-2011:16.18	examine complications	Sara	200	...	
	718	14-01-2011:14.33	check status	Ellen	100	...	
5	719	16-01-2011:15.50	examine casually	Mike	400	...	$\langle a, c, d, e, f, d, c, e, f, c, d, e, h \rangle$
	720	19-01-2011:11.18	operate	Sara	200	...	
	721	20-01-2011:12.48	examine complications	Sara	200	...	
	722	21-01-2011:09.06	examine casually	Sue	400	...	
	724	21-01-2011:11.34	check status	Pete	100	...	
	725	23-01-2011:13.12	operate	Sara	200	...	
	726	24-01-2011:14.56	consultation	Mike	200	...	
	871	06-01-2011:15.02	admission	Mike	50	...	
	873	06-01-2011:16.06	examine casually	Ellen	400	...	
6	874	07-01-2011:16.22	check status	Mike	100	...	$\langle a, c, d, e, g \rangle$
	875	07-01-2011:16.52	operate	Sara	200	...	
	877	16-01-2011:11.47	aftercare	Mike	200	...	

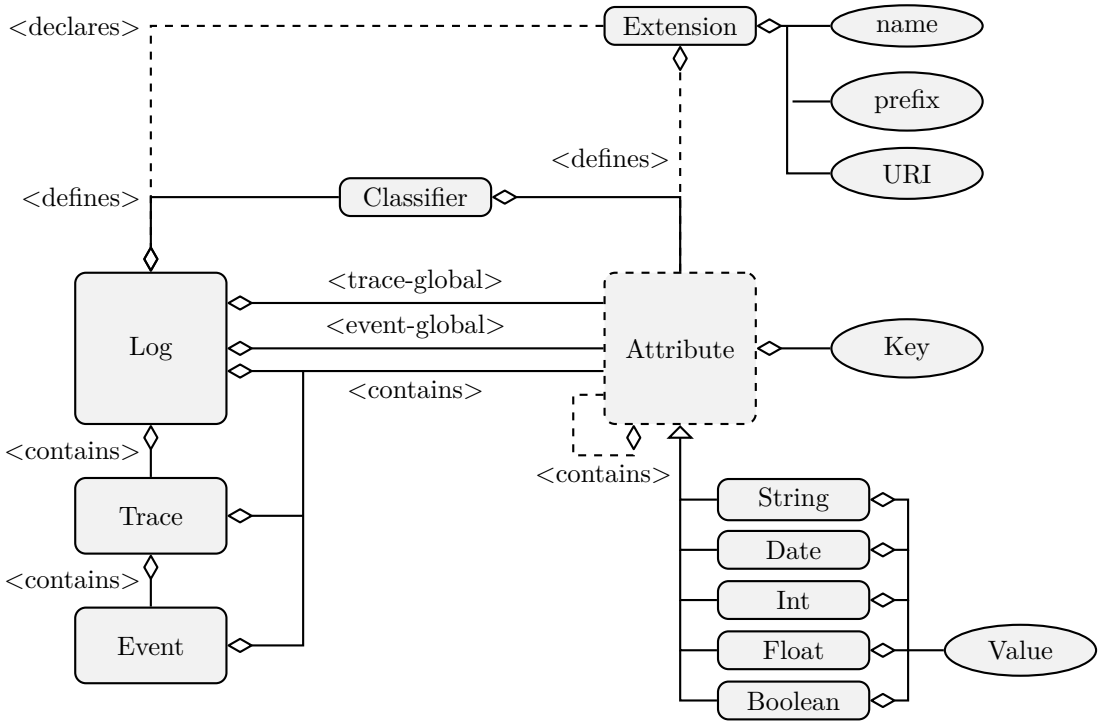
Source: adapted from Mans, Aalst and Vanwersch (2015) and Aalst (2016)

2.6 defines a meta-model of XES. A *log* contains *traces*, and each trace contains *events*. Logs, traces, and events have *attributes*. There are five core types: *String*, *Date*, *Int*, *Float*, and *Boolean* for attributes, each with its *value*. *Extensions* may define new attributes, and a log should declare the extensions used in it. For example, the "Time" extension defines a timestamp attribute of type *Date*. Extensions have a *name*, a *prefix*, and a *Uniform Resource Identifier* (URI).

Global attributes are attributes that are declared to be mandatory. Such attributes can be done at the trace (*trace-global*) or event (*event-global*) level. Attributes may be nested. Event classifiers are defined for the log and assign a "label" (e.g., activity name) to each event. There may be multiple classifiers.

Using the event log and their corresponding traces makes it possible to perform the process mining activities. *Conformance* and *enhancement* activities are executed using both process models and event logs as input. *Discovery* activity, described in the next section, takes an event log as input and calculates its corresponding process model (AALST et al., 2011a).

Figure 2.6 – Meta model of XES



Source: Günther and Verbeek (2009)

2.3 Discovery

The discovery techniques of process mining take as input the event logs, normally represented as traces, and produces as output a process model, as illustrated by Figure 2.7. Several techniques and algorithms have been proposed in the literature to address this problem: α-algorithm (AALST; WEIJTERS; MARUSTER, 2004), heuristic miner (WEIJTERS; RIBEIRO, 2011), genetic miner (MEDEIROS; WEIJTERS; AALST, 2007), language-based region miners (BERGENTHUM et al., 2007), inductive miner (LEEMANS; FAHLAND; AALST, 2013), etc. Formally (AALST, 2016):

Definition 6: General process discovery problem

Let L be an event log as defined in Definition 5. A process discovery algorithm is a function that maps L onto a process model such that the model is "representative" of the behavior seen in the event log.

Figure 2.7 – Discovery activity

Source: adapted from Aalst (2016)

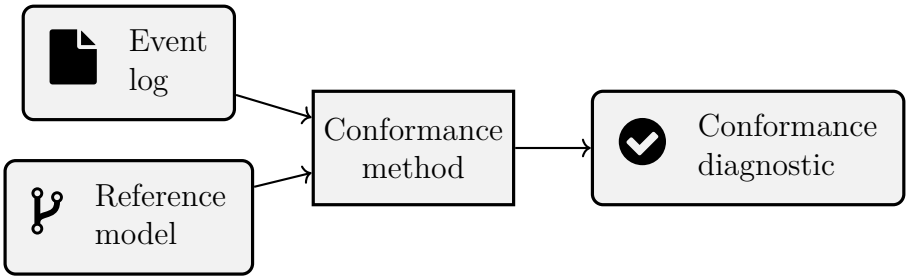
Each discovery technique has its characteristics (AALST, 2016). For instance, the α-algorithm does not support well loops of size 2, and it does not take into account the frequency of the traces and activities, but it is simpler. Inductive miner (LEEMANS; FAHLAND; AALST, 2013) can handle noise traces and takes a *noise threshold* as input. The noise threshold specifies the percentage of allowed infrequent behavior in the log to create the process model. As default, the noise threshold is defined as 0.2 based on the Pareto principle (KIREMIRE, 2011). Another example is the heuristic miner (WEIJTERS; RIBEIRO, 2011) that uses the directly-precedence relation, providing a way to handle noise and to find common constructs in the process model. To handle this, two threshold parameters are used. The *dependency threshold* defines the percentage of occurrences to a relation to be considered as an implication (default is 0.5). The *AND threshold* specifies a percentage of occurrences of relations so that two activities are not considered disjointed (default is 0.65).

For further details on discovery techniques, Dongen, Medeiros and Wen (2009), Garcia et al. (2019) present a systematic literature review about process discovery. Next, the second process mining activity, conformance, is presented.

2.4 Conformance

The conformance techniques of process mining take as input an event log and a process model, called *reference model*, and generate a diagnostic report about conformity, as illustrated by Figure 2.8. The main goal is identifying and quantifying deviations and discrepancies between the modeled and the observed behavior. Moreover, conformance-checking techniques can be used for measuring the performance of process discovery algorithms and to repair models that are not aligned well with reality (AALST, 2016; LEEMANS et al., 2021). There are two main types of conformance checking: token replay and alignments. The details are explored next. For additional information on conformance checking, Dunzer et al. (2019), Naderifar, Sahran and Shukur (2019) presents a literature review. This section may be skipped if the reader is familiar with conformance techniques.

Figure 2.8 – Conformance activity



Source: adapted from Aalst (2016)

2.4.1 Token Replay

The first conformance technique replays the traces in a log over the reference model, keeping track of what happens to tokens on the workflow. Token replay measures the fitness of the trace according to the model, i.e., the proportion of the behavior in the trace (or the log) possible according to the model. According to the equation (AALST, 2014):

Definition 7: Fitness of a trace by token replay

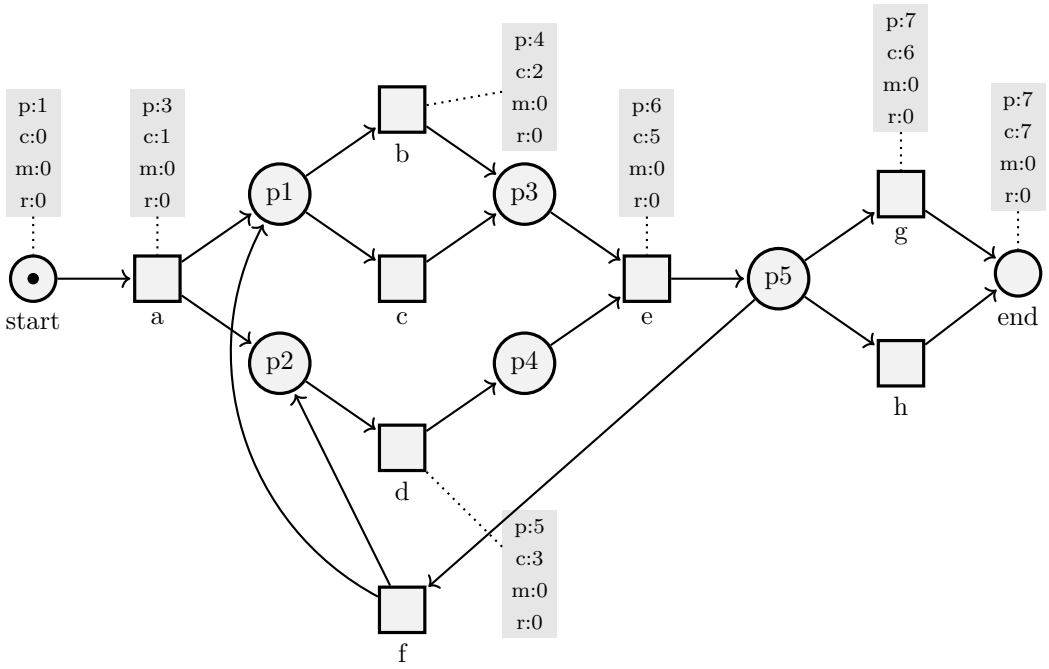
The fitness of a trace σ over a Petri net N is:

$$fitness(\sigma, N) = \frac{1}{2}\left(1 - \frac{m}{c}\right) + \frac{1}{2}\left(1 - \frac{r}{p}\right)$$

where p, c, m, and r are, respectively, the numbers of produced, consumed, missing, and remaining tokens.

To illustrate how the token replay works, Figure 2.9 shows an example of how the four counters (p, c, m, and r) are updated.

Figure 2.9 – Token replay example for trace $\langle a, b, d, e, h \rangle$



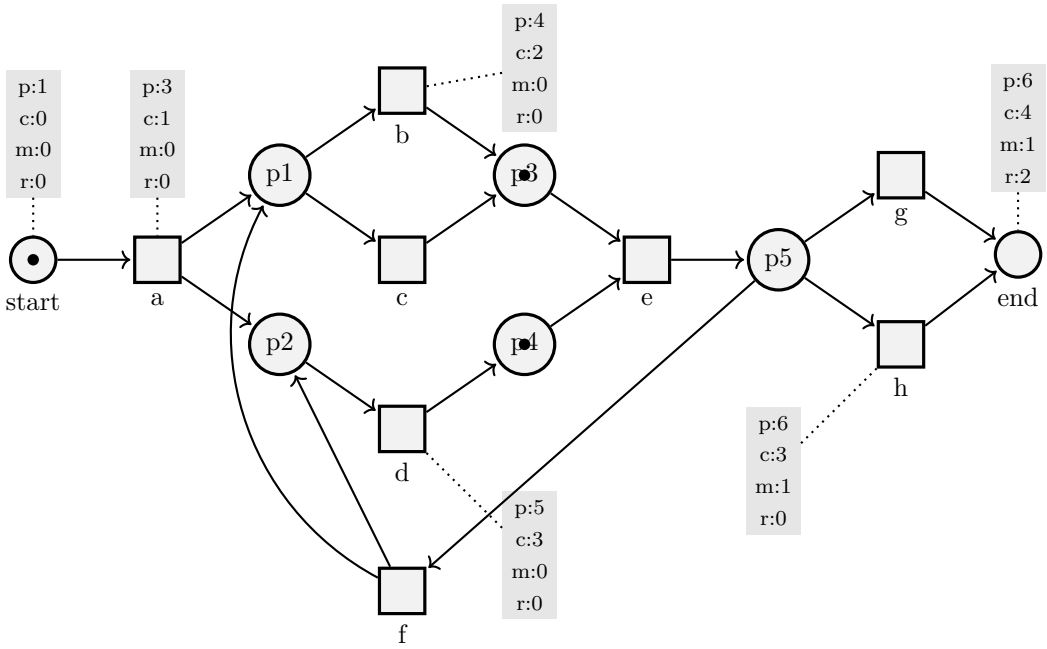
Source: adapted from Aalst (2016)

The example considers the trace $\langle a, b, d, e, h \rangle$. At the start place, p counter is set as 1, and all other counters are set as 0 p:1 c:0 m:0 r:0. After firing transition a, the token at the start place is consumed ($c = 1$) and 2 tokens are produced for places $p1$ and $p2$ p:3 c:1 m:0 r:0. Transition b consumes the token at $p1$ and produces a token for $p3$ p:4 c:2 m:0 r:0. Transition d then is executed consuming the token at $p2$ and producing a token for $p4$ p:5 c:3 m:0 r:0. Transition e consumes both tokens at $p3$ and $p4$ and produces a token for $p5$ p:6 c:5 m:0 r:0. After firing h, the token at $p5$ is consumed and 1 token is produced for end place p:7 c:6 m:0 r:0. Lastly, the token is consumed from end place p:7 c:7 m:0 r:0.

The fitness of trace $\langle a, b, d, e, h \rangle$ can be calculated from Equation 7 where $p = 7$, $c = 7$, $m = 0$, $r = 0$: $\frac{1}{2}\left(1 - \frac{0}{7}\right) + \frac{1}{2}\left(1 - \frac{0}{7}\right) = 1$. Because there are no missing or remaining tokens, trace $\langle a, b, d, e, h \rangle$ fits perfectly to the reference model.

Considering the trace $\langle a, b, d, h \rangle$, Figure 2.10 shows another example of the fitness difference. Transitions a, b, and d consume and produce tokens in the same way the previous example did p:5 c:3 m:0 r:0, and after firing them, the places $p3$ and $p4$ contain 1 token each. According to the trace, the next transition that would be fired is h, but $p5$ has no token. Therefore, transition h produces 1 token for the end place, and the missing counter is incremented p:6 c:3 m:1 r:0. Lastly, the token is consumed from the end place, and the remaining tokens in $p3$ and $p4$ are counted p:6 c:4 m:1 r:2.

Figure 2.10 – Token replay example for trace $\langle a, b, d, h \rangle$



Source: adapted from Aalst (2016)

The fitness of trace $\langle a, b, d, h \rangle$ calculated from Equation 8 where $p = 6$, $c = 4$, $m = 1$, $r = 2$ is $\frac{1}{2}\left(1 - \frac{1}{4}\right) + \frac{1}{2}\left(1 - \frac{2}{6}\right) = 0.708$. For the whole log:

Definition 8: Fitness of a log by token replay

The fitness of a whole log L over Petri net N is calculated by:

$$fitness(L, N) = \frac{1}{2}\left(1 - \frac{\sum_{i=1}^{n} m_i}{\sum_{i=1}^{n} c_i}\right) + \frac{1}{2}\left(1 - \frac{\sum_{i=1}^{n} r_i}{\sum_{i=1}^{n} p_i}\right)$$

where p_i, c_i, m_i, and r_i are, respectively, the numbers of produced, consumed, missing, and remaining tokens for each of the n traces in L.

Considering the above examples, $L = \{\langle a, b, d, e, h \rangle, \langle a, b, d, h \rangle\}$, the fitness of log L is $\frac{1}{2}\left(1 - \frac{0+1}{7+4}\right) + \frac{1}{2}\left(1 - \frac{0+2}{7+6}\right) = 0.878$. The fitness indicates that some traces did not fit correctly the reference model, i.e., some problem occurred. Furthermore, it is possible to observe *where* the problem occurred by checking in Figure 2.10 where the remaining tokens are ($p3$ and $p4$), what places had missing tokens ($p5$) and the transitions (e) involved with these places.

Using token-based replay, it is possible to split cases into fitting and non-fitting traces. However, the approach is Petri-net specific and can only be applied to other process models after conversion. Alignments are another way of checking conformance

that overcomes such problems (AALST; ADRIANSYAH; DONGEN, 2012).

2.4.2 Alignments

An alignment is a correspondence between a trace and a process model. For instance, the alignment λ between the trace $\sigma = \langle a, b, d, h \rangle$ and the model in Figure 2.2 can be expressed as:

$$\lambda = \begin{array}{|c|c|c|c|c|} \hline a & d & b & \gg & h \\ \hline a & d & b & e & h \\ \hline \end{array}$$

The top row corresponds to the trace σ, and the bottom row corresponds to a path from the initial marking to the final marking of the process model. Any process model can be used. The "$\gg$" symbol denotes misalignment. It means that the trace σ "lacked" an activity that the model has. Several alignments can be created for the same trace and model:

$$\lambda_1 = \begin{array}{|c|c|c|c|c|} \hline a & d & b & \gg & h \\ \hline a & d & b & e & h \\ \hline \end{array} \qquad \lambda_2 = \begin{array}{|c|c|c|c|c|c|} \hline a & d & b & \gg & \gg & h \\ \hline a & d & b & e & h & \gg \\ \hline \end{array}$$

$$\lambda_3 = \begin{array}{|c|c|c|c|c|c|} \hline a & d & b & \gg & \gg & h \\ \hline a & d & \gg & b & e & h \\ \hline \end{array} \qquad \lambda_4 = \begin{array}{|c|c|c|c|c|c|c|c|c|} \hline a & d & b & h & \gg & \gg & \gg & \gg & \gg \\ \hline \gg & \gg & \gg & \gg & a & d & b & e & h \\ \hline \end{array}$$

A so-called *optimal alignment* is the best match given a trace and a model. Given a trace σ and a model N, there is precisely one optimal alignment denoted $\lambda_{opt}(\sigma)$. A *worst-case alignment* happens too, when the model and trace are completely misaligned $\lambda_{worst}(\sigma)$. In the above example, with only one misalignment λ_1 is the optimal alignment, i.e., $|\lambda_{opt}(\sigma)| = 1$ and λ_4 is the worst-case alignment, i.e., $|\lambda_{worst}(\sigma)| = 9$. $|\lambda|$ denotes the number of $\gg$ in a alignment λ. The fitness is calculated as follows:

Definition 9: Fitness of a trace and a log by Alignments

The fitness of a trace σ, and a log L over a Petri net N is

$$fitness(\sigma, N) = 1 - \frac{|\lambda_{opt}(\sigma)|}{|\lambda_{worst}(\sigma)|}$$

$$fitness(L, N) = 1 - \frac{\sum_{\sigma \in L} |\lambda_{opt}(\sigma)|}{\sum_{\sigma \in L} |\lambda_{worst}(\sigma)|}$$

Considering now the same process model, but another log as an example $L = \{\sigma_1 = \langle a, b, d, e, h \rangle, \sigma_2 = \langle a, b, d, h \rangle\}$, the trace σ_1 fits perfectly to the model:

$$fitness(\sigma_1, N) = 1 - \frac{|\lambda_{opt}(\sigma_1)|}{|\lambda_{worst}(\sigma_1)|} = 1 - \frac{0}{10} = 1$$

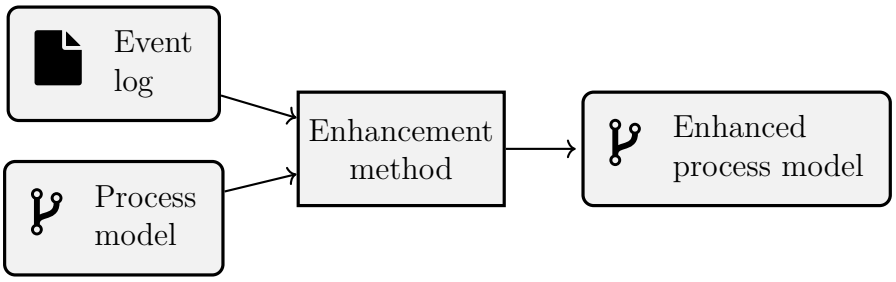
where $|\lambda_{opt}(\sigma_1)|= 0$ (no misalignment) and $|\lambda_{worst}(\sigma_1)|= 10$ (total misaligned between the model and σ_1). The trace σ_2 have the fitness:

$$fitness(\sigma_2, N) = 1 - \frac{|\lambda_{opt}(\sigma_2)|}{|\lambda_{worst}(\sigma_2)|} = 1 - \frac{1}{9} = 0.888$$

where $|\lambda_{opt}(\sigma_2)|= 0$ (one misalignment, because it misses e) and $|\lambda_{worst}(\sigma_2)|= 9$ (total misaligned between the model and σ_2). Considering the whole log L, the fitness is $fitness(L, N) = 1 - \frac{0+1}{9+10} = 0.947$. Conformance checking can be used to identify how deviating is the actual process (event log) from the process model. By identifying the non-conformity, it is possible to enhance the process model, as described in the next section.

2.5 Enhancement

Figure 2.11 – Enhancement activity



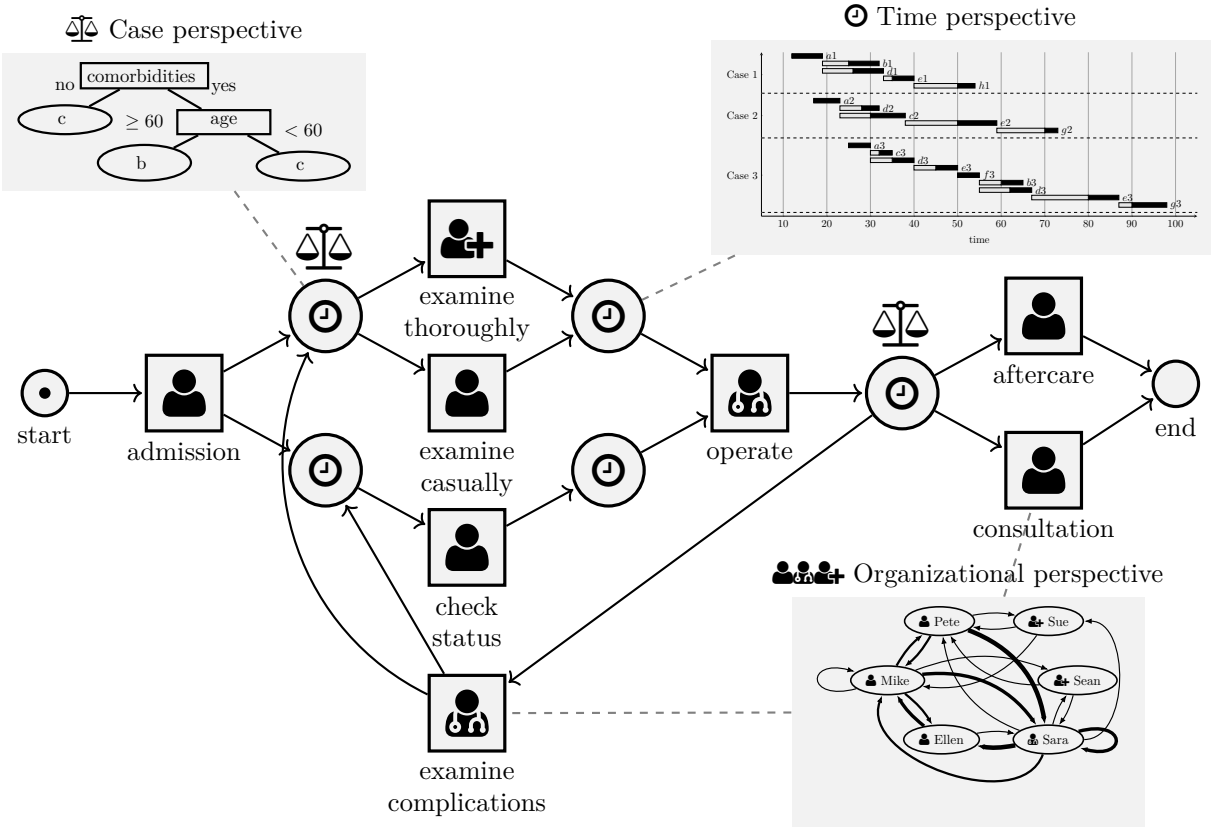
Source: adapted from Aalst (2016)

Enhancement techniques aim to improve, repair, or extend existing process models using recorded information in event logs (AALST, 2016). They take as input the event log and the original model and generate the enhanced model as depicted in Figure 2.11.

Repairing process models can be conducted when conformance checking identifies differences between the process model and its correspondent event log. A repaired model can be obtained by performing model editions. For instance, paths that are never taken can be removed from the model. Fahland and Aalst (2015) provide a method for obtaining the repaired model.

By analyzing the event logs, perspectives can be added to the model so an enhanced model version may be obtained. Figure 2.12 shows the same process model from Figure 2.2, but now with extra icons representing these perspectives that will be explained in the next sections. The *organizational perspective* may be added, represented in the figure by *people icons*. It is possible to analyze the social network between people involved in the process. Next, one can identify organizational entities that connect activities to resources. *Time perspective* can be added, represented by the *clock icons*. Timestamps

Figure 2.12 – An enhanced process model with perspectives



Source: adapted from Aalst (2016)

and frequencies can be used to learn probability distributions that describe service and waiting times and routing probabilities. Also, conformance checking can be modified to add the time perspective to process models. *Case perspective* can be explored for decision mining (represented by *scale icons*), i.e., which data is relevant and should be included in the model. Furthermore, other perspectives may be explored. For example, information on risks and costs can be added to the model. The cost perspective is explored in Chapter 3. Organizational, time, and case perspectives are detailed next. For further details about enhancement, Yasmin, Bukhsh and Silva (2018) presents a literature review about enhancement in Process Mining.

2.5.1 Organization Mining

Organizational mining adds the organizational perspective in the process model (AALST; REIJERS; SONG, 2005; SONG; AALST, 2008). This perspective uses log information to learn about process entities such as people, machines, roles, departments, work patterns, and work distribution. To illustrate this analysis, consider the event log presented in Frame 2.1. One person performs each activity in this example with an assigned role in the process.

Pete, Mike, and Ellen have the role *doctor* (), Sue and Sean have the role *expert* (), and Sara has the role *surgeon* ().

Frame 2.2 shows the resource-activity matrix that relates each person (or resource) to the execution of each activity. Each entry corresponds to the number of times the resource performed each activity. For clarity purposes, the zeros are omitted. For example, consider activity a exclusively executed by Pete, Mike, or Ellen. Pete executed a 50% of the time (3 of 6 cases), Mike executed a twice, and Ellen executed it once. Activities e and f are always executed by Sara. Since the log has 6 cases, activity e had to be redone in some cases because it was executed 9 times.

Frame 2.2 – Example of resource-activity matrix

	a	b	c	d	e	f	g	h
Pete	3		1	3				1
Mike	2		3	4			1	1
Ellen	1		1	2			2	1
Sue		1	1					
Sean		2						
Sara					9	3		

Source: adapted from Aalst (2016)

Furthermore, it is possible to quantify the similarity between resources by comparing the distances between the vectors (lines) on Frame 2.2. Distance measures such as the Hamming distance (NOROUZI; FLEET; SALAKHUTDINOV, 2012), and Pearson's correlation coefficient (SEDGWICK, 2012), or clustering techniques such as DBscan clustering (ESTER et al., 1996) can be used for that. Considering any of these groupings approaches, it is possible to create three groups: doctor group, which includes Pete, Mike, and Ellen and is more related to performing activities a, c, d, g and h; expert group, that includes Sue and Sean and is more related to performing activities b; and Surgeon group, that includes only Sara and is more related to performing activities e and f;

Another kind of analysis that can be conducted involves the handover of work shown on matrices in Figure 2.13. For instance, based on the event log, one can count the frequency the work is handed over from one resource to another. Matrix a shows the handover from one person to another. Matrix b shows the handover considering the roles.

It is possible to visually represent the handover of work matrices by using *social networks* like the one in Figure 2.14 (AALST, 2016). Social networks are directed graphs in which nodes represent the entities (in this case, people and roles), and arches have weights corresponding to the relation between these entities. In this case, the frequency in the handover of work matrices corresponds to the weights in the arches.

Figure 2.13 – Example of handover of work matrices

	Pete	Mike	Ellen	Sue	Sean	Sara
Pete		2		1		4
Mike	2	1	2		1	3
Ellen		3				1
Sue	1	1				
Sean	1					1
Sara	1	2	4	1	1	3

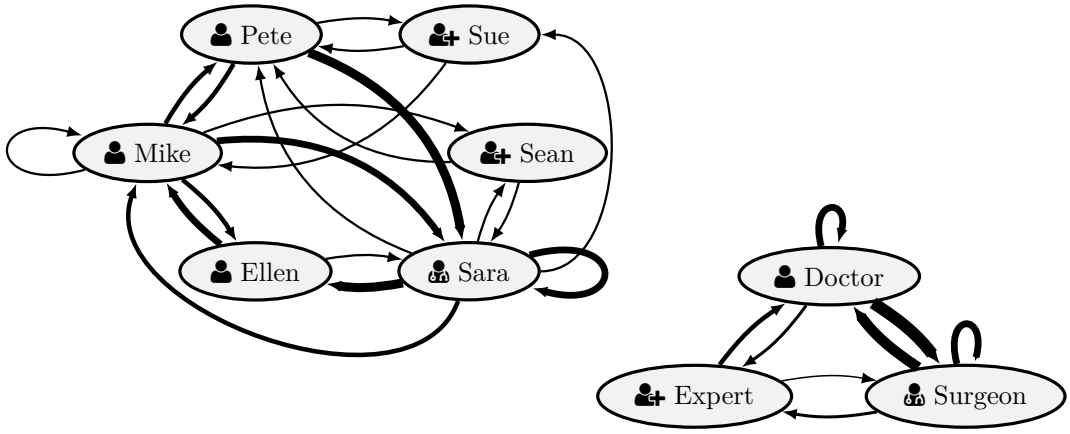
(a) Matrix at individual level

	Doctor	Expert	Surgeon
Doctor	10	2	8
Expert	3		1
Surgeon	7	2	3

(b) Matrix at role level

Source: adapted from Aalst (2016)

Figure 2.14 – Example of social network based on handover of work



(a) Social network at individual level (b) Social network at role level

Source: adapted from Aalst (2016)

It is possible to observe that Mike and Pete frequently hand over work to Sara: 7 times. Also, Sara mostly hands over work to Ellen. At the role level, most of the handovers occur between doctors and surgeons.

2.5.2 Time and Probabilities

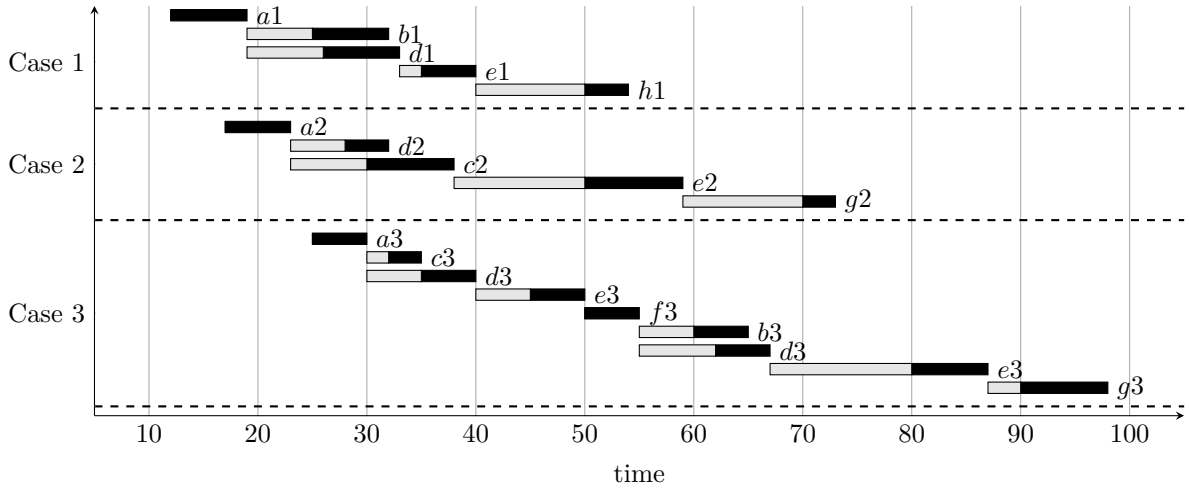
The *time perspective* relates to the timing and frequency of events. Commonly, event logs have events with timestamps. The granularity of timestamps may vary from milliseconds to day or month precision. According to Aalst (2016), the presence of timestamps enables the discovery of bottlenecks, the analysis of service levels, the monitoring of resource utilization, and the prediction of remaining processing times of running cases. Considering the first three cases in the event log presented in Frame 2.1 it is possible to create timelines

of activities.

Figures 2.15 and 2.16 present the timelines showing the activities instances per case and resource, respectively. For simplification reasons, the timestamps were represented by an integer unit of time. Each activity is labeled according to the case in which it was performed. For instance, $a3$ means activity a executed in case 3. The black drawings for each activity in the chart represent the time the activity spent to be executed from beginning to end, called *service time*. The gray drawings for each activity show the time the corresponding activity had to wait to start after being enabled, called *waiting time*.

For example, $a1$ was executed from time 12 to 19. When activity $a1$ finished (time 19), according to the process model, activities $b1$ and $d1$ could be executed, but the log recorded the start of execution of $b1$ at 25, and $d1$ at 26. That means activity $b1$ waited from 19 to 25, and $d1$ waited from 19 to 26.

Figure 2.15 – Example of timeline showing the activity instances per cases



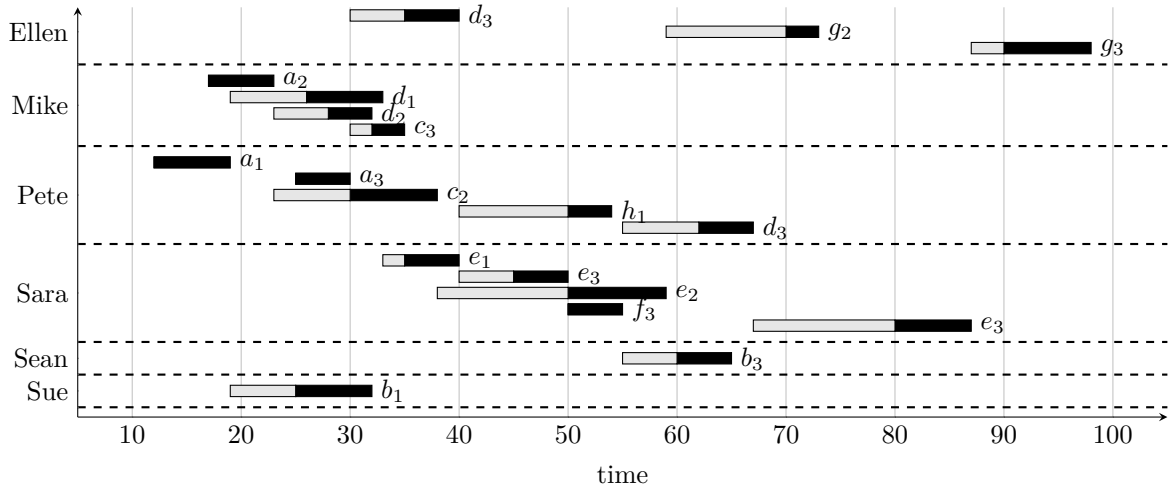
Source: adapted from Aalst (2016)

Considering real-world event logs, it is possible to keep track of service and waiting time to create statistics. One can fit a distribution or compute each activity's mean, standard deviation, minimum, and maximum. According to Aalst (2016), it is possible to compute confidence intervals to derive statements such as "the 85% confidence interval for the mean service time for activity a is between 4 and 5 minutes".

Figures 2.15 and 2.16 show that time analysis can be used to provide various kinds of performance-related information:

- Average waiting time for an activity can be attached to the process model;
- Activities with a high/low variation in service time could be highlighted in the model;

Figure 2.16 – Example of timeline showing the activity instances per resource



Source: adapted from Aalst (2016)

- Bottleneck detection and analysis;
- Cases that spend a long time in a particular activity can be further investigated;
- Overall flow time can be computed;
- Analysis of frequencies and time can be used to show routing probabilities in the model;
- Analysis of frequencies and time can be used to analyze performance by resource (Figure 2.16).

2.5.3 Decision Mining

The *case perspective* focuses on the influence of case, and event attributes on the routing of cases (AALST, 2016). For instance, in Figure 2.12, there are two decision points: after admission (activity a), either a thorough examination (b) or a casual examination (c) follows; and after operating (e), activity g (aftercare), activity h (consultation), or activity f (examine complications) follows.

Decision mining have the goal to find rules illustrating these choices in terms of properties of the case (ROZINAT; AALST, 2006). Classification techniques can be used to find these rules. For illustration purposes, consider the fragment of some healthcare event log in Frame 2.3.

Three case attributes are used for illustration: comorbidities (that can be *yes* in case the patient has comorbidity, or *no*), age, and gender. For this example, consider the

Frame 2.3 – A fragment of some healthcare event log

...	activity	...	comorbidities	gender	age	...
...	b	...	yes	male	65	...
...	c	...	no	male	65	...
...	b	...	yes	female	70	...
...	c	...	yes	male	50	...
...	c	...	no	female	30	...
...	c	...	no	male	70	...
...	c	...	no	female	20	...

decision point in Figure 2.12: after admission (activity a), either a thorough examination (b) or a casual examination (c).

As an example of a classification technique, a decision tree is used (QUINLAN, 1987). The input for decision tree learning is a table where every row lists one categorical response variable (e.g., the chosen activity) and predictor variables (e.g., the patient's attribute). The decision tree aims to explain the response variable in terms of the predictor variables. For this, the technique progressively divides sets of events into subsets such that the variation within each subset is smaller. The entropy measures this variation in each set with k instances by the equation:

$$E = -\sum_{i=1}^{k} p_i log_2(p_i) \tag{2.1}$$

such that p_i is the fraction of elements having the value i, in the set of events. The decision tree algorithm starts with the set of all events represented by the tree's root node. Interactively, it traverses across nodes and checks for entropy decrease. For each node and each attribute, it calculates the effects of dividing that node in terms of entropy. For each division in the tree, entropy is decreased. Considering the Frame 2.3, one can calculate the entropy for the whole set (E_W) and each of its variables' values: with comorbidities ($E_{comorb=yes}$), without comorbidities ($E_{comorb=no}$), female ($E_{gender=female}$), male ($E_{gender=male}$), age smaller than 60 ($E_{age<60}$) and age greater than or equal to 60 ($E_{age\geq 60}$).

The fraction p_i in each entropy calculation corresponds to the fraction of elements having the value i. For instance, $\frac{2}{7}$ of Frame 2.3 have the value b, and $\frac{5}{7}$ have the value c. Therefore, E_W is $-\frac{2}{7}log_2\left(\frac{2}{7}\right) - \frac{5}{7}log_2\left(\frac{5}{7}\right) = 0.86$. For $E_{gender=male}$, consider only the rows where the *gender* is *male*. In these rows, $\frac{1}{4}$ has the value b, and $\frac{3}{4}$ have the value c. The details of each entropy calculation are shown in Equation 2.2.

$$
\begin{aligned}
E_W &= -\frac{2}{7}log_2\left(\frac{2}{7}\right) - \frac{5}{7}log_2\left(\frac{5}{7}\right) = 0.86 \\
E_{comorb=yes} &= -\frac{2}{3}log_2\left(\frac{2}{3}\right) - \frac{1}{3}log_2\left(\frac{1}{3}\right) = 0.92 \\
E_{comorb=no} &= -\frac{4}{4}log_2\left(\frac{4}{4}\right) = 0 \\
E_{gender=male} &= -\frac{1}{4}log_2\left(\frac{1}{4}\right) - \frac{3}{4}log_2\left(\frac{3}{4}\right) = 0.81 \\
E_{gender=female} &= -\frac{1}{3}log_2\left(\frac{1}{3}\right) - \frac{2}{3}log_2\left(\frac{2}{3}\right) = 0.92 \\
E_{age<60} &= -\frac{3}{3}log_2\left(\frac{3}{3}\right) = 0 \\
E_{age\geq 60} &= -\frac{2}{4}log_2\left(\frac{2}{4}\right) - \frac{2}{4}log_2\left(\frac{2}{4}\right) = 1
\end{aligned}
\tag{2.2}
$$

The total entropy for each variable is calculated by a weighted sum of the n entropies of such variable as:

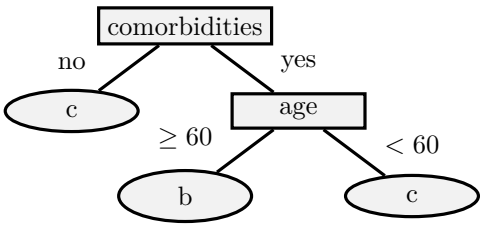
$$
E_{variable} = -\sum_{i=1}^{n} w_i E_{variable=i} \tag{2.3}
$$

where w_i corresponds to the fraction of rows in which the *variable* have the value i. For instance, variable *comorbidities* have the entropy calculated for *yes* ($E_{comorb=yes}$) and for *no* ($E_{comorb=no}$). *yes* corresponds to $\frac{3}{7}$ of the frame, while *no* corresponds to $\frac{4}{7}$. Therefore, the information gain for variable *comorbidities* is $E_{comorb} = \frac{3}{7}E_{comorb=yes} + \frac{4}{7}E_{comorb=no}$. The details of each variable entropy calculation are shown in Equation 2.4.

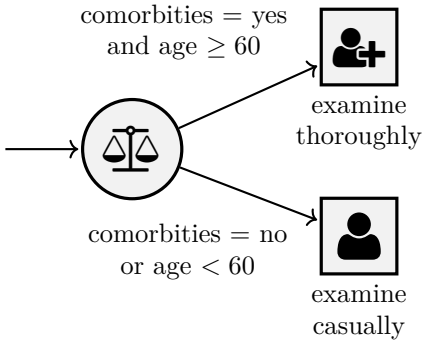
$$
\begin{aligned}
E_{comorb} &= \frac{3}{7}E_{comorb=yes} + \frac{4}{7}E_{comorb=no} = \frac{3}{7}0.92 + \frac{4}{7}0 = 0.39 \\
E_{gender} &= \frac{4}{7}E_{gender=male} + \frac{3}{7}E_{gender=female} = \frac{4}{7}0.81 + \frac{3}{7}0.92 = 0.86 \\
E_{age} &= \frac{4}{7}E_{age\geq 60} + \frac{3}{7}E_{age<60} = \frac{4}{7}1 + \frac{3}{7}0 = 0.57
\end{aligned}
\tag{2.4}
$$

In this case, the smaller entropy is on variable *comorbidities*. This variable is the first node in the decision tree. Interactively repeating the process of dividing sets and calculating entropy, it is possible to create the decision tree shown in Figure 2.17a. If the patient does not have *comorbidities*, then a casual examination is performed (activity c). If the patient has comorbidities, then the variable *age* is checked. A casual examination is performed if the patient is younger than 60 (activity c). Otherwise, activity b is executed (examine thoroughly). The rules discovered by the decision tree can be used to enhance the process model. Figure 2.17b shows a fragment of the Petri net presented for the example, now with the decision rules learned annotated.

Figure 2.17 – Example of decision mining



(a) Decision tree for the example



(b) Fragment of the annotated Petri net with the decision rules learned

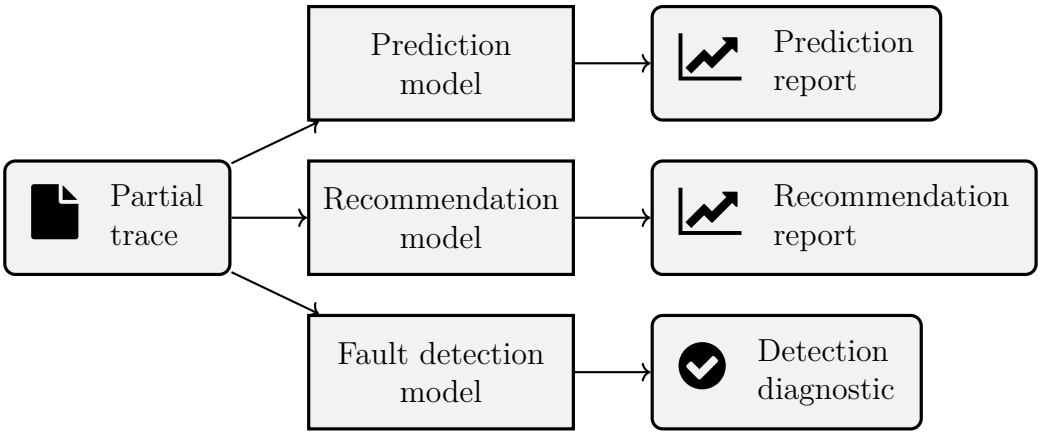
Source: adapted from Aalst (2016)

The above-described procedure can be repeated for all decision points in a process model. The results can be used to extend the process model, thus incorporating the *case* perspective. In addition to the *organizational* and *time* perspectives, the enhanced model illustrated in Figure 2.12 can be used for various purposes: it provides new insights about the process; it may generate new ideas for process improvement; it can be used as input for configuring BPM systems; it can be used to generate a simulation model (ROZINAT et al., 2009b); etc. Next, ongoing cases are analyzed for operational support.

2.6 Operational Support

According to Aalst (2016), most process-mining techniques analyze events that have already been completed and recorded in event logs. However, many data sources in today's systems are updated in real-time and may be available to analyze events when they occur. Therefore, process mining can also be used for online *operational support*. For example, deviations can be *detected* at run-time; the remaining flow time for a running case can be *predicted*; and suitable actions can be *recommended* to minimize costs (AALST; PESIC; SONG, 2010). Figure 2.18 illustrates how operational support is used. Based on the event log from concluded traces, models are created for prediction, recommendation, and deviation detection. When an ongoing case runs, the partial trace is used along with the created models to recommend, predict and detect faults. The following subsections explore these three operational support activities.

Figure 2.18 – Operational Support



Source: adapted from Aalst (2016)

2.6.1 Detect

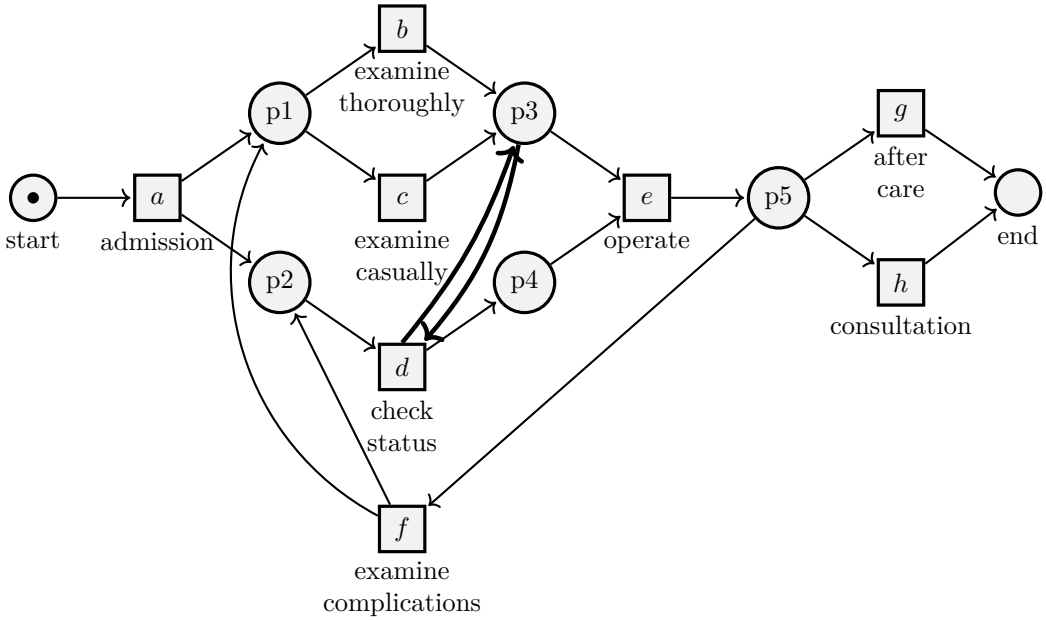
Detect activity is similar to conformance checking explored in Section 2.4. The difference consists of conformance checking being done "off-line", using a completed and recorded case as an event log, while detect activity is executed over ongoing cases checking partial traces (MYERS et al., 2018; AALST, 2016). The idea is to keep track of every activity in the partial trace, comparing it to a normative model. When an activity in the partial trace does not conform with the normative model, the operational support system detects and reports a violation to the user. This enables the user's intervention over the case while it is still happening.

Consider that one wants to add a constraint to the model of the previously presented example of a healthcare process. In this example, the constraint is that activity d (check status) can only occur after b or c (examine thoroughly/casually) has been completed. Figure 2.19 shows the previously presented model with two highlighted transitions implementing such constraint.

Traces $\langle a, d, c, e, g\rangle$, $\langle a, c, d, e, f, d, c, e, f, c, d, e, h\rangle$, and $\langle a, b, d, e, h\rangle$ are the examples of traces that are going to be checked partially to illustrate the detect activity. After each event, it is checked whether there is a deviation. In trace $\langle a, d, c, e, g\rangle$, the first event a is executed, and no deviation is found because it can be replayed in Figure 2.19 without missing tokens. The next event d is executed, and a deviation is detected: the execution of d consumes tokens for places p_2 and p_3, but because d cannot execute before b, p_3 has a missing token. Since a deviation occurred, an alert is generated and sent to the user.

For trace $\langle a, c, d, e, f, d, c, e, f, c, d, e, h\rangle$, events a, c, d, e and f occur without deviations. When d occur after these events, a deviation happens, an another alert is generated: the execution of d consumes tokens for places p_2 and p_3, but because d cannot execute before c, p_3 has a missing token. Trace $\langle a, b, d, e, h\rangle$ can be executed without

Figure 2.19 – A Petri net modeling a healthcare process with a constraint



Source: adapted from Mans, Aalst and Vanwersch (2015), Aalst (2016)

deviations. These examples illustrate how the token replay approach can also be used at run-time for detecting deviations when they occur.

2.6.2 Predict

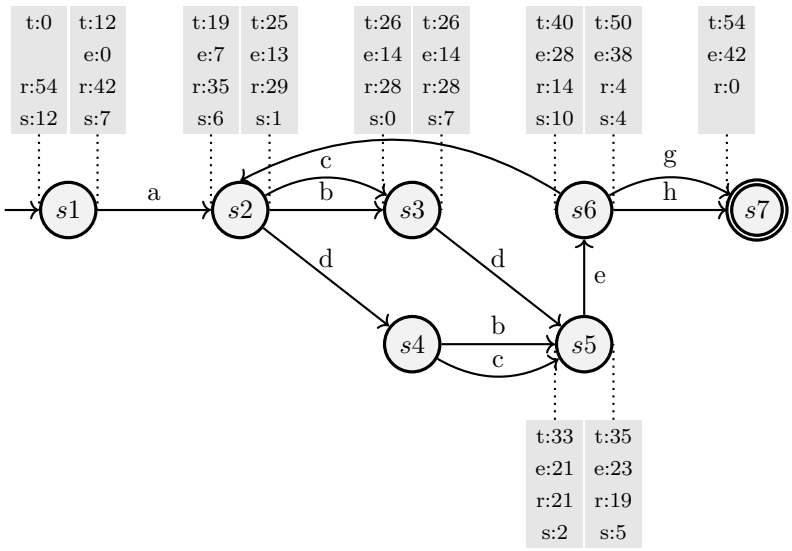
Predict activity keeps track of every activity in the partial trace and uses it to make predictions using a predictive model (FRANCESCOMARINO et al., 2018). The predictive model is based on the completed cases recorded in event logs. Aalst (2016) shows some examples of predictions that can be made:

- The predicted remaining flow time is *13 days*;
- The predicted probability of meeting the legal deadline is *0.13*;
- The predicted total cost of a case is *$130.00*;
- The predicted probability that activity a will occur is *1.31*;
- The predicted probability that person r will work on this case is *0.57*;
- The predicted probability that a case will be rejected is *0.67*; and
- The predicted total service time is *130 minutes*.

Several approaches may be used to make these predictions. Supervised learning techniques such as Classification and Regression can be used (HAND, 2007). Using feature extraction, the properties of the partial trace can be used as predictor variables. The response variable is often a performance indicator, e.g., remaining flow time, total costs, total service time, etc.

As an example, a particular technique (time-annotated transition system from Aalst, Schonenberg and Song (2011)) answering a specific predictive question (the remaining flow time) is presented. Consider the transition system that models the example of a healthcare process presented in Figure 2.5 and a trace with the information about the start and end times of each event: $\sigma = \langle a[12, 19], b[25, 26], d[26, 33], e[35, 40], h[50, 54] \rangle$. In this case, $x[t_s, t_e]$ denotes an event x with a start time t_s and an end time t_e in time units. Figure 2.20 shows the time-annotated transition system for trace σ.

Figure 2.20 – Time-annotated transition system for time-flow prediction



By replaying the trace over the model, it is possible to keep track of time statistics t, e, r, s for each state. This can be done for when each event "arrives" and "leaves" each state, with exception of the end state. t: keeps track of the number of time units since time zero. For instance, activity a starts at t:12 (state $s1$) and ends at t:19 (state $s2$). Activity b starts at t:25 (state $s2$) and ends at t:26 (state $s3$). e: is the elapsed time since the start of the trace. For instance, activity b concludes arriving at state $s3$ at the time t:26 and, since the trace started at t:12, the elapsed time is t:26 - t:12 = e:14. r: is the remaining flow time, that is, the number of time units until the end of the trace. For instance, when activity b leaves state $s2$ the remaining flow time is r:29. When activity h concludes and arrives at *end* state, the remaining flow time is r:0. s: is the sojourn time, the time spent to start the next event. For instance, activity a starts at t:12 (state $s1$) and ends at t:19 (state $s2$), being in this case the sojourn time t:19 - t:12 = s:7.

It is possible to annotate the whole model with time statistics from trace σ. Furthermore, this process can be repeated for every trace in the log. Assuming a large event log, there may be hundreds or even thousands of annotations per state. For each state, it is possible to create an array for each time statistics. For instance, it is possible to have an array of remaining times for activities leaving the state $s3$. Using this collection of remaining times values, it is possible to calculate statistics such as the mean remaining flow time for activity leaving the state $s3$; the maximum/minimum remaining flow time at that point, or even to use a statistical distribution on this sample data. When a partial trace arrives, it is possible to use the statistical data to predict outcomes (such as arriving/leaving time, remaining flow time, elapsed time, etc.) and even associate a likelihood to it.

Outcomes such as "with 95% confidence, the remaining flow time is predicted to be between 10 and 15 minutes" or "38% of similar cases exceed 5 days". According to Aalst (2016), annotated transition systems are one of many approaches that could be used for prediction. For example, the short-term simulation could be used to explore the possible futures of a particular case in a particular state. For further details, several prediction approaches are present in the literature. Francescomarino et al. (2018) presents a literature review about prediction enumerating the approaches to predict numerical and categorical outcomes.

2.6.3 Recommend

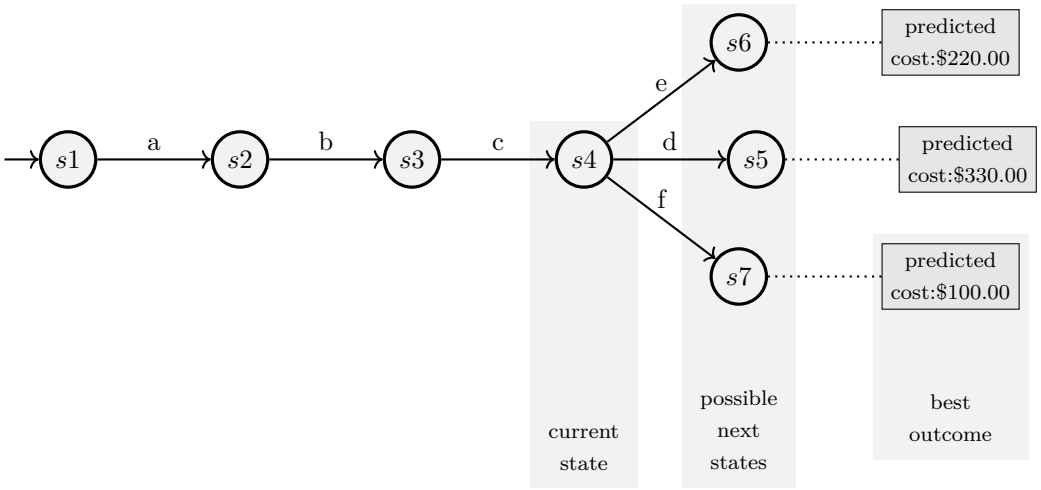
The *recommendation* operational support activity keeps track of every activity in the partial trace and uses it to make recommendations using a model learned from event data. According to Aalst (2016), a recommendation is given concerning a specific goal, such like:

- Minimize the remaining flow time;
- Minimize the total costs;
- Minimize resource usage;
- Maximize the fraction of cases handled within 2 days;
- Maximize the fraction of cases that is accepted; and
- Balance between cost reduction and flow time reduction.

To make recommendations, a *performance indicator* is defined, e.g., remaining flow time or total costs. A recommendation makes statements about possible actions, i.e., the *decision space*.

To illustrate the recommendation process, Figure 2.21 shows an example of recommendation based on predictions. In the example, the partial trace is $\langle a, b, c \rangle$, and the *current state* is $s4$. The operational support system uses the model to identify the possible next states, the *decision space*: $s5, s6$ and $s7$. Considering that the goal of the recommendation is to minimize the cost, it is possible to predict the cost for each state. Respectively, the predicted costs for states $s5, s6$ and $s7$ are \$330.00, \$220.00 and \$100.00. Therefore, the operational support system identifies that \$100.00 is the minimum achievable cost. The state that should be accessed is state $s7$, so the recommended activity to be executed is f.

Figure 2.21 – Example of recommendation based on predictions



The recommendation can include information about its reliability, depending on the prediction technique, e.g., confidence or certainty. For further details about Recommendation, Eili, Rezaeenour and Sani (2021) presents a systematic literature review on process-aware recommender systems.

2.7 Final considerations

Process mining is a new research area that has grown in the last years. Several process mining activities can be performed by analyzing recorded information (event logs) from real-world processes. Process mining allows to discover process models from recorded data. It also allows checking the conformance of cases with respect to a reference model. Enhanced models can be created by including perspectives such as time, resource, and case perspectives. Online operational support activities can be conducted, such as prediction, recommendation, and fault detection. According to (GARCIA et al., 2019), several studies have been conducted that applied process mining in areas such as healthcare, ICT,

manufacturing, education, financial, logistics, public services, security, call centers, robotics, entertainment, utility, pharmacy, hostelry, and agriculture.

3 Cost-aware Process mining

This chapter presents the theoretical references and literature approaches for linking costs to process mining techniques. It is based on the related studies from A.1, describing in detail the approaches that contribute to the goal of this thesis. This chapter explains the cost definition and how costs can be estimated in a process (Section 3.1), how a log can be enhanced with cost information (Section 3.2), how conformance checking can be conducted to consider cost constraints (Section 3.4), and how operational support activities may be done to consider cost outcomes (Section 3.5).

3.1 Costs

According to Kaplan and Atkinson (1998), to carry out activities, organizations acquire and use resources such as people, equipment, materials, external services, and facilities. *Costs* are the amount or equivalent paid or charged from the acquisition and use of these resources and are recorded by the financial system.

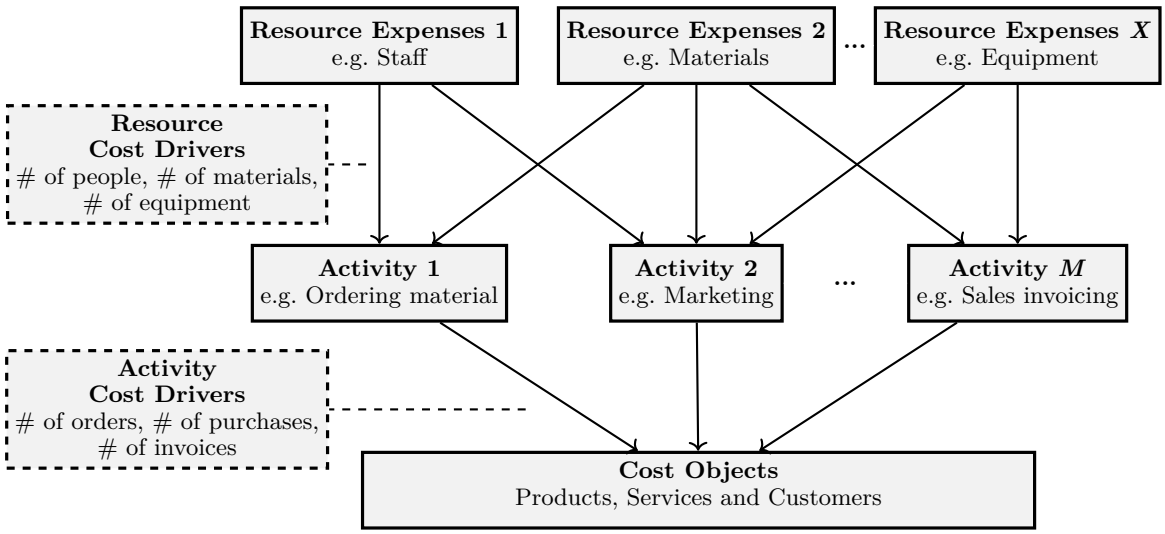
One way organizations can be more profitable is by reducing costs. In an organization where all costs can be attributed directly to a product or service, it is easy to make decisions about cost reduction. However, nowadays, costs are related not only to production but to indirect business functions like administration, marketing, sales, distribution, and IT. The literature has reported various costing techniques based on the notion of activities (SIGUENZA-GUZMAN et al., 2013), being the two main techniques Activity-Based Costing (ABC) and Time-Driven Activity-Based Costing (TDABC) presented in the following 2 subsections.

3.1.1 Activity-Based Costing

The first costing technique is Activity-Based Costing (KAPLAN; ATKINSON, 1998) shown structurally in Figure 3.1. This technique first identifies the M *activities* contributing to the production of a product or delivery of a service. For instance, activities can be ordering material, marketing, or sales invoicing. Next, the X resources expenses are identified. Resources can be staff, materials, equipment and money, for example. Resource expenses are linked to the different activities through the use of *resource cost drivers*. A resource cost driver indicates the amount of resources an activity requires. It considers the per unit cost of each resource for each activity. After that, cost can be aggregated by activity. The next step is identifying how much each *cost object*, such as products or services, require from each activity. Activity costs are linked to cost objects using *activity cost drivers*. An

activity cost driver indicates the number of activities an object utilizes.

Figure 3.1 – Activity-Based Costing Structure



Source: Kaplan and Atkinson (1998), Siguenza-Guzman et al. (2013)

Figure 3.2 – Example of ABC drivers

	Resources			# resources per activity		
	Cost ($)	***#***	***Cost per unit***	***Ordering***	***Marketing***	***Invoicing***
Staff	10,000.00	5	2,000.00	1	1	3
Material	500.00	500	1.00	0	500	0
Equipment	1,000.00	10	100.00	1	3	6
Total	**11,500.00**		**Total per activity:**	**2,100.00**	**2,800.00**	**6,600.00**

(a) Resource Cost Driver

	Activities			# activities per cost object		
	Cost ($)	***#***	***Cost per unit***	***Service 1***	***Service 2***	***Product A***
Ordering	6,300.00	3	2,100.00	1	1	1
Marketing	2,800.00	1	2,800.00	0	1	0
Invoicing	6,600.00	1	6,600.00	0	0	1
Total	**15,700.00**		**Total per cost object:**	**2,100.00**	**4,900.00**	**8,700.00**

(b) Activity Cost Driver

To illustrate the cost drivers, consider the examples of Resource Cost Driver (Figure 3.2a) and of Activity Cost Driver (Figure 3.2b). In this example, the process has three activities: *ordering*, *marketing*, and *invoicing*; and three types of resources: *staff*, *material*, and *equipment*. Also, this illustrational process offers to clients *service 1*, *service 2*, and *product A*.

The Resource Cost Driver in Figure 3.2a maps what every activity needs in terms of resources and its costs. For instance, activity *ordering* needs 1 unit of resource *staff* (one person) and 1 unit of resource *equipment* (one computer). Since 1 unit of staff costs $2,000.00 and 1 unit of equipment costs $100.00, the cost for executing activity *ordering* is $2,100.00. The Activity Cost Driver in Figure 3.2b maps what every cost object (product

or service) needs in terms of activities and its costs. For instance, *service 2* needs activities *ordering* and *marketing* to be performed once each. The total cost for offering *service 2* is $4,900.00.

According to Nauta (2011), the ABC costing technique has some flaws. Information gathering and updating are costly because it needs to be updated with every change in activities and resources. Moreover, the excess capacity of resources is not accounted for. The second costing technique is presented next.

3.1.2 Time-Driven Activity-Based Costing

The second cost technique is Time-Driven Activity-Based Costing and overcomes the difficulties presented in ABC systems (KAPLAN; ANDERSON, 2007). TDABC estimates resource usage using *time equations* to determine the time needed to perform each activity. It assigns resource costs directly to the cost objects using two parameters: the cost per time unit of supplying resource capacity; and an estimate of the time units required to perform an activity or a service. The first parameter is calculated as in Equation 3.1.

$$\mathrm{ct} = \frac{c}{pc} \tag{3.1}$$

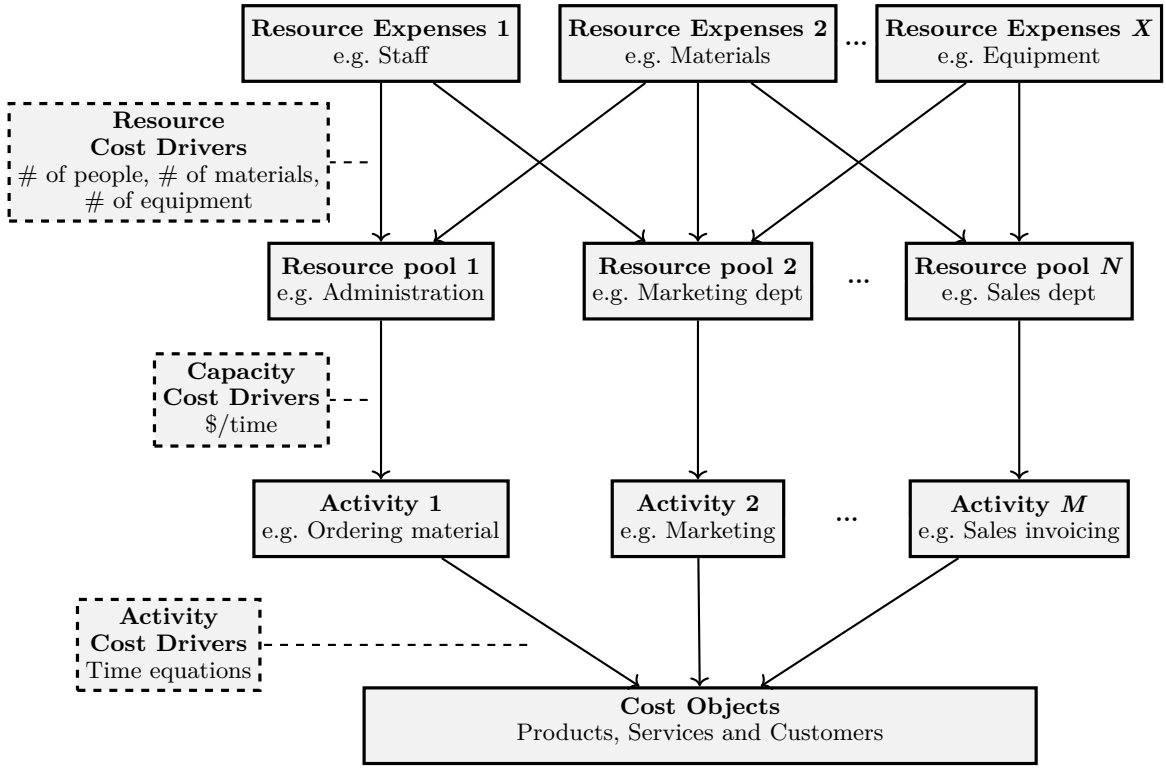
where:

- c is the *cost of supplying resource capacity* in monetary value, i.e., the cost of all the resources supplied to a department or process. For example, for the *marketing department* to be running, there are costs with personnel, supervision, some equipment, technology, and infrastructure. All these costs are aggregated, and the cost of keeping the running marketing department is calculated;
- pc is the *practical capacity* in time units, i.e., the amount of time that the department runs without idle time; and
- ct is the *cost per time unit of supplying resource capacity*.

Figure 3.3 shows structurally the TDABC technique (KAPLAN; ATKINSON, 1998; SIGUENZA-GUZMAN et al., 2013). Initially, in the same way, ABC does, it first identifies the M *activities* and the X resources expenses. Resource expenses are allocated to the activities using drivers and *resource pools*. Resource pools are groups of resources necessary to execute some activity, like a department in a company. Examples of resource pools are administration, marketing, and sales departments. The resource cost driver calculates the quantity of each resource in each resource pool. Then, the *capacity cost driver* links each resource pool to each activity specifying the cost per time unit. The next step is identifying how much each *cost object* requires from each activity. Activity costs are linked to cost

objects using *activity cost drivers*. An activity cost driver indicates what activities an object utilizes. Activity costs are then distributed to cost objects by multiplying the cost per time unit of the resources by estimating the time required to perform the activities. This estimate is made by *time equations*.

Figure 3.3 – Time-Driven Activity-Based Costing Structure



Source: Kaplan and Atkinson (1998), Siguenza-Guzman et al. (2013)

A time equation defines how much time an instance of an activity will take to occur. Time equations can be as simple as constants, calculated from the mean of historical data, for example, or complex involving several variables. For example, consider the resource cost driver in Figure 3.4a, the capacity cost driver in Figure 3.4b, and the activity cost driver in Figure 3.4c.

In this example, the process has three activities: *ordering*, *marketing*, and *invoicing*; three types of resources: *staff*, *material*, and *equipment*; and three cost objects: *service 1*, *service 2*, and *product A*. The activities in this example are executed exclusively by *administration*, *marketing dept*, and *sales dept*, the resource pools. The amount of resources consumed by each resource pool is described in Figure 3.4a along with the cost to execute each resource pool for a given time. For instance, *administration* consumes one resource from *staff* and one from *equipment*, totaling up $2,100.00. Figure 3.4b relates the cost of each resource pool to the time spent by the resource pool to utilize the resources, calculating then the cost per time unit. For instance, *administration* has a cost of $2,100.00 in 700 time units. Therefore, the *cost per time unit* of *administration* is three. Figure

Figure 3.4 – Example of TDABC drivers

	Resources			**# resources per resource pool**		
	Cost ($)	***#***	***Cost per unit***	***Administration***	***Marketing dept***	***Sales dept***
Staff	10,000.00	5	2,000.00	1	1	3
Material	500.00	500	1.00	0	500	0
Equipment	1,000.00	10	100.00	1	3	6
Total	**11,500.00**		**Total per resource pool:**	**2,100.00**	**2,800.00**	**6,600.00**

(a) Resource Cost Driver

Resource pool	***Cost ($)***	**Time**	***Cost per time unit***
Administration	2,100.00	700	3.00
Marketing dept	2,800.00	280	10.00
Sales dept	6,600.00	330	20.00

(b) Capacity Cost Driver

	Activities			**# activities per cost object**		
	Cost ($)	**Time**	***Cost per time unit***	***Service 1***	***Service 2***	***Product A***
Ordering	90.00	30	3.00	1	1	1
Marketing	1,000.00	100	10.00	0	1	0
Invoicing	2,000.00	100	20.00	0	0	1
			Total per cost object:	**90.00**	**1,090.00**	**2,090.00**

(c) Activity Cost Driver

3.4c relates each object (product or service) to the activities necessary to obtain it. For example, for *Service 1*, activity *ordering* needs to be executed. *Ordering* takes 30 time units to be executed, the value obtained using a time equation. Considering the cost per time unit of *ordering* ($3.00), and the time that it takes to *ordering* be executed (30 time units), the cost of *ordering* is $90,00. Since *Service 1* only needs *ordering* to be executed, the cost of *Service 1* is $90,00.

The mentioned costing techniques can allocate direct and indirect costs to activities producing products or delivering services. Next, a standard is presented to represent cost information in event logs, estimated by the costing techniques.

3.2 Event log with costs

Cost information of each activity can be estimated by the techniques from the previous section. Next, the cost information can be included in the event logs. As presented in Section 2.2.1, XES is a tag-based language for representing event logs (GÜNTHER; VERBEEK, 2009). XES defines logs, traces, events, and their attributes. XES also defines *extensions* as primarily a vehicle for attaching semantics to a set of defined attributes per element (IEEE, 1999). The XES standard definition implements the *cost extension* that defines elements to store information about the cost associated with a log. Frame 3.1 shows the description of cost extension elements.

Cost information can be recorded at *trace* or *event* levels. Each trace/event has a *total*, a *currency*, and zero or more *cost elements*. A cost element contains three data

Frame 3.1 – Cost extension elements description

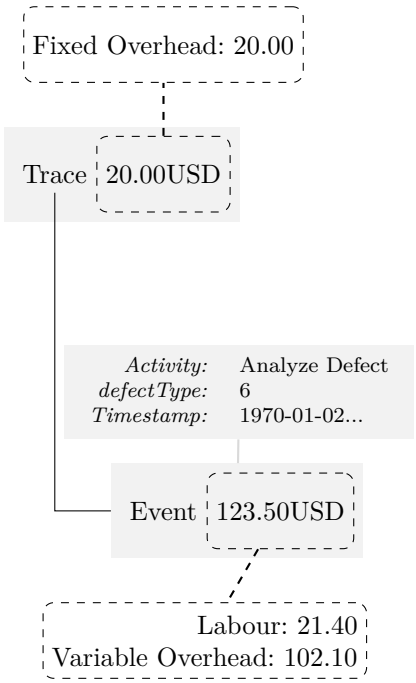
Attribute Level	Key	Type	Description
trace, event	total	float	Total cost incurred for a trace or an event. The value represents the sum of all the cost element amounts within the element.
trace, event	currency	string	The currency of all costs elements of this element in any valid currency format.
meta	amount	float	The value contains the cost amount for a cost element.
meta	driver	string	The value contains the id for the cost element.
meta	type	string	The value contains the cost type of the cost element (e.g., Fixed, Overhead, Materials).

Source: IEEE (1999)

elements *cost amount*, *cost driver*, and *cost type*. The total is the sum of all cost elements amounts in a trace/event. The currency is the same for all the cost elements in a trace/event. Figure 3.5 shows an example of XES with cost information. The example shows one trace with one event. When a case is started, it costs $20. The *total* is set to *"20.00"*, and the *currency* is set as *USD* (American Dollars). These $20 are detailed by the cost element *"xyz123"* of this trace. The cost element has the amount *"20"* and the type *"Fixed Overhead"*.

Figure 3.5 – Example of a trace in XES with cost information

```
...<trace>
  <string key="cost:currency" value="USD" />
  <float key="cost:total" value="20.00">
    <string key="xyz123" value="">
      <float key="cost:amount" value="20.00" />
      <string key="cost:driver" value="xyz123" />
      <string key="cost:type" value="Fixed Overhead" />
    </string>
  </float>
  <event>
    <string key="concept:name" value="Analyze Defect"/>
    <date key="time:timestamp" value="1970-01-02T17
    :10:00.000+10:00"/>
    <string key="defectType" value="6"/>
    <string key="cost:currency" value="USD" />
    <float key="cost:total" value="123.50">
      <string key="d2f4ee27" value="">
        <float key="cost:amount" value="21.40" />
        <string key="cost:driver" value="d2f4ee27" />
        <string key="cost:type" value="Labour" />
      </string>
      <string key="abc124" value="">
        <float key="cost:amount" value="102.10" />
        <string key="cost:driver" value="abc124" />
        <string key="cost:type" value="Var Overhead" />
      </string>
    </float>
  </event>...
</trace>...
```



Source: IEEE (1999)

This means that this trace has a 20 dollars cost to be started due to a fixed overhead cost.

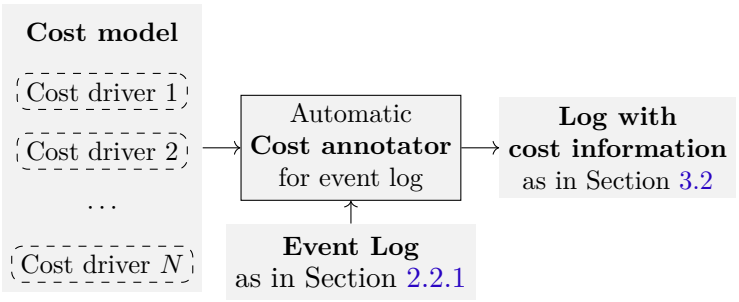
The trace contains one event from activity *Analyze Defect*. This event has a data attribute *defectType* with value *6* and a timestamp (1970-01-02T17:10:00.000+10:00). This event has two cost elements: *"d2f4ee27"* and *"abc124"*. The first cost element represents a *"Labour"* cost of *"21.40"*. The second cost element denotes a *"Variable overhead"* cost of "102.10". This event costs the sum of the individual cost elements (*total* is 123.50) and is expressed in dollars (*currency* in USD).

With XES cost extension, it is possible to represent cost information in the event log, as modeled by cost techniques from Section 3.1. For that purpose, automatic cost annotators may be used, as in the next section.

3.3 Automatic cost annotator for event log

Section 3.2 shows how to represent cost information in event logs through XES cost extension. The cost information may be estimated from the real-world process by using accountability techniques such as ABC and TDABC from Section 3.1. This section presents a method proposed by Wynn et al. (2014) and Nauta (2011) to automatically annotate event logs with cost information based on cost models. The result is a cost-annotated log like the one presented in Section 3.2. Figure 3.6 shows how the above-mentioned method works structurally.

Figure 3.6 – Automatic cost annotator for event log



Source: adapted from Wynn et al. (2014)

First, a *cost model*, containing one or more *cost drivers*, is created to specify how costs can be modeled based on several aspects, such as the occurrence of activities, their duration, data values, etc. Next, the cost model and the event log (without cost information) are given as input to the *cost annotator*. The result is a cost-annotated event log, i.e., a log with cost information.

3.3.1 Creating the cost model

The first step is to create a cost model composed of one or more cost drivers. The cost model is an XML file specifying a business process's cost-related data. The XML schema can be seen in Nauta (2011). A *cost driver* defines how the cost is related to process elements (resource, activity, case data) and the cost rate. According to Wynn et al. (2014), several types of costs can be used, such as:

- the cost rate of resource "Sara" is $50 per hour (variable labour cost);
- each invocation of activity "Examine" costs $20 (fixed processing cost);
- each process instance has an overhead of $100 (fixed overhead cost);
- the cost rate of a resource with a "expert" role performing activity "Examine" is $30 per hour (labor cost depends on a task);
- the fixed cost of activity "Examine" with the data attribute "Age" being ">60" is $50 while activity "Examine" with the data attribute "Age" being "<=60" is $25 (fixed cost depends on a data attribute);
- the variable cost rate of two resources "Sara" and "Sue" working together on an activity "Examine" with data attribute "Age" being ">60" is $200 per hour. (variable/labor cost of multiple resources working together dependent on task and data attribute).

For illustration purposes, consider the event log with no cost information in Figure 3.7. It contains one trace with one event. The event is *Analyze Defect* with timestamp *1970-01-02T17:10:00.000+10:00*, and data attribute *defectType* with value *6*. Also, this event is performed by resource *Tester 4*. This log is used next to illustrate the creation of a cost model and annotation of an event log with cost information.

Figure 3.7 – Example of trace in XES with no cost information

```
...
<trace>...
  <event>
      <string key="concept:name" value="Analyze Defect"/>
      <date key="time:timestamp" value="1970-01-02T17
    :10:00.000+10:00"/>
      <string key="defectType" value="6"/>
      <string key="org:resource" value="Tester4" />
  </event>
</trace>...
```

For illustration purposes, consider that activity *Analyze Defect* has three cost drivers associated with it:

- A *Fixed Cost* of $21.00 for each invocation of activity *Analyze Defect*;
- A *Labour* cost of $4.00 per hour spent by resource *Tester 4* on activity *Analyze Defect*;
- A *Variable Overhead* cost of $1.00 times the defect type for *Analyze Defect*;

These three cost drivers can be expressed in an XML cost model as in Figure 3.8. The driver to implement the *fixed cost* is labeled *fc_ad*. The tag *workflowElements* includes every workflow element involved in the driver. In this case, it is involved in activity *Analyze Defect*. The cost calculus is defined in *unitCost* tag. For the fixed cost driver, it is specified as *$21.00* for each invocation of activity *Analyze Defect*.

The drivers to specify the *Labour* cost and the *Variable Overhead* cost are labeled *l_ad* and *vo_ad*, respectively. *Analyze Defect* activity is involved in both drivers. *l_ad* driver uses the resource *Tester 4* and calculates the cost of $4.00 per hour. *vo_ad* driver uses the value of *defect type*, and its cost is *1* $\times$ *defectType* in dollars. For further details on creating the cost model, see Nauta (2011).

3.3.2 Annotating the log

The cost annotator takes the event log and the cost model as input and calculates the cost-annotated log. First, it calculates for each event and traces the duration. For that, it uses the *timestamp* information. Next, it uses the cost drivers to perform the calculation for each trace and event. Finally, it aggregates the costs calculated for each cost driver for each trace or event.

For illustration, consider the log with no cost information in Figure 3.7 and the cost model in Figure 3.8. First, suppose that the duration of activity *Analyze Defect* is 2 hours. In this case, the cost associated with the event are:

- A *fixed cost* of *$21.00* imposed by *fc_ad* driver;
- A *labour* cost of *$8.00* ($4.00 per hour $\times$ 2 hours of duration) imposed by *l_ad* driver; and
- A *variable overhead* cost of *$6.00* (1 $\times$ the value of *defectType*) imposed by *vo_ad* driver.

The algorithm proposed by Nauta (2011), Wynn et al. (2014) iterates over each trace and, in a trace, over each event. Then, it verifies for each event all drivers in the cost model that have such event as a workflow element. The cost is calculated according to each driver's rule and aggregated to the cost of the event/trace. Using the log with no

Figure 3.8 – Example of cost model in XML

```
<driver id="fc_ad">
   <costType>Fixed Cost</ costType>
   <workflowElements>
      <workflowElement type="task" name="Analyze Defect" />
   </workflowElements>
   <unitCost>
      <amount>21.0</amount>
      <currency>USD</currency>
      <unit>invocation</unit>
   </unitCost>
</driver>
<driver id="l_ad">
   <costType>Labour</ costType>
   <workflowElements>
      <workflowElement type="task" name="Analyze Defect" />
      <workflowElement type="resource" name="Tester 4" />
   </workflowElements>
   <unitCost>
      <amount>4.0</amount>
      <currency>USD</currency>
      <unit>hour</unit>
   </unitCost>
</driver>
<driver id="vo_ad">
   <costType>Variable Overhead</ costType>
   <workflowElements>
      <workflowElement type="task" name="Analyze Defect" />
      <workflowElement type="data" name="defectType" />
   </workflowElements>
   <unitCost>
      <amount>1 * defectType</amount>
      <currency>USD</currency>
      <unit>invocation</unit>
   </unitCost>
</driver>
```

Source: adapted from Wynn et al. (2014)

cost information and the cost model makes it possible to obtain the cost-annotated log in Figure 3.9.

3.4 Conformance and fault detection

In Section 2.4 conformance techniques were presented. They take an event log (or completed trace) and a reference model as input and identify deviations between them. In a similar way, in Section 2.6, the fault detection algorithm takes a partial trace and a normative model, detecting and reporting when a deviation occurs. The difference between the two sections is that conformance checking is done over the completed trace/whole log, while fault detection is achieved by using a partial trace. This section shows how both conformance checking and fault detection can be conducted to consider the cost dimension.

For this purpose, the works of Leoni, Aalst and Dongen (2012), Leoni and Aalst (2013) and Borrego and Barba (2014) are used. These works present the multi-perspective conformance checking that observes not only the "structural" conformance checking, as pre-

sented in Section 2.4, but other perspectives such as data, resources, and time. "Structural" conformance checking/fault detection still needs to be done to identify structural problems, such as the wrong order of activities. Still, this section focuses on the data perspective, assuming that the corresponding techniques have detected structural problems. The data perspective is used because the cost information of each activity can be stored as data. To check the conformance of a complete trace/event log with cost information, and to detect faults in a partial trace with cost information, a Petri net with data (DPN-net) is used.

Figure 3.9 – Example of cost-annotated log

```
<trace>
...
  <event>
  ...
    <string key="concept:name" value="Analyze Defect" />
    <string key="org:resource" value="Tester4" />
    <date key="time:timestamp" value="1970-01-02T17:10:00.000+10:00" />
    <string key="defectType" value="6" />
    <string key="cost:currency" value="USD" />
    <float key="cost:total" value="35.00">
      <string key="fc_ad" value="">
        <float key="cost:amount" value="21.00" />
        <string key="cost:driver" value="fc_ad" />
        <string key="cost:type" value="Fixed Cost" />
      </string>
      <string key="l_ad" value="">
        <float key="cost:amount" value="8.0" />
        <string key="cost:driver" value="l_ad" />
        <string key="cost:type" value="Labour" />
      </string>
      <string key="vo_ad" value="">
        <float key="cost:amount" value="6.0" />
        <string key="cost:driver" value="vo_ad" />
        <string key="cost:type" value="Variable Overhead" />
      </string>
    </float>
  </event>
</trace>
```

Source: adapted from Wynn et al. (2014)

Definition 10: DPN-net

A DPN-net is a Petri Net in which transitions can write variables (SIDOROVA; STAHL; TRČKA, 2011) formally defined as $N = (P, T, F, V, W, G)$ where:

- P, T, F are the set of places, the set of transitions, and the flow relation of a Petri net, respectively;
- V is a set of variables. Each variable $v \in V$ has a specific domain $Dom(v)$;
- W is a write function $W : T \to 2^V$ that labels each transition with a set of write operations. The write operations update the value of the variables before the transitions to a new value after the transition is fired. For clarity purposes, the value of a variable v after the transitions are fired are denoted v'; and
- G is a guard function $G : T \to G_V$ that labels each transition with a guard $g \in G_V$. G_V is the set of guards that can be created with variables in V. A guard is a logical expression that can be evaluated, and its result is *True* or *False*.

Figure 3.10 shows an example of a DPN net similar to the Petri net presented in Figure 2.2. The difference is that some transitions are labeled with write operations ($v' \leftarrow \dots$) and guards ($v < \dots$). This DPN-net represents an example of a certain buying process. In this process, the client order a certain product (activity a), and the order value is captured by the write operation and assigned to Ov variable ($Ov' \leftarrow$Order value). Next, either a 10% discount can be applied to the order (activity b) or not (activity c). The discount should be applied only when the order value exceeds \$100.00. In this case, to transition b are associated a guard that verifies the order value ($Ov > 100$) and a write operation that applies the discount ($Ov' \leftarrow Ov \times 0.9$). At the same time, the delivery department prepares the product for shipping (activity d), and the shipping cost is captured by the write operation and assigned to Sh variable ($Sh' \leftarrow$Shipping).

Next, the invoice is generated (activity e), and the total cost is calculated by summing the order value and the shipping cost ($Tt' \leftarrow Ov + Sh$). Finally, the order can be paid either by a digital transaction (activity g) or by a physical payment (activity h). Physical payments are only allowed when the total cost does not exceed \$1,000.00. For this purpose, a guard is associated to activity h ($Tt <= 1000$).

Structurally, the non-conformance can be checked in the same way that in Section 2.4, and the fault detection can be done as in Section 2.6. To illustrate how cost conformance/fault detection can be made, consider the following two traces:

It can be observed that both traces ($\langle a, c, d, e, g \rangle, \langle a, b, d, e, g \rangle$) are structurally compliant to the model in Figure 3.10. However, the cost information needs to be checked.

Figure 3.10 – Conformance with cost information

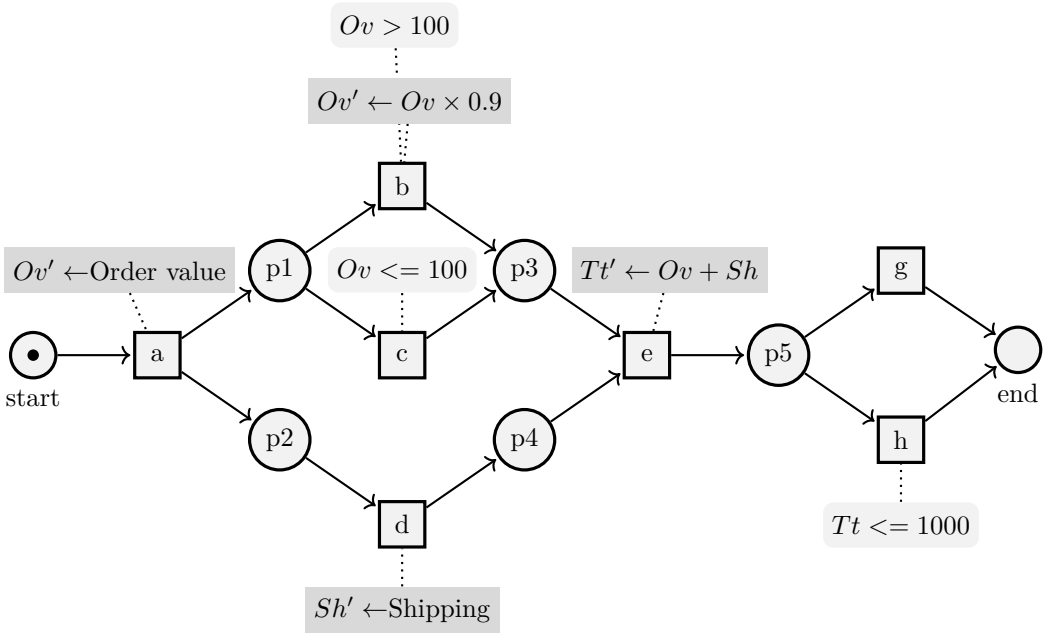
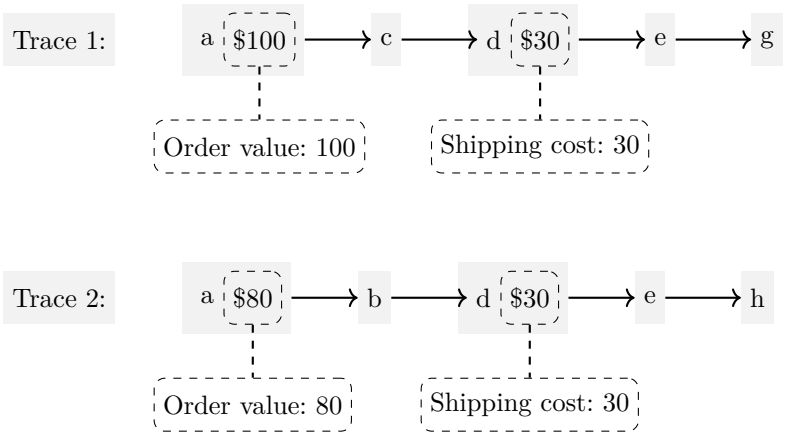


Figure 3.11 – Example of traces with cost information



For trace 1, activity a occurs and assigns \$100 to Ov. Next, activity c occurs, and since $Ov <= 100$, no fault is detected. Shipping is handled (activity d), and \$30 is assigned to Sh. Then, activity e calculates $Tt = Ov + Sh = \$100 + \$30 = \$130$. Finally, the activity g occurs. No cost violation is detected throughout trace 1.

For trace 2, activity a occurs and assigns \$80 to Ov. Next, activity b occurs. In this case, the value of Ov is not greater than \$100, therefore violating the guard in transition b. The other activities (d, e and h) occur normally. Because trace 2 violates a guard, trace 2 is considered not compliant with the reference model.

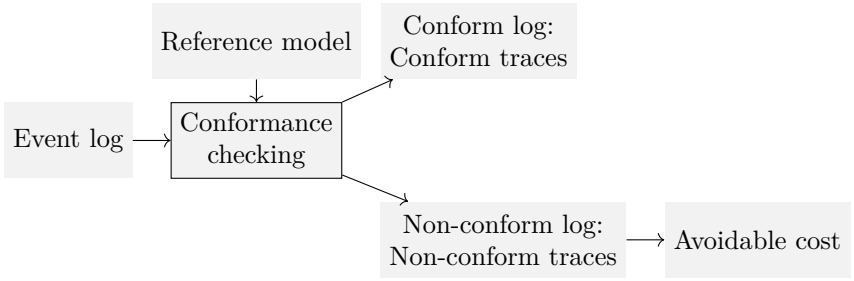
This way, cost information can also be used for conformance checking. Further, Mannhardt et al. (2016a) presents a method to check both structural and value information conformance simultaneously, reducing computational resources.

3.4.1 Avoidable cost

According to Black, Hashimzade and Myles (2012), the avoidable cost is an expense that will not be incurred if a particular activity is not performed. In the case of cost-aware process mining, specially conformance checking, the avoidable cost can be estimated by identifying the traces that are not conform, and calculating the total cost that non-conform traces impose.

Therefore, conformance techniques can be used to split the event log, based on conformity with a reference model. The result of the log splitting is a *conform log* containing the conform traces, and the *non-conform log* containing the non-conform traces. Figure 3.12 illustrates structurally how to estimate avoidable cost based on conformance checking splitting.

Figure 3.12 – Event log splitting based on conformance



The next section presents how prediction and recommendation tasks can be conducted regarding cost information.

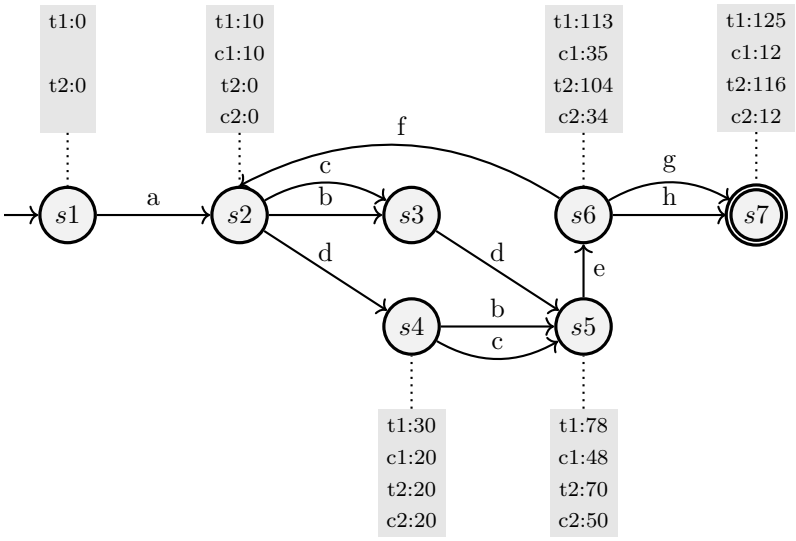
3.5 Prediction and recommendation

Predict activity keeps track of every activity in the partial trace and uses it to make predictions using a predictive model. The predictive model is based on the completed cases recorded in event logs. As an example, a particular technique (cost-annotated transition system) answering a specific predictive question (the total cost of the case) is presented. This is similar to time-prediction presented in Section 2.6.

Consider the transition system that models the example of a healthcare process presented in Figure 2.5 and traces with cost information of each event: $\sigma_1 = \langle a[10], d[20], b[48], e[35], h[12]\rangle$ and $\sigma_2 = \langle a[0], d[20], b[50], e[34], h[12]\rangle$. In this case, $x[c]$ denotes an event x with a total cost c in cost units. Figure 3.13 shows the cost-annotated transition system for traces σ_1 and σ_2.

By replaying the traces over the model, it is possible to keep track of cost stats for both traces `t1, c1, t2, c2` for each state. `c1:` and `c2:` keep track of the cost for the last performed activity in trace 1 and 2 respectively. For example, when activity a is performed,

Figure 3.13 – Cost-annotated transition system for cost-flow prediction



c1:10 represents that a had a cost of 10, for trace 1. When activity e is performed, c2:34 represents that e had a cost of 34, for trace 2. t1: and t2: keep track of the total cost of traces 1 and 2 since the start. For instance, at state $s4$ for trace 1, activities a and d have been performed. The total cost trace at this point is t1:30 because the cost of a is 10 and of b 20.

It is possible to annotate the whole model with cost stats from traces. Furthermore, this process can be repeated for every single trace in the log. Assuming a large event log, it is possible to create an array for each cost stat and to calculate statistics such as the mean cost for performing activity b; the maximum/minimum total cost of traces at a certain state, etc. When a partial trace arrives, it is possible to use the statistical data to predict outcomes (such as total cost) and even associate a likelihood to it. Outcomes such as "with 95% confidence, the total cost of the case is predicted to be between 10 and 15 dollars" or "38% of similar cases exceed 10 dollars".

The cost-annotated log can be used to make recommendations, as described in Section 2.6. The recommendation can include information about its reliability, depending on the prediction technique used.

3.6 Final considerations

Cost is a primordial part of process analysis. Several process mining activities can be performed considering costs. The recorded information (event logs) can be annotated with cost information. Cost information can be estimated from the real-world process using costing techniques such as ABC and TDABC. Event logs with cost information allow us

to check the conformance of cases concerning a reference model and to cost specifications and rules. Cost-annotated transition systems can be used to perform cost prediction and recommendation.

Part II

Multifactor Process Mining Framework

This part shows the main contribution of this thesis. It explores how multifactors can be defined and modeled, how they can be annotated in an event log, and how to perform Process Mining activities with multifactors.

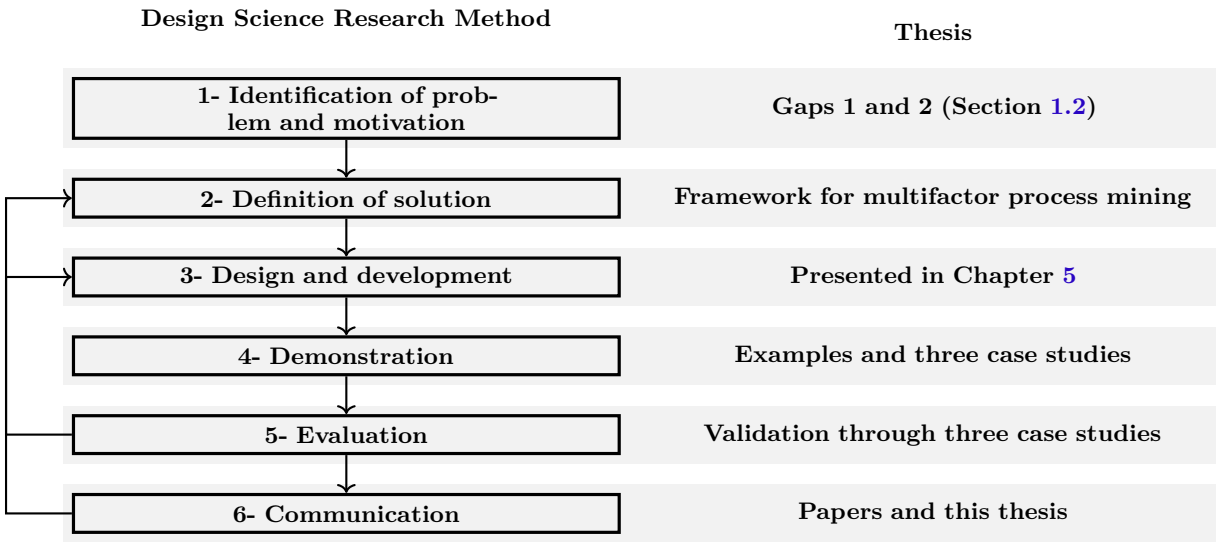
■ Chapter 4 explores the method used in this research. The definition of factors is presented. The multifactor model and the XES extension to represent multifactors are defined.

■ Chapter 5 presents the components of the multifactor framework. For each component, an example is given of how it could be used and what are its inputs and outputs.

4 Methods

The research method used in this thesis is the Design Science Research Method (DSRM) proposed by Peffers et al. (2007). The design science process includes six steps: problem identification and motivation, definition of the objectives for a solution, design and development, demonstration, evaluation, and communication. Figure 4.1 illustrates the steps of the design science research method and the corresponding activities in this thesis.

Figure 4.1 – Design science research method



Source: adapted from Peffers et al. (2007)

The *problem and motivation* (DSRM 1) of this research are related to reducing the two gaps presented in Section 1.2. First, the heterogeneity of the approaches, and the difference between the systems where they are implemented, may hinder the use of cost-aware techniques in process mining. Secondly, no study presents a multifactor capability of modeling and annotating factors to the log and supporting reports, predictions, and other activities with multifactors. The research contribution of this work is to explain how multifactors can be included when performing process mining activities.

The proposed *solution* (DSRM 2) for the identified problem is a computational framework that enables multifactor process mining. This framework supports the following tasks: modeling of multifactors based on aspects of the process such as duration, the occurrence of events, traces, and data attributes; annotation of modeled multifactors in the event log; construction of reports based on multifactor information; factor-based prediction, recommendation, conformance checking, and process model enhancement using the multifactor information; and representation of annotated event log with multifactor information as a data frame suitable to other data mining activities.

The framework was *designed and developed* (DSRM 3) modularly, that is, a set of software components with specifications and defined input and output forms the framework. The details of each component are presented in Chapter 5. In summary, the framework is set to have the following components:

1. Multifactor annotator;
2. Factor-based color enhancement component;
3. Multifactor conformance check component;
4. Reporting component;
5. Prediction and recommendation component;
6. Data mining component.

To *demonstrate* (DSRM 4) the use of the framework, an example is given for each of its components. Also, the framework is applied to four real-world cases in the following areas: education, healthcare, and telecommunication.

To *evaluate* (DSRM 5) the framework, the Case Study Design Science Research Method (CSDSRM) was applied (LEE; RINE, 2004). The CSDSR is a qualitative software validation method in which the software is applied to selected case studies. The results of each case study are analyzed and discussed by a specialist from the area. This validation method does not evaluate quantitative metrics such as runtime and memory usage.

Three case studies were used to validate the framework using CSDSRM: a Brazilian telecommunication company that provides telephony, television, and broadband Internet subscription services (CS1); an educational dataset from a Brazilian public university (CS2); and a healthcare dataset from a Brazilian surgery center (CS3). Frame 4.1 shows the components used in each case study.

Frame 4.1 – Case studies vs. framework components used

Components	**CS1**	**CS2**	**CS3**
Multifactor model		✓	✓
Multifactor annotator		✓	✓
Factor-based color enhancement component		✓	
Multifactor conformance check component		✓	
Reporting component		✓	✓
Prediction and recommendation component		✓	
Data mining component	✓		

For validation purposes, a qualitative questionnaire was applied. The questionnaire follows the ISO 9241-11 (2018) standard to check if the software conforms to user expectations in usefulness and correctness. The questionnaire is based on the usability scale proposed by Brooke et al. (1996). The full questionnaire is presented in Appendix E.

The questionnaire contains a set of statements about each component and the framework overall. Each statement is related to an aspect: usefulness or correctness. Each statement is followed by a Likert scale (LIKERT, 1932) of 5 points: strongly disagree, disagree, neutral, agree, and strongly agree. Statements with an odd number (1, 3, ..., 23, 25) are positive statements and can be evaluated from 0 (strongly disagree) to 4 (strongly agree). Statements with an even number (2, 4, ..., 22, 24) are negative statements and can be evaluated from 4 (strongly disagree) to 0 (strongly agree). Mixing positive and negative statements prevents response biases caused by respondents not having to think about each statement (CHYUNG; BARKIN; SHAMSY, 2018). The relation between statements, framework components, and usefulness or correctness is shown in Frame 4.2.

Frame 4.2 – Questionnaire statements vs. components

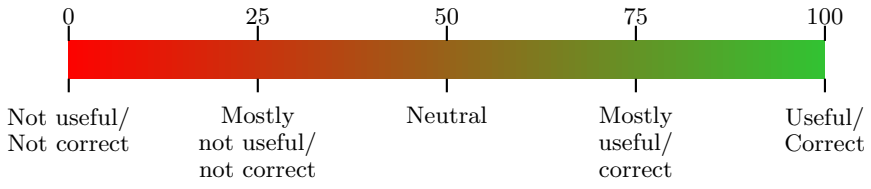
i	**Components**	Usefulness	Correctness
0	Process mining with multifactors	1	2
1	Multifactor model and annotator	3	4
2	Factor-based color enhancement component	5	6
3	Multifactor conformance check component	7, 8	9
4	Reporting component	11, 12, 13, 14, 15, 21	10, 16, 17, 18, 19, 20
5	Prediction and recommendation component	23	22
6	Data mining component	25	24

The usefulness questions were submitted to process mining researchers and students. A video explaining each component and the overall framework was presented along with the questionnaire. The usefulness and correctness questions were applied to process specialists for each case study. However, only the questions related to components used in the corresponding case study were applied (see Frame 4.1).

The usefulness score (Us_i) and correctness score (Cs_i) of each component i can be calculated by the average of all answers for the corresponding statements in Frame 4.2. For instance, the usefulness score for reporting component (Us_4) is the average of all answers for statements 11, 12, 13, 14, 15, and 21, and the correctness score for the data mining component (Cs_6) is the average of all answers for statement 24. All the scores are multiplied by 25, so the final score value can vary from 0 to 100 (Figure 4.2). The framework's overall usefulness score (Us) and correctness score (Cs) can be calculated by the average of the components' scores:

$$Us = \sum_{i=0}^{6} Us_i \qquad\qquad Cs = \sum_{i=0}^{6} Cs_i$$

Figure 4.2 – Us and Cs score scale



Frame E.1 shows the answers for the questionnaire, and Frame 4.3 shows the obtained scores for each component and the overall framework:

Frame 4.3 – Usefulness and correctness scores

i	**Components**	Usefulness	Correctness
0	Process mining with multifactors	97.2	100.0
1	Multifactor model and annotator	77.8	100.0
2	Factor-based color enhancement component	93.8	100.0
3	Multifactor conformance check component	82.8	100.0
4	Reporting component	85.7	97.9
5	Prediction and recommendation component	90.6	100.0
6	Data mining component	75.0	75.0
	Overall framework	85.7	96.9

The correctness score of each component is in the *mostly correct* - *correct* range (≥ 75.0), being the overall framework correctness score 96.9 given by the process specialists. The overall framework usefulness score is 85.7. In general, process mining students and researchers think it is useful to include multifactors in process mining analysis (score 97.2) especially coloring a process model based on a factor value (score 93.8), making predictions and recommendations (score 90.6), and creating reports (score 85.7). Although the multifactor model and annotator, data mining component, and conformance check component have the lowest usefulness scores, they are still considered useful by most answers (scores ≥ 75.0).

This thesis and papers related to case studies are the *communication* (DSRM 6) of the research results. The following sections explore the method used to model and represent multifactors in an event log.

4.1 Factors

A factor F is a set of specifications to assign *values* (v_e) to some events (or cases, v_c) in an event log. A factor defines a *domain* (dom_F) such that all assigned values are in this domain. A factor can also assign a set of *element values* (Lv_e) that are also in dom_F. The set of element values is used to calculate the assigned value using an aggregation function (agg_F). Formally:

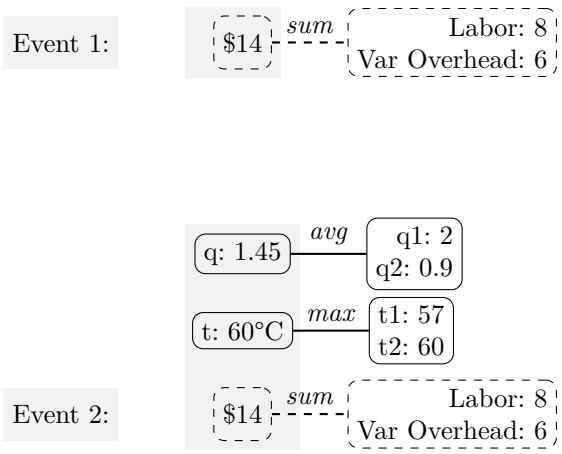
Definition 11: Factor

A *factor* F for a event log with set of cases C and set of events E is a tuple $(dom_F, agg_F, E_F : E \rightharpoonup V_e, C_F : C \rightharpoonup V_c)$ such that:

- $dom_F \subseteq \mathbb{R}$ is the factor F *domain*;
- agg_F is the *aggregation function* for factor F;
- $E_F : E \rightharpoonup V_e$ is a partial map from each event $e \in E$ to a pair $(v_e, Lv_e) \in V_e$ such that $v_e \in dom_F$, $agg_F(Lv_e) = v_e$, and, for each $lv_e \in Lv_e$, $lv_e \in dom_F$.
- $C_F : C \rightharpoonup V_c$ is a partial map from each case $c \in C$ to a pair $(v_c, Lv_c) \in V_c$ such that $v_c \in dom_F$, $agg_F(Lv_c) = v_c$, and, for each $lv_c \in Lv_c$, $lv_c \in dom_F$.

For instance, Figure 4.3 shows an event with cost information (event 1) and an event with multifactor information (event 2). In this example, three factors are illustrated: cost (in dashed lines), q is a quality indicator, and t is a temperature indicator.

Figure 4.3 – Example of events with one factor - cost - (event 1) and multifactors (event 2)



In the example, event 1 has cost information associated with it: a value of \$14 that can be calculated by the *sum* of the element values: \$8 from labor and \$6 from variable overhead. In event 2, the same cost information is presented besides factors q and t. Factor q is composed of the *average* of $q1$ and $q2$. Factor t is the *maximum* value between $t1$, and $t2$. Explicitly:

Factor	$\boldsymbol{v_e}$	$\boldsymbol{agg_F}$	$\boldsymbol{Lv_e}$
Cost	14.0	sum	[8.0, 6.0]
Temperature t	60.0	max	[57.0, 60.0]
Quality q	1.45	avg	[2.0, 0.9]

4.2 XES multifactor extension

An XES multifactor extension is proposed. The cost factor is implemented in the same way as the cost extension for retro compatibility. Frame 4.4 shows the description of multifactor extension elements. The XES cost extension presented in Section 3.2 was adapted to include the multifactor information. The idea is to consider the cost information as one of many factors that may be included.

Frame 4.4 – Multifactor extension elements description

Attribute Level	**Key**	**Type**	**Description**
meta	factor	string	The value contains the id for the factor.
meta	unit	string	The value contains the unit for the factor.
meta	aggreg	string	The value contains the type of aggregation function for the factor: average, maximum, minimum, or sum.
meta	value	float	Factor value for a trace v_c or an event v_e.
meta	amount	float	The value for a factor element $lv_e \in Lv_e$ or $lv_c \in Lv_c$.
meta	driver	string	The id for a factor drive.
meta	type	string	The type of factor drive.

Cost information can be recorded at *trace* or *event* levels, having a *total*, a *currency*, and zero or more *cost elements* (*cost amount*, *cost driver* and *cost type*), in the same way as the cost extension presented in Section 3.2. Zero or more other *factors* can be used along with cost information. A factor has a *factor id*, a *factor value*, a *factor aggregation function*, and a *factor unit*. Factors can be associated at trace or event levels. Each factor is composed of one or more factor elements. A factor element has an *amount*, a *driver id*, and a *type*.

Figure 4.4 shows an example of XES with multifactor information. The example shows one trace with one event (trace 2 from Figure 4.3). The code in green is the additional code to represent the multifactors. The trace contains one event that has cost, quality, and temperature factors, each one with two elements.

4.3 Multifactor model

The multifactor model is an XML file that specifies multifactor-related data for a business process. A *factor driver* defines how a factor is related to process elements (resource, activity, case data), the factor rate, and how the factor is calculated. The multifactor model should have the following tags:

- **<factor>:** contains the specifications of a certain factor. The model can have one or more factor tags. The inner tags are:

Figure 4.4 – Example of XES with multifactor information

```
<event>
  <string key="cost:currency" value="USD" />
  <float key="cost:total" value="14.00">
    <string key="l_cost_ad" value="">
      <float key="cost:amount" value="8.0" />
      <string key="cost:driver" value="l_cost_ad" />
      <string key="cost:type" value="Labour" />
    </string>
    <string key="vo_cost_ad" value="">
      <float key="cost:amount" value="6.0" />
      <string key="cost:driver" value="vo_cost_ad" />
      <string key="cost:type" value="Var Overhead" />
    </string>
  </float>
  <string key="quality" value="">
    <string key="multifactor:factor" value="quality" />
    <string key="multifactor:aggreg" value="average" />
    <float key="multifactor:value" value="1.45">
      <string key="p_qual_ad" value="">
        <float key="multifactor:amount" value="0.9" />
        <string key="multifactor:driver" value="p_qual_ad" />
        <string key="multifactor:type" value="q1" />
      </string>
      <string key="t_qual_ad" value="">
        <float key="multifactor:amount" value="2.0" />
        <string key="multifactor:driver" value="t_qual_ad" />
        <string key="multifactor:type" value="q2" />
      </string>
    </float>
  </string>
  <string key="temperature" value="">
    <string key="multifactor:factor" value="temperature" />
    <string key="multifactor:factorUnit" value="Celsius" />
    <string key="multifactor:aggreg" value="max" />
    <float key="multifactor:value" value="60.0">
      <string key="s_temp_ad" value="">
        <float key="multifactor:amount" value="57.0" />
        <string key="multifactor:driver" value="s_temp_ad" />
        <string key="multifactor:type" value="t1" />
      </string>
      <string key="v_temp_ad" value="">
        <float key="multifactor:amount" value="60.0" />
        <string key="multifactor:driver" value="v_temp_ad" />
        <string key="multifactor:type" value="t2" />
      </string>
    </float>
  </string>
</event>
```

  - **<aggreg>:** defines the aggregation function to be applied to the factor elements.
  - **<isCost>** (optional): specifies if the factor is the cost factor.
  - **<string key="driver_id">:** defines one or more cost drivers to be used for the factor.

- **<drive id="driver_id">:** contains the specifications of a certain drive, responsible for defining how a factor element is calculated. The model can have one or more driver tags. The inner tags are:

- **<factorType>:** a description of the factor element.
- **<workflowElements>:** a list of <workflowElement>. Each tag defines a workflow element to be used by the driver to perform the calculation. An element defines a *type* (task, resource, data, etc.) and a *name*.
- **<unitFactor>:** defines how the factor calculation is performed using the workflow elements:
  - **<amount>:** a real number; or a equation using a *data* workflow element.
  - **<unit>:** the unit used in the calculation. It can be a time unit (seconds, minutes, hours, days) or by invocation of activity.
  - **<factorUnit>** (optional): a descriptive unit to be used for the factor element. Example: USD, Celsius, etc.

4.3.1 Example of multifactor model

Consider the event log with no cost information in Figure 4.5 for illustration purposes. It contains one trace with one event. The event is *Analyze Defect* with timestamp *1970-01-02T17:10:00.000+10:00*, and data attribute *defectType* with value *6*. Also, this event is performed by resource *Tester 4*. This log is then used to illustrate the creation of a multifactor model and annotation of an event log with multifactor information.

Figure 4.5 – Example of trace in XES with no cost information

```
...
<trace>...
  <event>
      <string key="concept:name" value="Analyze Defect"/>
      <date key="time:timestamp" value="1970-01-02T17
    :10:00.000+10:00"/>
      <string key="defectType" value="6"/>
      <string key="org:resource" value="Tester4" />
  </event>
</trace>...
```

Consider that activity *Analyze Defect* has six factor drivers associated with it. Three factors are used as examples: *cost*, *quality*, and *temperature*. These factor drivers can be expressed in an XML factor model as in Figures 4.6, 4.7, and 4.8. Like the cost drivers from Chapter 3, the tag *workflowElements* includes every workflow element that is involved in the driver. The factor calculus is defined in *unitFactor* tag. The details of each factor driver, and the corresponding XML tag, are described next:

- For *cost* factor, the total cost is assigned by *summing* the individual costs of the factor drivers. Two drivers are considered:
  - A *Labour* cost of $4.00 per hour spent by resource *Tester 4* on activity *Analyze Defect*;

– A *Variable Overhead* cost of \$1.00 times the defect type for *Analyze Defect*;

In that way, the cost factor of activity *Analyze Defect* is expressed as:

$$\text{Cost of } \textit{Analyze Defect} = \$4.00 \times \textit{duration} + \$1.00 \times \textit{defectType}$$

Observe that cost drivers from Chapter 3 are a specific case of a factor driver like this one.

Figure 4.6 – Example of factor drivers for ***factor cost*** in XML

```
...
<driver id="l_cost_ad">
   <factorType>Labour</ factorType>
   <workflowElements>
      <workflowElement type="task" name="Analyze Defect" />
      <workflowElement type="resource" name="Tester 4" />
   </workflowElements>
   <unitFactor>
      <amount>4.0</amount>
      <factorUnit>USD</factorUnit>
      <unit>hour</unit>
   </unitFactor>
</driver>
<driver id="vo_cost_ad">
   <factorType>Variable Overhead</ factorType>
   <workflowElements>
      <workflowElement type="task" name="Analyze Defect" />
      <workflowElement type="data" name="defectType" />
   </workflowElements>
   <unitFactor>
      <amount>1 * defectType</amount>
      <factorUnit>USD</factorUnit>
      <unit>invocation</unit>
   </unitFactor>
</driver>...
```

- For *quality* factor, the quality is assigned by *averaging* the individual quality of the factor drivers. Two drivers are considered:

  – A *Procedure quality* of 0.9 for each invocation of activity *Analyze Defect*;

  – A *Time quality* of 1.0 × the duration of activity *Analyze Defect* in hours;

  In that way, the quality factor of activity *Analyze Defect* is expressed as:

$$\text{Quality of } \textit{Analyze Defect} = \frac{0.9 + (1.0 \times \textit{activity duration})}{2}$$

- For *temperature* factor, the temperature is assigned by *considering the maximum* of the individual temperature of the factor drivers. Two drivers are considered:

  – A *Standard* temperature of 57°C for each invocation of activity *Analyze Defect*;

Figure 4.7 – Example of factor drivers for ***factor quality*** in XML

```
...
<driver id="p_qual_ad">
   <factorType>Procedure quality</ factorType>
   <workflowElements>
      <workflowElement type="task" name="Analyze Defect" />
   </workflowElements>
   <unitFactor>
      <amount>0.9</amount>
      <unit>invocation</unit>
   </unitFactor>
</driver>
<driver id="t_qual_ad">
   <factorType>Time quality</ factorType>
   <workflowElements>
      <workflowElement type="task" name="Analyze Defect" />
   </workflowElements>
   <unitFactor>
      <amount>1.0</amount>
      <unit>hour</unit>
   </unitFactor>
</driver>...
```

– A *Variable* temperature of 10°C times the defect type for *Analyze Defect*;

Hence, the temperature factor of activity *Analyze Defect* is expressed as:

$$\text{Temperature of } \textit{Analyze Defect} = max\left(57^{\circ}C, 10^{\circ}C \times defectType\right)$$

The multifactor model configuration for the above-mentioned factor drivers is shown in Figure 4.9.

Figure 4.8 – Example of factor drivers for ***factor temperature*** in XML

```
    ...
<driver id="s_temp_ad">
   <factorType>Standard Temperature</ factorType>
   <workflowElements>
      <workflowElement type="task" name="Analyze Defect" />
   </workflowElements>
   <unitFactor>
      <amount>57</amount>
      <factorUnit>Celsius</factorUnit>
      <unit>invocation</unit>
   </unitFactor>
</driver>
<driver id="v_temp_ad">
   <factorType>Variable temperature</ factorType>
   <workflowElements>
      <workflowElement type="task" name="Analyze Defect" />
      <workflowElement type="data" name="defectType" />
   </workflowElements>
   <unitFactor>
      <amount>10* defectType</amount>
      <factorUnit>Celsius</factorUnit>
      <unit>invocation</unit>
   </unitFactor>
</driver>...
```

Figure 4.9 – Example of multifactor configuration in XML

```
<factor>
   <aggreg>sum</aggreg>
   <isCost>True</isCost>
   <string key="l_cost_ad" value="">
   <string key="vo_cost_ad" value="">
</factor>
<factor>
   <aggreg>average</aggreg>
   <string key="p_qual_ad" value="">
   <string key="t_qual_ad" value="">
</factor>
<factor>
   <aggreg>max</aggreg>
   <string key="s_temp_ad" value="">
   <string key="v_temp_ad" value="">
</factor>
```

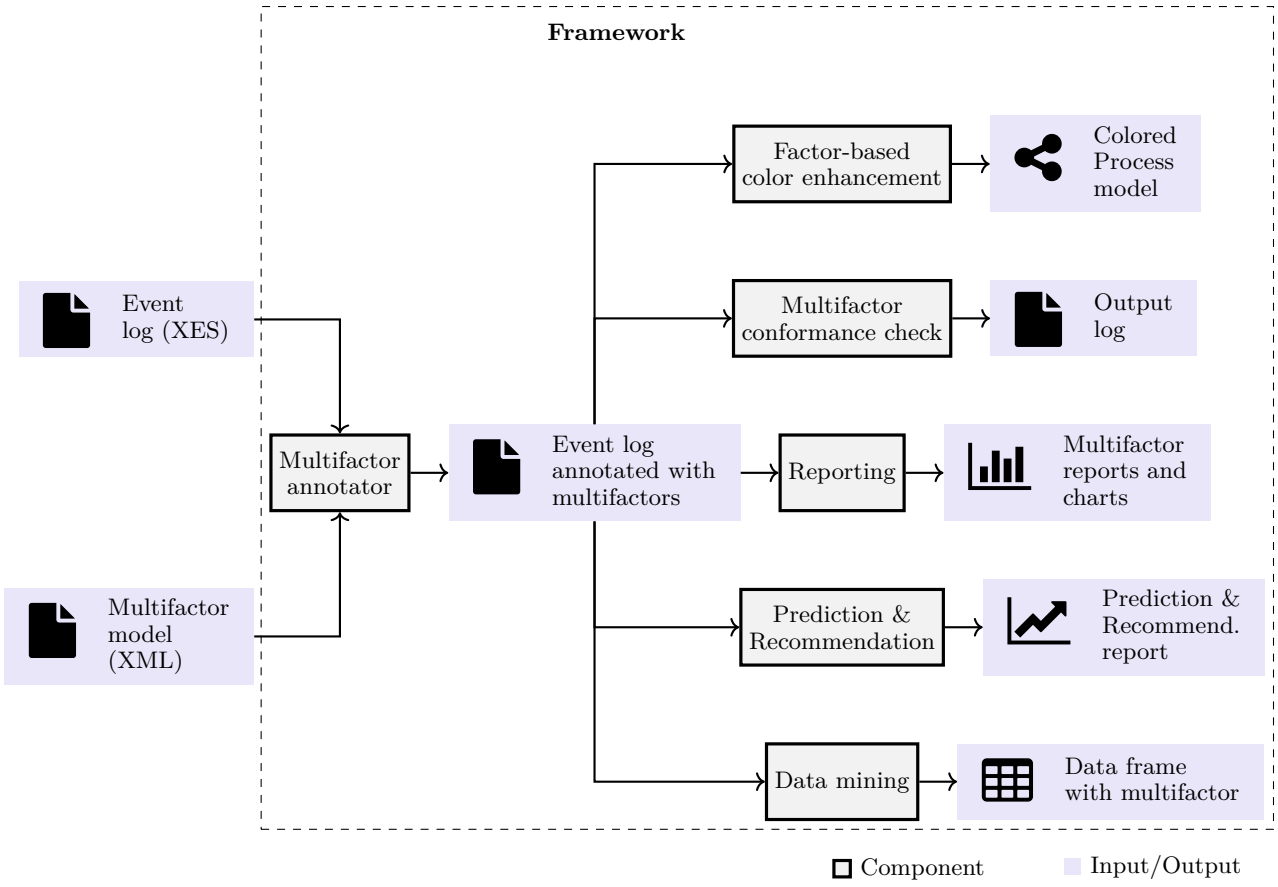
4.4 Final considerations

This Chapter shows how the Design Science Research Method was used in the context of this thesis. Also, it is described how the framework is applied to four real-world cases for validation purposes. The validation was conducted in real-world case studies and endorsed by domain specialists. The definition of factors, how to model them (multifactor model), and how to represent factors in an event log (XES multifactor extension) are presented.

5 Framework

As presented in Chapter 1, the main objective of this thesis is to present a framework for multifactor Process Mining. The structure of the framework is presented in Figure 5.1.

Figure 5.1 – Proposed framework for multifactor process mining



This chapter describes each framework component in detail, presenting an example of use and the component's interface. The components are:

1. The **multifactor annotator** responsible for creating an event log annotated with multifactors: an XES file with the multifactor extension defined in Section 4.2. The component receives an XES event log and a *XML multifactor model* as defined in Section 4.3.

2. The **factor-based color enhancement** that colors a process model based on a specific factor, such as cost. The component receives an event log and outputs the colored model. The details are described in Section 5.2.

3. The **multifactor conformance check** component uses data constraints to check conformance rules over multifactors. The component splits the input event log into

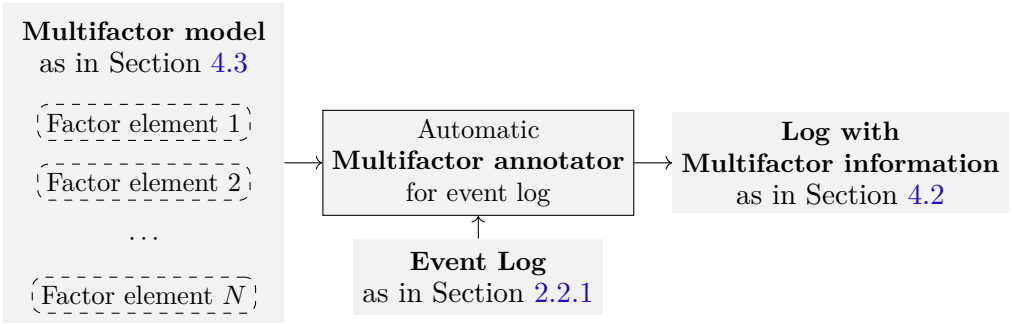
conform and non-conform logs for further analysis. The details are described in Section 5.3.

4. The **reporting** component, described in Section 5.4, takes as input the multifactor event log and displays reports with tables and charts such as:
   - multifactor per case and activity; and aggregated by resource or time (daily, weekly, monthly, or year)
   - multifactor timeline histograms;
   - multifactor heatmaps;
   - multifactor statistical analysis;
5. The **prediction and recommendation** component calculates a multifactor-annotated process model and uses it to make predictions and recommendations based on a statistical analysis of the historical data. The details are described in Section 5.5.
6. The **data mining** component takes the multifactor event log as input and gives as output a data frame suitable to data mining activities. The details are described in Section 5.6

5.1 Multifactor annotator

The multifactor annotator is the component responsible for creating an annotated event log with multifactor information, as depicted in Figure 5.2. First, a *multifactor model*, containing one or more *factor elements*, is created to specify how multifactors can be modeled based on several aspects, such as the occurrence of activities, their duration, data values, etc. Next, the multifactor model and the event log (without multifactor information) are given as input to the *multifactor annotator*.

Figure 5.2 – Automatic multifactor annotator



Algorithm 1 is the responsible for the annotation and was adapted from the cost annotator algorithm proposed by Nauta (2011). First, it calculates the duration of every event in every trace in the log (line 4). Next, it verifies for each event if there are drivers

Algorithm 1: Multifactor annotator

```
   Input: Multifactor Model (XML), Event Log (XES)
   Output: Multifactor-annotated Event Log (XES)
 1 foreach Factor in Multifactor Model do
 2     foreach Trace in the Event Log do
 3         foreach Event in the Trace do
 4             calculate Event duration;
 5             foreach Driver in Factor that is applicable to Event do
 6                 if Factor is Cost then
 7                     calculate cost information according to Driver;
 8                     annotate Event with cost information according to cost extension;
 9                 else
10                     calculate factor information according to Driver;
11                     annotate Event with factor information according to multifactor extension;
12                 end
13             end
14         end
15         calculate Trace duration;
16         foreach Driver in Factor that is applicable to Trace do
17             if Factor is Cost then
18                 calculate cost information according to Driver;
19                 annotate Trace with cost information according to cost extension;
20             else
21                 calculate factor information according to Driver;
22                 annotate Trace with factor information according to multifactor extension;
23             end
24         end
25     end
26 end
```

that apply to that event (line 5). The calculation is performed by the driver using the relevant event information. It may use event data or event duration (lines 7 for cost factor, and 10 for other factors). If the driver is for factor cost, then the information is annotated using cost extension (line 8). If the driver is not for factor cost, then the information is annotated using the multifactor extension and considering the specified aggregation function (11). Traces are annotated in the same way (lines 15 to 22).

5.1.1 Example

By using the multifactor model, composed by the factor drivers in Figures 4.6, 4.7, and 4.8, and by the multifactor configuration in Figure 4.9, it is possible to annotate the event log in Figure 4.5 with multifactor information, resulting in the multifactor-annotated event log shown in Figure 5.3. For this case, suppose that the duration of activity *Analyze Defect* is 2 hours. In this case, the factors associated with the event are:

- A *labour* cost of *$8.00* ($4.00 per hour × 2 hours of duration) imposed by *l_cos_ad* driver;
- A *variable overhead* cost of *$6.00* (1 × the value of *defectType*) imposed by *vo_cost_ad* driver;
- A *Procedure quality* of 0.9 imposed by *p_qual_ad* driver;

Figure 5.3 – Example of XES with multifactor information

```
...
<trace>
  ...
  <event>
    <string key="concept:name" value="Analyze Defect" />
    <date key="time:timestamp" value="1970-01-02T17:10:00.000+10:00" />
    <string key="defectType" value="6" />
    <string key="org:resource" value="Tester4" />
    <string key="cost:currency" value="USD" />
    <float key="cost:total" value="14.00">
      <string key="l_cost_ad" value="">
        <float key="cost:amount" value="8.0" />
        <string key="cost:driver" value="l_cost_ad" />
        <string key="cost:type" value="Labour" />
      </string>
      <string key="vo_cost_ad" value="">
        <float key="cost:amount" value="6.0" />
        <string key="cost:driver" value="vo_cost_ad" />
        <string key="cost:type" value="Variable Overhead" />
      </string>
    </float>
    <string key="quality" value="">
      <string key="multifactor:factor" value="quality" />
      <string key="multifactor:aggreg" value="average" />
      <float key="multifactor:value" value="1.45">
        <string key="p_qual_ad" value="">
          <float key="multifactor:amount" value="0.9" />
          <string key="multifactor:driver" value="p_qual_ad" />
          <string key="multifactor:type" value="Procedure quality" />
        </string>
        <string key="t_qual_ad" value="">
          <float key="multifactor:amount" value="2.0" />
          <string key="multifactor:driver" value="t_qual_ad" />
          <string key="multifactor:type" value="Time quality" />
        </string>
      </float>
    </string>
    <string key="temperature" value="">
      <string key="multifactor:factor" value="temperature" />
      <string key="multifactor:factorUnit" value="Celsius" />
      <string key="multifactor:aggreg" value="max" />
      <float key="multifactor:value" value="60.0">
        <string key="s_temp_ad" value="">
          <float key="multifactor:amount" value="57.0" />
          <string key="multifactor:driver" value="s_temp_ad" />
          <string key="multifactor:type" value="Standard Temperature" />
        </string>
        <string key="v_temp_ad" value="">
          <float key="multifactor:amount" value="60.0" />
          <string key="multifactor:driver" value="v_temp_ad" />
          <string key="multifactor:type" value="Variable Temperature" />
        </string>
      </float>
    </string>
  </event>
</trace>...
```

- A *Time quality* of 2.0 (1.0 × 2 hours of duration) imposed by *t_qual_ad* driver;
- A *Standard Temperature* of 57°C imposed by *s_temp_ad* driver; and
- A *Variable temperature* of 60°C (10°C × the value of *defectType*) imposed by *v_temp_ad* driver.

The cost associated with the event is \$14.00 (\$8.00 + \$6.00). The quality associated is 1.45 $\left(\frac{2+0.9}{2}\right)$. The temperature is 60°C (max(57°C,60°C)).

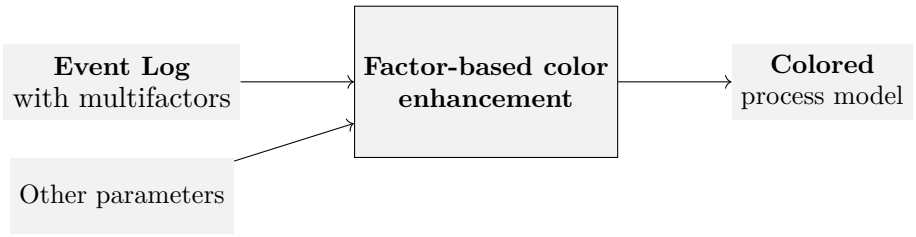
5.2 Factor-based color enhancement

Several process mining software can present their process models enhanced with colors, but in general, the colors represent process elements such as frequency of activities or duration (FLUXICON, 2023; PM group of Fraunhofer FIT, 2021; IBGE, 2023). We propose a component so the process model can be colored based on a specific factor, such as cost. The key idea is to map each possible value that a factor can assume (dom_F) to a color. For that, color maps are used:

Definition 12: Color, Color map

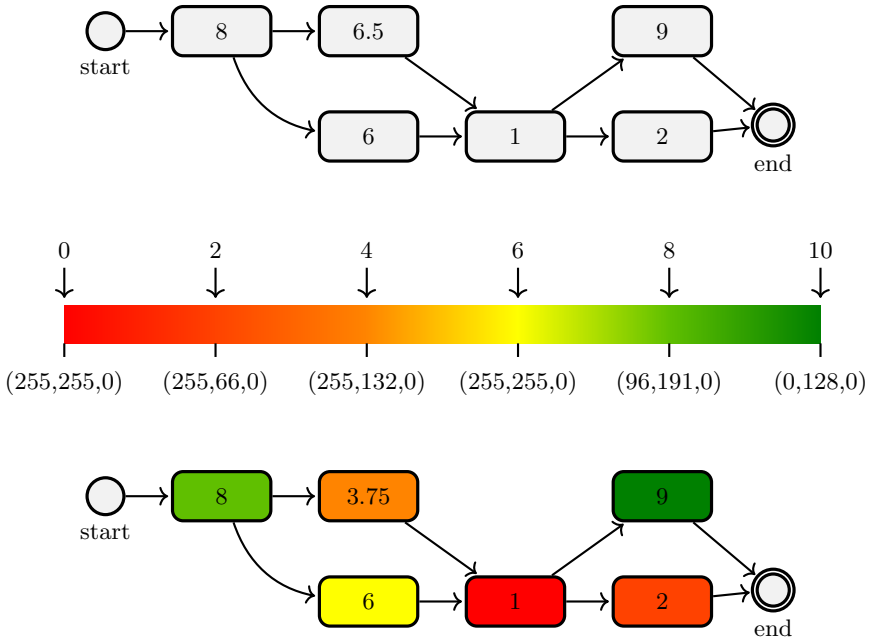
A *color* $c \in C$ is a triplet (r, g, b) where $0 \leq r \leq 255$, $0 \leq g \leq 255$, $0 \leq b \leq 255$ (HIRSCH, 2004).
A *color map* Cm is a function $Cm : Vdom_f \subseteq dom_F \rightarrow C$ where dom_F is the factor F domain, and C is a set of colors.

Figure 5.4 – Factor-based color enhancement component



For instance, Figure 5.5 presents a certain process model with a factor F aggregated by activities. Some values ($Vdom_f = \{0, 2, 4, 6, 8, 10\}$) are mapped to specific colors $(255, 255, 0), (255, 66, 0), (255, 255, 0), (96, 191, 0), (0, 128, 0)$. The bottom model presents the same model enhanced by colors based on factor F.

Figure 5.5 – Example of factor-based color enhancement



5.2.1 Interface

The color enhancement component creates from a given event log a colored DFG based on factor information. The input parameters (•) and outputs (–) are:

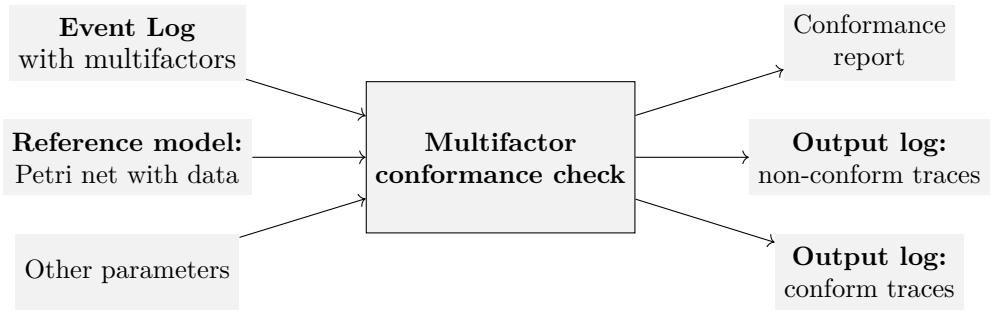
- **Event log** with multifactor information.
- **Factor** in the event log to be used for coloring.
- **Aggregation function** to aggregate factor values by activity. One of the following: sum, max, min, and average.
- Ordered **list of factor values**. This list will define the ranges in the color map. Each value from the list is mapped to a color. The intermediary values are interpolated to intermediary colors.
- **List of colors** to which the factor values are mapped.

– **Factor-based colored DFG**.

5.3 Multifactor conformance check

The conformance component, illustrated in Figure 5.6, is responsible for creating a conformance report. It also can split the input log into the conform log and non-conform log.

Figure 5.6 – Multifactor conformance check component



5.3.1 Interface

The input parameters (•) and outputs (–) are:

- The **event log** with multifactor information defined in an XES file.
- **Model**. The petri net with data used as reference model.
- **Method (Default: "alignment")**. The method to check structural conformance. The options are "token" for token-based replay, and "alignment".

– **By-trace diagnostic**. For each trace the following information is returned depending on the selected method:

  – **Activated transitions**. List of transitions activated in the model (in case of method="token").

  – **Reached marking**. Marking reached at the end of the replay (in case of method="token").

  – **Missing tokens**. The number of missing tokens (in case of method="token").

  – **Consumed tokens**. The number of consumed tokens (in case of method="token").

  – **Remaining tokens**. The number of remaining tokens (in case of method="token").

  – **Produced tokens**. The number of produced tokens (in case of method="token").

  – **Cost**. Cost of the alignment (in case of method="alignment").

  – **Fitness**. The fitness according to the structural method.

  – **Alignment**. The alignment operations: sync moves, moves on log, and moves on the model (in case of method="alignment").

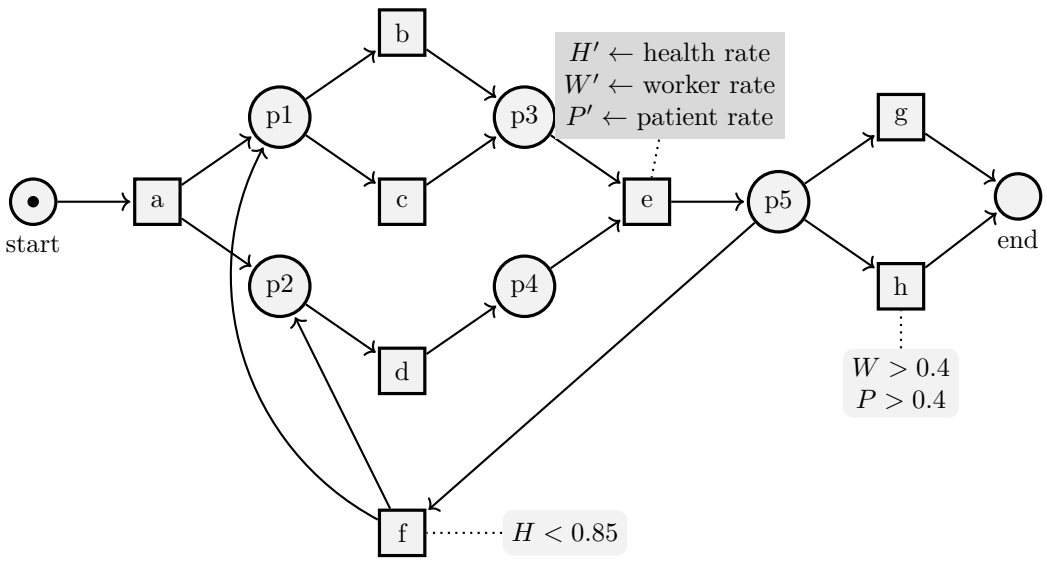
  – **Guards violated**. List of guards that were violated in the model.

- **Multifactor diagnostic**. Conformance report for the whole log analyzing the multifactors, including for each factor:
  - **Factor fitness**. Percentage of traces that don't violate any guard that includes the corresponding factor.
  - **Violations**. Number of times in which the value of the factor causes a violation of a guard.
- **Output log**. Several output logs can be selected to be exported from the conformance component:
  - **Violating structural log**. Event log containing the traces that have fitness different than 1.
  - **Non-violating structural log**. Event log containing the traces that have fitness equal to 1.
  - **Violating factor log**. Event log containing the traces that violated at least one of the guards for the corresponding factor.
  - **Non-violating factor log**. Event log containing the traces that do not violate any of the guards for the corresponding factor.

5.3.2 Example

For example, the log presented in Frame 5.1 can be given as input to this component, along with the reference model in Figure 5.7. The log contains three illustrative quality

Figure 5.7 – Example of reference model for conformance checking with multifactor information



Frame 5.1 – A fragment of some healthcare event log with cost and three quality indicators

Case id	Event id	Properties								
		Timestamp	Activity	Label	Resource	Cost	W	H	P	...
1	423	30-12-2010:11.02	admission	*a*	Pete	50	0.9	1.0	0.8	...
	424	31-12-2010:10.06	examine thoroughly	*b*	Sue	999	0.6	1.0	0.9	...
	425	05-01-2011:15.12	check status	*d*	Mike	100	0.5	1.0	0.8	...
	426	06-01-2011:11.18	operate	*e*	Sara	200	0.9	0.9	1.0	...
	427	07-01-2011:14.24	consultation	*h*	Pete	200	0.9	0.8	0.8	...
2	483	30-12-2010:11.32	admission	*a*	Mike	50	0.9	1.0	0.9	...
	485	30-12-2010:12.12	check status	*d*	Mike	100	0.5	1.0	0.9	...
	487	30-12-2010:14.16	examine casually	*c*	Pete	400	0.9	0.5	0.8	...
	488	05-01-2011:11.22	operate	*e*	Sara	200	0.9	0.9	1.0	...
	489	08-01-2011:12.05	after care	*g*	Ellen	200	0.9	0.9	0.9	...
3	521	30-12-2010:14.32	admission	*a*	Pete	50	0.9	1.0	0.8	...
	522	30-12-2010:15.06	examine casually	*c*	Mike	400	0.9	0.5	0.9	...
	524	30-12-2010:16.34	check status	*d*	Ellen	100	0.5	1.0	0.8	...
	525	06-01-2011:09.18	operate	*e*	Sara	200	0.9	0.9	1.0	...
	526	06-01-2011:12.18	examine complications	*f*	Sara	700	0.9	1.0	0.9	...
	527	06-01-2011:13.06	examine thoroughly	*b*	Sean	999	0.6	1.0	0.9	...
	530	08-01-2011:11.43	check status	*d*	Pete	100	0.5	1.0	0.7	...
	531	09-01-2011:09.55	operate	*e*	Sara	200	0.9	0.9	0.7	...
	533	15-01-2011:10.45	after care	*g*	Ellen	200	0.9	0.9	0.9	...
4	641	06-01-2011:15.02	admission	*a*	Pete	50	0.9	1.0	0.8	...
	643	07-01-2011:12.06	check status	*d*	Mike	100	0.5	1.0	0.6	...
	644	08-01-2011:14.43	examine thoroughly	*b*	Sean	999	0.6	0.9	0.9	...
	645	09-01-2011:12.02	operate	*e*	Sara	200	0.9	0.9	0.9	...
	647	12-01-2011:15.44	consultation	*h*	Ellen	200	0.9	0.8	0.8	...
5	711	06-01-2011:09.02	admission	*a*	Ellen	50	0.9	1.0	0.9	...
	712	07-01-2011:10.16	examine casually	*c*	Mike	400	0.9	0.5	1.0	...
	714	08-01-2011:11.22	check status	*d*	Pete	100	0.5	1.0	0.6	...
	715	10-01-2011:13.28	operate	*e*	Sara	200	0.9	0.9	1.0	...
	716	11-01-2011:16.18	examine complications	*f*	Sara	700	0.9	1.0	0.9	...
	718	14-01-2011:14.33	check status	*d*	Ellen	100	0.5	1.0	0.5	...
	719	16-01-2011:15.50	examine casually	*c*	Mike	400	0.9	0.5	0.9	...
	720	19-01-2011:11.18	operate	*e*	Sara	200	0.9	0.9	1.0	...
	721	20-01-2011:12.48	examine complications	*f*	Sara	700	0.9	1.0	0.9	...
	722	21-01-2011:09.06	examine casually	*c*	Sue	400	0.9	0.5	0.9	...
	724	21-01-2011:11.34	check status	*d*	Pete	100	0.5	1.0	0.7	...
	725	23-01-2011:13.12	operate	*e*	Sara	200	0.9	0.9	0.7	...
	726	24-01-2011:14.56	consultation	*h*	Mike	200	0.9	0.8	0.7	...
6	871	06-01-2011:15.02	admission	*a*	Mike	50	0.9	1.0	0.9	...
	873	06-01-2011:16.06	examine casually	*c*	Ellen	400	0.9	0.5	0.9	...
	874	07-01-2011:16.22	check status	*d*	Mike	100	0.5	1.0	0.9	...
	875	07-01-2011:16.52	operate	*e*	Sara	200	0.9	0.9	1.0	...
	877	16-01-2011:11.47	after care	*g*	Mike	200	0.9	0.9	0.9	...

factors: the *health rate* H, the *worker rate* W, and the *patient rate* P. The selected method is token-based replay. In the reference model, H, P, and W of activity e are captured ($H' \leftarrow$ health rate, $W' \leftarrow$ worker rate, $P' \leftarrow$ patient rate), and activity f can only be performed in case this health rate is less than 0.85 ($H < 0.85$). Also, activity h can only be performed if W and P are greater than 0.4.

Frame 5.2 shows the by-trace diagnostic for the example. It is possible to observe that all traces reached the *end* marking. Also, the structural fitness (Section 2.4) for all traces is 1.0; that is, all produced tokens were consumed, and no missing or remaining tokens were detected. However, traces 3 and 5 violated the guard on the reference model. Activity f should be executed only when the health rate from activity e is smaller than 0.85. That is not the case for those traces, where the health rate from activity e is 0.9.

Furthermore, trace 5 violated the guard twice.

Frame 5.2 – Example of by-trace diagnostic generated by the conformance component

Trace	Activated transitions	Reached marking	Missing tokens	Consumed tokens	Remaining tokens	Produced tokens	Structural fitness	Guards violated
1	abdeh	end	0	7	0	7	1.0	
2	adceg	end	0	7	0	7	1.0	
3	acdefbdeg	end	0	12	0	12	1.0	$H < 0.85$
4	adbeh	end	0	7	0	7	1.0	
5	acdefdcefcdeh	end	0	17	0	17	1.0	$H < 0.85$ $H < 0.85$
6	acdeg	end	0	7	0	7	1.0	

Frame 5.3 – Example of multifactor diagnostic generated by the conformance component

Factor	Factor fitness	Violations
cost	1.0	0
H	0.6	3
W	1.0	0
P	1.0	0

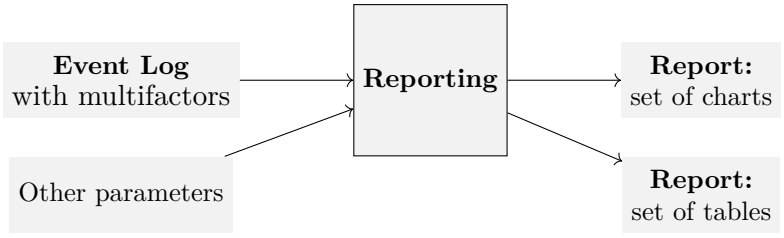
Additionally, the multifactor diagnostic is generated for the example and presented in Frame 5.3. It identifies that factor health rate was involved in guard violations 3 times, in one third of the log.

The output logs can be used for further analysis. The violating output logs can be used to estimated the avoidable cost, in the same way as presented in Section 3.4.1. For example, one can output the *Violating factor log* and use it as input to the *reporting* and *organizational* components. In this case, the charts and tables from theses components could identify *who was involved in the violating traces*, *how much the violating traces costed*, and *when the violating traces occurred*, for example.

5.4 Reporting

The reporting component is responsible for creating a report based on a given event log with multifactor information, as shown in Figure 5.8. The report contains charts and tables according to the user's specifications.

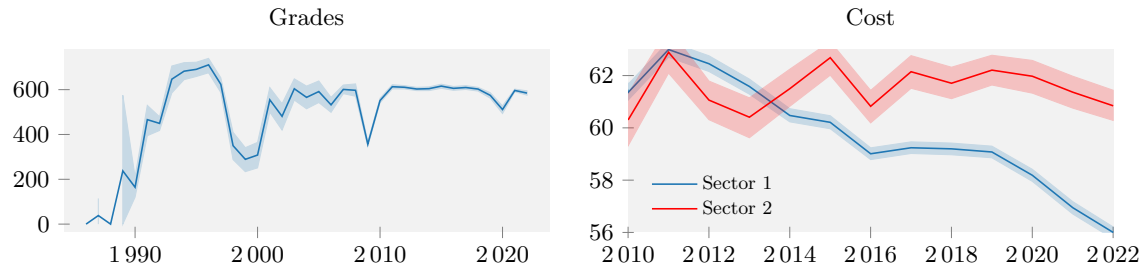
Figure 5.8 – Reporting component



The following reports are supported. The data table of each chart can also be exported for further analysis. Examples are shown for illustration purposes:

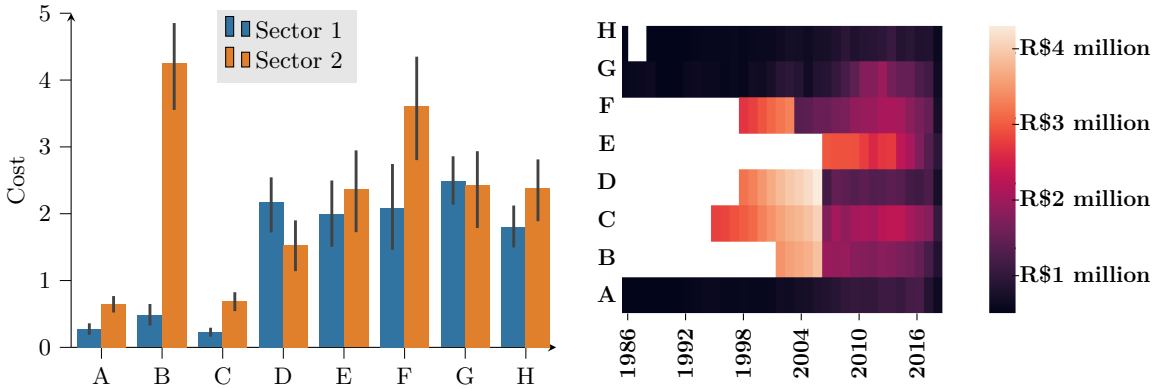
- **Factor vs. numerical:** one factor can be chosen to be plotted vs. a numerical variable, such as time. The numerical variable can be split into bins and aggregated by some aggregation function (max, min, or average). The factor can be plotted by factor elements or another descriptive column. For example, Figure 5.9 shows factors *grades* (left) and *cost* (right) plotted by time. Each line shows the mean (hard line) and 95% confidence interval (the soft area around the mean).

Figure 5.9 – Factor vs. numerical report example



- **Factor vs. categorical:** it plots one factor vs. one or two categorical variables. It can be used to plot a factor by activity, resource, case, or other categorical variables. Aggregations can be used. The chart can be a *heatmap* or *columns*. Figure 5.10 (left) shows the average cost factor by activity and sector using the column option and the total cost factor by activity and year (right) using the heatmap option.

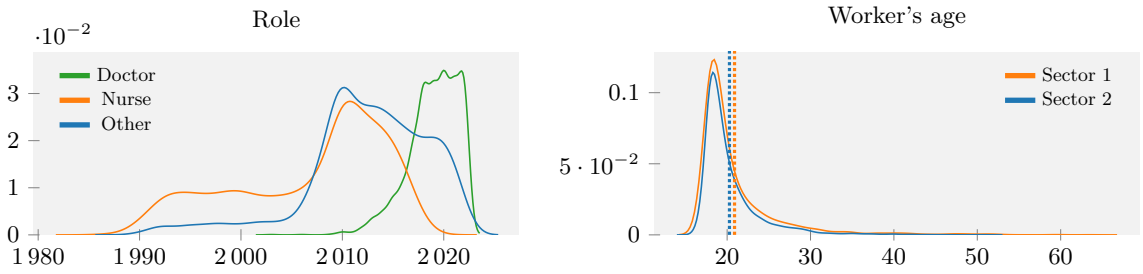
Figure 5.10 – Factor vs. categorical report example - column (left) and heatmap (right)



- **Histograms:** it plots a Kernel Density Estimate (KDE) histogram (HÄRDLE et al., 2004) of events or case attributes, including factors. Figure 5.11 shows the histogram (density) of workers of a certain company. On the left, the histogram shows the workers by role over the years; on the right, the workers by age and sector. The mean of each distribution can be plotted too (dashed lines).

- **Statistical analysis:** it exports a descriptive analysis of the dataset (Frame 5.4). Case and event attributes are summarized with statistical data: for numerical

Figure 5.11 – Histogram report example



variables (or factors), maximum, minimum, median, average, and standard deviation; for categorical variables, the proportion and the total number of each category. A categorical variable can be set as the outcome. In that case, the correlation between variables and the outcome is measured (Student's t-test (BONEAU, 1960) and chi-square test (CORDER; FOREMAN, 2014)), and a correlation table (Frame 5.5) is generated. Finally, an entropy test (MAIMON; ROKACH, 2014) to measure each correlated variable's information gain with respect to the defined outcome can be conducted (Figure 5.12). Next, as an example, a log containing patients under certain treatments was used. Frame 5.4 describes the variables: age, number of applications, gender, prescription, and outcome. The correlation between all variables and outcome is measured in Frame 5.5, indicating that all variables correlate to it ($p < .001$). Figure 5.12 shows the Information Gain (IG) from the entropy test, indicating that variable *prescription* has a higher IG.

Frame 5.4 – Example of descriptive analysis table.

Variable	Description				
Numerical	Max	Min	Median	Average	Std
Age	60	17	20.4	20.5	4.12
Applications	13	5	10	12.67	1.07
Categorical		Total		Proportion	
Gender					
Male		7,033		72.64%	
Female		2,648		27.36%	
Prescription					
Drug A		3,215		33.21%	
Drug B		3,024		31.22%	
Drug C		3,442		35.55%	
Outcome					
Successful		5,415		55.93%	
Unsuccessful		4,266		44.07%	

5.4.1 Interface

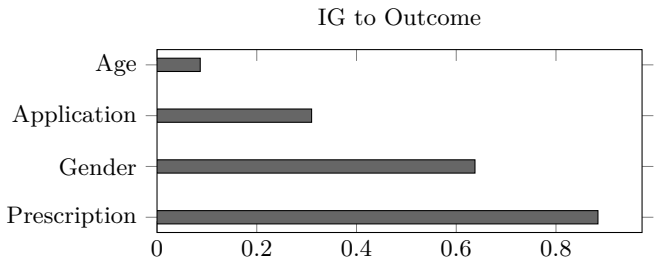
The input parameters (•) and outputs (–) are:

- **Event log**. The event log with multifactor information defined in a XES file.

Frame 5.5 – Example of correlation table.

Variable	Outcome Successful (5,415)	Outcome Unsuccessful (4,266)	P
Age	20.89 (5.14)	20.28 (3.60)	<.001
Gender			<.001
Male	3,497 (64.58%)	3,536 (82.89%)	
Female	1,918 (35.42%)	730 (17.11%)	
Prescription			<.001
Drug A	1,012 (18.69%)	2,203 (51.64%)	
Drug B	2,017 (37.25%)	1,007 (23.61%)	
Drug C	2,386 (44.06%)	1,056 (24.75%)	
Applications	12.10 (1.10)	5.10 (2.00)	<.001

Figure 5.12 – Example of Information Gain of each variable with respect to the defined outcome.



- **Report**. The user can choose one or more of the available reports: Factor vs. numerical, Factor vs. categorical, Histogram, and Statistical analysis.
- **Configuration**. Each report has a set of configurations to be defined:
  - Factor vs. numerical:
    - **Factor** to be included in the chart.
    - **Numerical column** to be included in the chart.
    - **Aggregation function** to aggregate factor values.
    - **Descriptive column (optional)** to be used for different factor types.
    - **Bins (optional)** to be used for the numerical variable.
  - Factor vs. categorical:
    - **Factor**, **Aggregation function**, and **Descriptive column** as in Factor vs. numerical.
    - **Categorical column** to be included in the chart.
    - **Type (Default: columns)** of the chart: it can be *columns* or *heatmap*.
  - Histogram:
    - **Factor** to be included in the chart. Another numerical variable can also be used.
    - **Descriptive column (optional)** to be used for different histograms.
    - **Type (Default: trace)** of the histogram - what will be used to count: "trace" or "event".

- Statistical analysis:
  - **Outcome (optional)** to be used for correlation and IG.

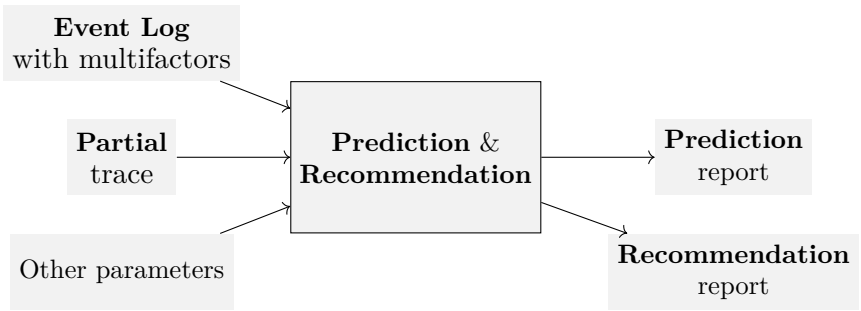
– PNG **Image** file generated for the chart.

– CSV text file **table** generated for each chart.

5.5 Prediction/Recommendation

The prediction/recommendation component is responsible for predicting a resulting variable for a partial trace. Its input is an event log to be used as historical data, and the prediction is made by statistical analysis over the log. The method used for prediction is stat-annotated event log (AALST; SCHONENBERG; SONG, 2011) presented in Section 2.6.2. It is possible to set a *goal*, and based on the prediction, a recommendation for the next activity is made.

Figure 5.13 – Prediction and Recommendation component



5.5.1 Interface

The input parameters (•) and outputs (–) are:

- The **event log** with multifactor information defined in an XES file. The event log is used as historical data to create an annotated transition system as prediction/recommendation structure.
- A **partial trace** to be used in the prediction/recommendation.
- **Task** to be performed by the component. One or more of the following can be selected:
  - **"end of trace prediction"**. It predicts the value of each factor, or time, for the end state. An aggregation function can be used to predict the max, min, average, or sum of the factor.

- **"next-state prediction"**. It predicts the value of each factor, or time, for the next state. An aggregation can be used.
- **"recommendation"**. It recommends the next activity to be performed based on *goal*.

- **Goal** configuration for recommendation task:
  - **Goal**. The goal for the recommendation. It can be "minimize" or "maximize".
  - **Goal variable**. The variable to be used for the recommendation.
  - **Aggregation function** to be applied to the variable.

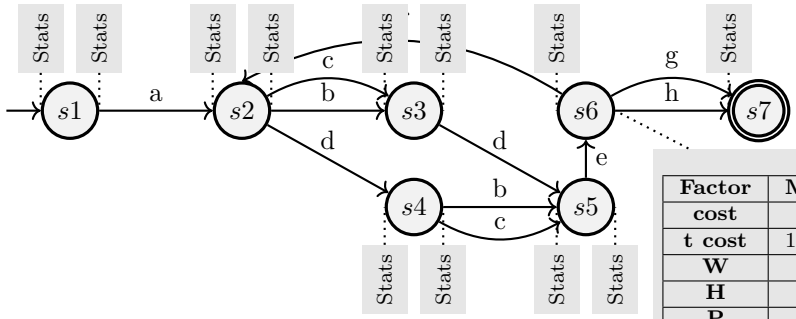
– **Prediction report**. A prediction based on the options set.

– **Recommendation report**. A recommendation report detailing the possible next states and the best outcome according to the set options.

5.5.2 Example

Considering the log presented in Frame 5.1 as an example, it is possible to illustrate the *prediction report* and the *recommendation report*. In this case, the partial trace used is $\langle a, b, d, e \rangle$. First, the component generates a transition system with statistical information about time and multifactor based on the event log. Figure 5.14 illustrates a multifactor-annotated transition system.

Figure 5.14 – Multifactor-annotated transition system for prediction



Stats:

Factor	Mean	Max	Min	Count	St Dev
cost	200	200	200	6	0
t cost	1749.5	750	3550	6	1145.39
W	0.77	0.9	0.5	6	0.18
H	0.88	1	0.5	6	0.18
P	0.85	1	0.6	6	012
e	10	8	12	6	1.6
r	10	7	13	6	1.8
s	10	6	14	6	1.9

The transition system is annotated with information about time, like in Section 2.6, and the multifactors (cost, W, H, and P). Also, the total cost factor is recorded. Mean, minimum, maximum, quantity, and standard deviation are recorded for each factor in each state. For clarity purposes, only one note in the figure is shown in detail.

The component is configured to *end of trace prediction*, to predict the total (aggregation function sum) of time and cost, and to predict the average final value of all factors. An example of the *prediction report* is shown in Frame 5.6. Also, the component is set to *recommendation*. The goal is to minimize the total cost (goal variable = "cost", goal = "minimize", aggregation = "sum"). An illustrative example of the *recommendation report* is shown in Frame 5.7.

Frame 5.6 – Example of multifactor prediction report

Factor	Predicted value	Certainty
cost	309.8	0.96
H	0.94	0.96
W	0.76	0.75
P	0.86	0.82
total cost	1549.10	0.96
total time	100	0.97

Frame 5.7 – Example of multifactor recommendation report

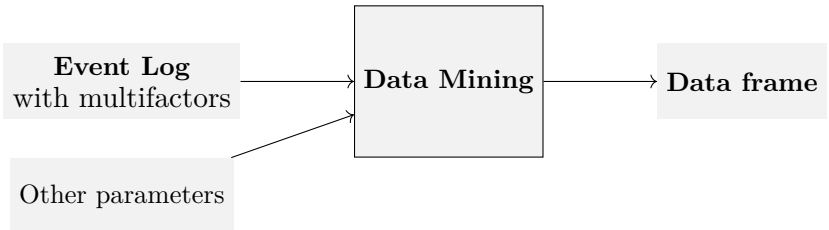
Possible next activities	Predicted values	Best outcome
g	1549.10	True
h	1560.76	False

In Frame 5.6, it is possible to observe the predicted value for each factor and the degree of certainty associated with it. Frame 5.7 shows the next possible activities (h and g) and the predicted total value for each state. Since the goal is to minimize the total cost value, the best outcome is activity g.

5.6 Data Mining

The Data Mining component creates a data frame representation of the event log. This representation can be useful for data mining activities such as classification or clustering. The component can represent case and event attributes, including factors, in the data frame information. As shown in Figure 5.15, the input is the event log to be represented as data frame.

Figure 5.15 – Data Mining component



5.6.1 Interface

The input parameters (•) and outputs (–) are:

- The **event log** with multifactor information to be represented as a data frame.
- **List of representations** to be included on the data frame. One or more of the following can be selected:
  - **"case attributes"** to be included in the representation, including factors. Categorical attributes are encoded into a numerical representation.
  - **"event frequency"**. Includes the number of occurrences of each activity as a case attribute.
  - **"transition frequency"**. Includes the number of occurrences of each transition as a case attribute.
  - **"event attributes"** to be included in the representation, including factors. An aggregation function needs to be used to represent event-level information at the trace level.

– **Data frame:** each row represents a case. Each column contains a representation.

5.6.2 Example

Consider the event log in Figure 5.1. A data frame representation, shown in the Frame 5.8, can be calculated using the data mining component.

Frame 5.8 – Example of data frame

	Activity								Sum	Average			Resource					
Case	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**Cost**	**W**	**H**	**P**	**Pete**	**Mike**	**Sue**	**Sara**	**Ellen**	**Sean**
1	1	1	0	1	1	0	0	1	1549	0.76	0.94	0.86	0	1	2	1	0	1
2	1	0	1	1	1	0	1	0	950	0.82	0.86	0.90	1	2	1	1	0	0
3	1	1	1	2	2	1	1	0	2949	0.78	0.91	0.84	2	1	2	3	1	0
4	1	1	0	1	1	0	0	1	1549	0.76	0.92	0.80	1	1	1	1	1	0
5	1	0	3	3	3	2	0	1	3750	0.81	0.85	0.82	2	3	2	5	0	1
6	1	0	1	1	1	0	1	0	950	0.82	0.86	0.92	1	3	0	1	0	0

In this example, the following representations were included:

- "event frequency": for each case, the number of occurrences of activities (a to h) is used.
- "event attributes": the total *cost* of each case (sum of cost factor) and the average of W, H, and P are used. Also, the number of occurrences of each resource in each case is used.

Part III

Case studies

This part shows the case studies performed intending to validate the Multifactor Process Mining Framework and the results obtained from them. In the *method* section of each chapter (Sections 6.2, 7.1.1, and 8.1), it is presented a description of what components from the framework were used in the study.

■ Chapter 6 explores the finding of a case study on a real-world Brazilian telecommunication company that provides telephony, television, and broadband Internet subscription services. It explores the association between Process Mining and Root Cause Analysis (RCA) to identify the possible causes for factors such as short and long-duration services, high rework rates, and activity repetition.

■ Chapter 7 presents a case study on an educational dataset from a Brazilian public university. The aim was to understand how programs, courses, and students interact and what statistically significant and meaningful patterns lead students to different paths and outcomes (dropouts vs. graduates, long vs. short graduation time, and high vs. low grades). We compared students' outcomes and paths, considering enrollment and course variables along with students' demographic information. We performed analysis using cost, grades, and attendance factors.

■ Chapter 8 presents an ongoing case study on the healthcare domain. The study explores a surgery center in a Brazilian hospital. The duration, frequency, and revenue of surgery rooms were analyzed.

6 Telecommunication

Process Mining can be associated with Root Cause Analysis (RCA), a problem-solving technique based on the assumption that a problem can only be solved by addressing its underlying cause (HERAVIZADEH; MENDLING; ROSEMANN, 2008). In a process, one may want to find the cause for high duration, high cost, high rework rate, or any other defined outcome. With PM, it is also possible to analyze how data attributes influence the choices along the process workflow (ROZINAT; AALST, 2006).

Several works have been proposed regarding the use of PM for RCA. RCA sought to understand the correlation between resources' workload and performance using PM, and linear regression (NAKATUMBA; AALST, 2009). Another study uses RCA to explain the root cause of the occurrence of idle times in a manufacturing test process (ROZINAT et al., 2009a). In (AGUIRRE; PARRA; ALVARADO, 2012), RCA was used to identify the major delay causes in a specific purchase requisition approval. The main detected delay cause is that the process has physical documents that are not handled in a central repository.

In (VASILYEV; FERREIRA; IIJIMA, 2013), the authors try to find the reasons for delays in a certain business process. Delays can have adverse outcomes for organizations, such as extra costs, poor service, missed deadlines, etc. The study proposes an approach that uses a decision tree to split process instances according to their duration. By following the path in the tree, it is possible to provide an explanation for the delay. Validation was performed in a set of synthetic logs. In (LEHTO; HINKKA; HOLLMÉN, 2016), an RCA approach is proposed using classification rule mining. The cases are separated into either problematic or successful. Measures for reporting the results to business people are defined. Two real-life case studies are performed for validation.

In the study (QAFARI; AALST, 2020), the authors present a way to find not only the features that cause a certain problem but also the effect of an intervention on any of the features. For that, they used causal equation models for processes. They have implemented this method and evaluated it using a real and a synthetic log. Another study proposes an approach to measure cause-effect relations in event logs (HOUDT; DEPAIRE; MARTIN, 2022). The approach uses probabilistic temporal logic to define and test hypotheses for causal relations from data. The approach was validated in a real-world event log. The study (QAFARI; AALST, 2022) proposes a technique for finding the structural equation model of the process that can be used for causal analysis, i.e., a method for discovering the set of features with a possible causal effect with respect to a certain problem. The technique was also evaluated using real and synthetic event logs.

These related studies reaffirm the applicability and importance of PM and RCA to help organizations to understand why their business process instances have certain outcomes. In light of that, we performed a case study on a real-world Brazilian telecommunication company that provides telephony, television, and broadband Internet subscription services. In this case study, we use the company event log along with PM and RCA to identify the possible causes for services normal and long duration, high rework rate, and activity repetition. For that, we applied both decision tree classification (VASILYEV; FERREIRA; IIJIMA, 2013) and rule mining (LEHTO; HINKKA; HOLLMÉN, 2016).

In Section 6.1, we explain how RCA can be performed using decision tree classification (VASILYEV; FERREIRA; IIJIMA, 2013) and rule mining (LEHTO; HINKKA; HOLLMÉN, 2016). In Section 6.2, we describe the dataset and the used method. Finally, in Section 6.3, we present and discuss the results.

6.1 Background

To perform RCA, every case in the event log should be mapped into a vector representation suitable to decision tree classification (VASILYEV; FERREIRA; IIJIMA, 2013) and rule mining (LEHTO; HINKKA; HOLLMÉN, 2016). Each vector represents a case, and relevant event-level attributes have to be represented as case-level attributes, i.e., for a relevant event attribute in the log $\#_\sigma(L)$ to be included in the analysis, it should exist a function f such that $f(\#_\sigma(L)) = \#_\gamma(L)$.

If we perform decision tree classification using entropy criterion, a tree structure can be obtained (as described in Section 2.5.3). In the tree, the paths from the root to leafs represent classification rules. The tree also offers insights into which attributes contribute more to the defined outcome. We can use the obtained rules in the tree in the format of association rules $A \rightarrow B$, where A is the antecedent and B the consequent (LEHTO; HINKKA; HOLLMÉN, 2016): the occurrence of A is associated with the occurrence of B. Three metrics can be calculated for each rule (LAROSE; LAROSE, 2014). *Support* ($S_\rightarrow$) shows how frequently the rule $A \rightarrow B$ appears in the data. *Confidence* ($C_\rightarrow$) is the percentage of all cases satisfying A that also satisfy B. *Lift* ($L_\rightarrow$) is the ratio of the calculated support to that expected if A and B were independent. If $L_\rightarrow = 1$, then A is independent of B. If $L_\rightarrow > 1$, then $L_\rightarrow$ is the degree of dependence of A and B. If $L_\rightarrow < 1$, then the occurrence of A has a negative effect on the occurrence of B and vice versa. Formally (LAROSE; LAROSE, 2014):

$$support(A) = \frac{\text{number of records containing A}}{\text{total number of records}}$$

$$support(A \rightarrow B) = \frac{\text{number of records containing A and B}}{\text{total number of records}}$$

$$confidence(A \rightarrow B) = \frac{\text{number of records containing A and B}}{\text{number of records containing A}}$$

$$lift(A \rightarrow B) = \frac{support(A \rightarrow B)}{support(A) \times support(B)}$$

By combining decision tree learning and associating rule metrics, it is possible to have a set of rules that may explain the root cause for a defined class of cases in an event log. It is also possible to rank attributes by their importance. In the next section, we explain the details of the application of this approach in the real-world case study.

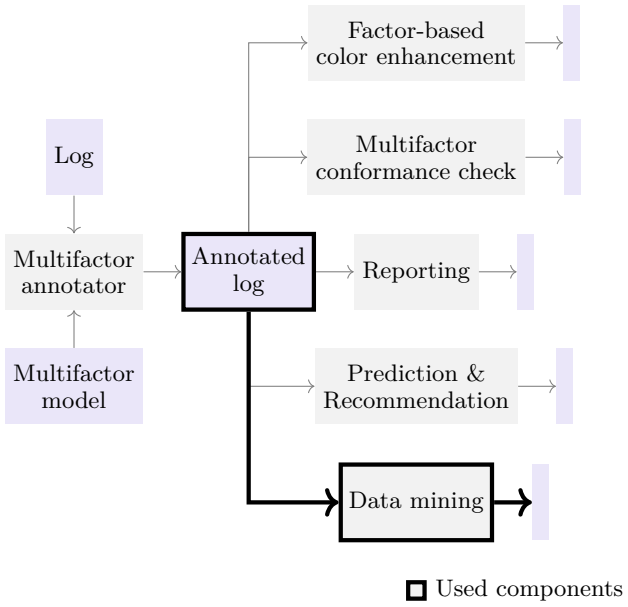
6.2 Method

By using the approach mentioned above, we performed root cause analysis in a real-world case study. The case study is performed over an event log of a Brazilian tech company containing 766.892 events recorded, with 50 attributes, in 90.436 cases. The log was made available anonymized in a secure environment by Upflux (UPFLUX, 2023) in the context of the DAI project.

The log contains events from July to September 2022 in three categories of services: Support and Maintenance (360.601 events), Sales and Installation (269.256 events), and Cancellation (79.085 events). 47.715 events were eliminated in the preprocessing phase. We analyzed three aspects for each category of services: duration, rework rate, and activity repetition.

We considered the following categorical attributes at the case level when transforming the event log into classification format: customer type (natural or legal person), customer city name, customer service channel type (phone, chatbot, Whatsapp-Wpp, etc.), service name (svc.), type of protocol, support level, selling unit, day of the week, and week of the month. At the event level, we consider the following categorical attributes: working group, sector, worker name, and if the worker is a bot or not. Also, as numerical attributes, we consider the number of occurrences of each activity (act.). As depicted in Figure 6.1, we used **data mining** component from the multifactor framework to create the vector representation for each case.

Figure 6.1 – Framework's used components in the telecommunication case study



When splitting cases on the duration for each category of services, we considered "long duration" cases lasting more than the average duration of all cases in that category plus one standard deviation. This corresponds to 9 days for Cancellation, four days for Support, and 16 days for Installation. For the rework rate, we considered "high rework" cases in which, in 30 days or less, the same customer appears in more than one case soliciting the same service. For activity repetition, we consider only the repetitions of activity "protocol transfer", and, as cases with "high repetition", we consider cases with two or more occurrences of this activity. We performed the experiments using unbalanced decision trees, with three levels at most.

6.3 Results

Frames 6.1, 6.2,and 6.3 show the discovered rules for the Support, Cancellation, and Sales datasets, respectively. All calculated lifts indicate a dependence of A and B ($L_{\rightarrow} > 1$). Some rules have low support/confidence, especially the ones with long/high classes. Since the classes were unbalanced, this was expected. Process specialists validated all the rules. They reported that some rules were known patterns inside the process, while others will take further investigation. Next, these patterns are detailed.

Regarding the Support dataset (Frame 6.1):

- The occurrence of activity "protocol transfer" causes a long-duration case. This is because the first support worker cannot resolve the service and has to transfer the protocol to another worker, causing the case to last longer. If activity "standard progress" occurs and the support level is 1, the cases have normal duration. Support

level different than one is assigned to more complex cases that will take longer to be solved. The occurrence of sector 1 also takes the case to a long duration. Sector 1 is responsible for field support, which takes longer to be solved due to logistics.

- Worker 422 is a bot worker, and when the service is not "slow connection", it resolves the case without rework. Support for "slow connection" problem tend to have rework because of the nature of the service: the slow connection problem persists even after the protocol end, so the customer asks for support again. When worker 422 is not involved, and the customer is a natural person, the service also tends to have rework, because legal people normally are treated with priority by support. Regarding legal people, when there is an IVR (Interactive voice response) protocol, the case tends to be rework as well. This is because IVR normally suggests some steps to solve the problem; the customer ends the call to try to follow the steps and sometimes has to ask for support again.

- Sector 1 also takes the case to have a high repetition rate of "protocol transfer" activity. Group 30, from Sector 1, tends not to cause high repetition. When Sector 1 is not involved, the repetition occurs when activities "customer response" or "standard progress" occur.

Frame 6.1 – Discovered rules for support dataset

	Antecedent (A)			Class (B)	Support	Confidence	Lift
Duration	No act. protocol transfer	No act. standard progress	No sector 1	Normal	0.6993	0.9984	1.0564
			Sector 1	Long	0.0004	0.0645	1.1758
		Act. standard progress	No support level 1	Long	0.0009	0.1796	3.2728
			Support level 1	Normal	0.0044	0.9717	1.0281
	Act. protocol transfer			Long	0.0523	0.1844	3.3610
Rework	No worker 422	Natural person		High	0.0047	0.0890	6.8083
		Legal person	No IVR protocol	Normal	0.2223	0.9931	1.0063
			IVR protocol	High	0.0029	0.0374	2.8623
	Worker 422	No svc. slow connection		Normal	0.5686	0.9950	1.0082
		Svc. slow connection		High	0.0010	0.0143	1.0936
Act. Repetition	No sector 1	No act. standard progress	No act. costumer response	Normal	0.7066	0.9990	1.1020
			Act. costumer response	High	0.0010	0.1761	1.8836
		Act. standard progress	No act. protocol taking	Normal	0.0174	0.9275	1.0231
			Act. protocol taking	High	0.0045	0.3190	3.4122
	Sector 1	No act. protocol taking	No group 30	High	0.0635	0.2860	3.0590
			Group 30	Normal	0.0041	1.0000	1.1031
		Act. protocol taking		Long	0.0223	0.8066	8.6281

With respect to Cancellation dataset (Frame 6.2):

- In the same way as the Support dataset, in the Cancellation dataset, we found that the activity "protocol transfer" occurrence and Sector 1 (field support) cause a long-duration case. The retention group, which tries not to lose the customer, also takes the case to a long duration. We found that worker 87 when using Whatsapp for customer service, takes a long time. Worker 87 has other tasks to be performed

and leaves Whatsapp open while performing them, which could be the explanation for the delay.

- When analyzing rework in the Cancellation dataset, we found that City S is decisive: when cancellation service is not performed in City S, it would not be reworked (99% of confidence). Further exploration needs to be done to understand this aspect. Also, if the service "cancellation info" does not occur, normally there is no rework. This shows a pattern of customers contacting the cancellation team several times regarding "cancellation info". Bot users tend not to end up in rework.

- Concerning the repetition of activity "protocol transfer", we observe that activity "visit change", group "equipment pick-up", and group "retention" are associated with high duration. This can be explained because these three actions are related to the change of workers, so repetition of the activity "protocol transfer" is expected. On the other hand, further investigation needs to be conducted to understand why Group 28 tends to have high repetition and Group 24 tends to do the opposite.

Frame 6.2 – Discovered rules for cancellation dataset

	Antecedent (A)			**Class (B)**	**Support**	**Confidence**	Lift
Duration	No act. protocol transfer	No worker 87		Normal	0.6522	0.9960	1.0894
		Worker 87	No Wpp svc. channel	Normal	0.0004	1.0000	1.0937
			Wpp svc. channel	Long	0.0011	0.7692	8.9753
	Act. protocol transfer	No sector 1	No group retention	Normal	0.0709	0.9805	1.0724
			Group retention	Long	0.0051	0.1281	1.4943
		Sector 1		Long	0.0755	0.3266	3.8107
Rework	No city S			Normal	0.9795	0.9983	1.0037
	City S	No svc. cancellation info		Normal	0.0080	1.0000	1.0054
		Svc. cancellation info	No bot	High	0.0037	0.4198	78.8622
			Bot	Normal	0.0020	1.0000	1.0054
Act. Repetition	No act. visit change	No group retention	No group equip. pick-up	Normal	0.7381	0.9971	1.1540
			Group equip. pick-up	High	0.0025	0.2840	2.0879
		Group retention	No group 28	High	0.0200	0.5444	4.0028
			Group 28	Normal	0.0229	0.9724	1.1254
	Act. visit change	No group equip. pick-up		High	0.0328	0.3005	2.2096
		Group equip. pick-up	No group 24	High	0.0779	0.9663	7.1053
			Group 24	Normal	0.0009	1.0000	1.1574

Regarding the Sales dataset (Frame 6.3):

- With respect to duration, for customers with no support level, long duration tends to occur on Mondays. Sale requisitions from across the weekend tend to accumulate and, on Monday, start to be resolved, causing long duration. Sales to customers with some support level tend to be longer when workers 95 and 587 are involved. This is another point to be investigated.

- City C is the most important variable when identifying the cause for rework. According to specialists, Infrastructural features from the city can explain why this behavior

occurs. The only exception in City C is when service "contract renew" and activity "protocol taking" occur, tending not to be reworked. Rework can also be observed when the customer service channel is phone, indicating process inefficiencies.

- In the same way that the Cancellation dataset, Sector 1 takes the case to have a high repetition rate of "protocol transfer" activity. The exception to this is selling unit 62. Also, on Monday, the cases tend to have high repetition in the same way it takes longer to be resolved (weekend accumulation). When Sector 1 is not involved, a high repetition rate is observed when activity "protocol taking" and group "house appointment" occur.

Frame 6.3 – Discovered rules for sales datasets

	Antecedent (A)			**Class (B)**	**Support**	**Confidence**	**Lift**
Duration	Has support level	No worker 95	No worker 587	Normal	0.2209	1.0000	1.1235
			User 587	Long	0.0001	0.5000	4.5498
		Worker 95		Long	0.0001	1.0000	9.0996
	No support level	Monday		Long	0.0811	0.2102	1.9129
		No Monday		Normal	0.3643	0.9272	1.0417
Rework	No city C	No actv integration		Normal	0.7796	1.0000	1.0008
		Actv. integration	No costumer svc. phone	Normal	0.1290	1.0000	1.0008
			Costumer svc. phone	High	0.0003	0.0039	4.9006
	City C	No act. protocol taking	No svc. contract renew	High	0.0005	0.1224	153.3184
			Svc. contract renew	Normal	0.0026	1.0000	1.0008
		Act. protocol taking		Normal	0.0033	1.0000	1.0008
Act. Repetition	No sector 1	No act. protocol taking	No group house appointment	Normal	0.3906	0.9747	2.3088
			Group house appointment	High	0.0066	1.0000	1.7306
		Act. protocol taking	No Act. term validation	Normal	0.0197	0.4387	1.0392
			Act. term validation	High	0.0085	0.9907	1.7146
	Sector 1	No selling unit 62		High	0.5272	0.9811	1.6979
		Selling unit 62	Monday	High	0.0001	1.0000	1.7306
			No Monday	Normal	0.0016	1.0000	2.3687

6.4 Final considerations

By performing RCA in this case study, we could identify several causes for the long duration, rework rate, and repetition of activity "protocol transfer". In general, all the identified causes were considered valid by the process specialists. Some further exploration needs to be conducted to identify why: group 30 tends to normal repetition rate on the Support dataset; groups 28 and 24 tend to have high and normal repetition, respectively (Cancellation dataset); and workers 95 and 587 are involved in long-duration sales. This case study shows the usefulness of combining RCA and process mining to discover causes for inefficiency in a real-world process.

7 Education

As described in Chapter 2 PM is widely applied to several areas, and types of processes, such as Healthcare, Information and Communication Technology, Manufacturing, Financial, Logistics, and Education (GARCIA et al., 2019). When PM is applied to educational data, it is called Educational Process Mining (EPM) (TRCKA; PECHENIZKIY; AALST, 2010). In the last decade, several works have been reported to use EPM for a wide range of educational problems (GHAZAL; IBRAHIM; SALAMA, 2017; BOGARÍN; CEREZO; ROMERO, 2018; GRIGOROVA; MALYSHEVA; BOBROVSKIY, 2017; INTAYOAD; KAMYOD; TEMDEE, 2018).

EPM has been used in computer-supported collaborative learning and collaborative writing to identify process patterns for high and low group performance (SOUTHAVILAY; YACEF; CALLVO, 2010; BOLT et al., 2015a). It has also been used to analyze professional training processes and their conformance with curriculum constraints (ROZINAT; AALST, 2005; CAIRNS et al., 2014), student registration processes (AYUTAYA; PALUNGSUNTIKUL; PREMCHAISWADI, 2012), and computer-based assessment processes (AALST; GUO; GORISSEN, 2013; JUHAŇÁK; ZOUNEK; ROHLÍKOVÁ, 2019; CEREZO et al., 2020; ROMERO et al., 2016). Predicting academic performance has been achieved by using EPM in massive open online courses (ROMERO et al., 2013) and online discussion forums (UMER et al., 2017). The literature also presents studies related to understanding what factor influences dropout students (ARAQUE; ROLDÁN; SALGUERO, 2009; CERDEIRA et al., 2018). Some educational institutions design curricula so students can follow differing paths from start to end. When that is the case, EPM can be applied to curriculum mining, that is, to gain insights about the different paths; understand how programs, courses, and students interact; compare paths that successful and less successful students tend to take, highlighting discrepancies between them; analyze statistically significant and meaningful patterns that lead to different outcomes; provide students, educators, and program coordinators with indicators that could be taken into consideration when interfering in those paths; and help predict the outcome of an attending student (WANG; ZAÏANE, 2015; SCHULTE et al., 2017).

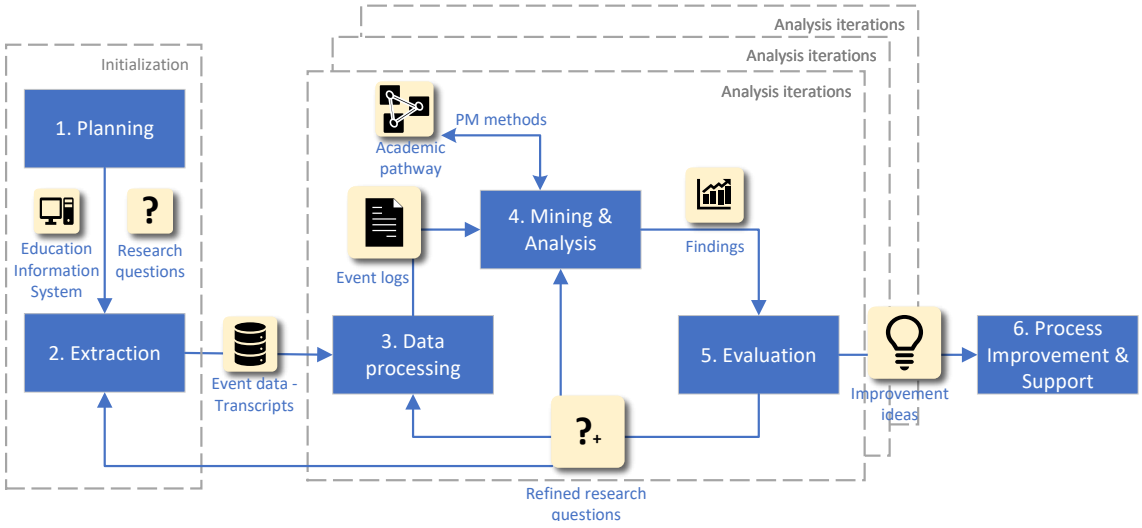
Bolt et al. (2015b) presents an EPM framework, but it focuses on tracking students' classroom activities and performance, not curriculum mining. Therefore, this study presents a method that can be used for curriculum mining, validating it in a case study of a Brazilian public university. In this case study, students may have different paths because some courses are elective, slots for failing courses are not always available, and, in general, courses have prerequisites, i.e., some specified courses must be taken before the student takes another one. Therefore, curriculum mining can be applied to this case. Another

interesting aspect of this particular institution is that admission is currently accepted via a Unified Selection System (SISU), and students can apply from anywhere in Brazil. It means that several students from different backgrounds and aspects can be admitted. Section 7.1 describes the methodology used. In Section 7.2, we present and discuss the results of applying the proposed methodology in the case study.

7.1 Method

In this study, a PM^2 (ECK et al., 2015) specialization for curriculum mining was used, shown in Figure 7.1. The initialization includes Planning and Extraction. Curriculum mining research questions guide us: how programs, courses, and students interact, and what statistically significant and meaningful patterns lead students to different paths and outcomes (dropout and graduate, long vs. short graduation, high GPA vs. low GPA, etc.). In curriculum mining, an event is represented by a student taking a course in the program's scope. Therefore, we extracted the students' transcripts from the educational information system of the university[1]. Since we also wanted to analyze students' background information (gender, age, city of birth, etc.), we extracted that information too.

Figure 7.1 – An overview of the PM^2 methodology specialization for curriculum mining



We processed the event data to create an event log (step 3 on Figure 7.1). Each entry in the log corresponds to a course taken by a student enrolled in a bachelor's degree program. The log contains information about 12,185 enrollments from 11,290 students in the university between 1986 and 2022. Some students have re-enrolled. The log contains students' personal information and information about each enrollment, such as the full transcript with courses, grades, and attendance. In total, the log contains 437,690 entries. The data corresponds to undergraduate students from eight bachelor's degree programs

[1] The study in this chapter was approved by the research ethics committee (CAAE: 57106622.2.0000.0177) (Plataforma Brasil, 2023)

offered on one university campus. A description of the extracted dataset is presented in Frame 7.1. For categorical variables, we present the total count of categories or the percentages of each category. For numerical variables, we show the median (with IQR) and mean (with standard deviation).

Frame 7.1 – Educational Dataset description

Variable	Description
Enrollment Id	Identifier of enrollment (12,185 enrollments).
Undergraduate program Id	Identifier of course: Agronomy (15.44%), Accounting (12.89%), Chemistry (9.62%), Civil Engr. (13.06%), Computer Engr. (10.52%), Electrical Engr. (12.74%), Management (12.87%), or Mechanical Engr. (12.87%).
Shift	Time of the day of the program: morning and afternoon (60.92%), afternoon and night (13.32%), or only night (25.76%).
Admission type	Students could apply locally at the university via an entry test (26.29%). Nowadays, students can apply from anywhere in Brazil via a Unified Selection System (SISU, 2022) shared by most universities (61.52)%. They may also be admitted via program change (within the university) (6.17%), transference (from another university) (1.99%), SISU waiting list (3.68%), and other (0.34%).
Admission score	Grade from 0 to 1000: 613.59 (IQR 533.52-664.50), 566.20 (SD 191.69).
Admission year	Years from 1986 to 2022: 2013 (IQR 2009-2018), 2011.82 (SD 7.55).
Admission age	Ages from 16 to 64: 19 (IQR 18-21), 20.5 (SD 4.43).
Admission season	Students can apply twice a year: Fall admission (69.74%), and Spring admission (30.26%).
Admission quota group	Students could be admitted when no quota policy was implemented (26.70%). Since 2012 (BRASIL, 2012), students can apply as quota holders (33.92%) or as no quota holders (39.38%). Quota groups include candidates with disabilities, self-declared black, brown, or indigenous, with *per capita* gross family income $\leq$ 1.5 minimum wage, or who have attended public high school.
Enrollment situation	Dropout (44.44%), Graduated (35.01%), and Attending (20.55%).
Person Id	Identifier of a student (11,290 students).
Gender	Male (64.47%), or Female (35.53%).
Brazilian	Yes (99.53%), or No (0.47%).
Same state	The state of birth is the same state as the one where the campus is located (59.14%), or other (40.86%).
Same city	The city of birth is the same city as the one where the campus is located (19.68%), or other (80.32%).
High school type	Public (61.82%), Private (32.45%), or not informed (5.72%).
Ethnic group	White (55.30%), Brown (11.43%), Black (1.36%), Yellow (1.06%), Indigenous (0.09%), or not declared (30.76%).
Situation	Students have one or more enrollments. We considered 3 groups: students that have graduated at least once (37.64%), students currently attending a program and who have never graduated (22.11%), and dropouts (have never graduated and are not attending any course) (37.64%).
Event Id	Identifier of an event: a student took a course on the scope of an enrollment (437,690 events)
Course	Course identifier (813).
Class	Class identifier (680).
Total time (hours)	60 (IQR 45-75) 64.17 (SD 24.37).
Grade	Grade obtained in the course, from 0 to 100: 74 (IQR 6q-84) 66.91 (SD 25.45).
Attendance	Attendance obtained in the course, from 0 to 1000: 930 (IQR 850-991) 887.36 (SD 163.51).
Course type	Courses can be mandatory (93.28%) or elective (6.72%).
Teacher	Name of the teacher responsible for the course (4610 different teachers).
Course situation	Students may have passed (69.64%), failed (15.58%), canceled (3.54%), not completed (2.34%), had the credit validated (8.86%), or been dismissed (0.23%).
Date	Start of the course: 2015 Jan (IQR 2010 Jul-2019 Jan) 2013-Jul

Next, we used Mining and Analysis (step 4 on Figure 7.1). First, we enhanced the log by adding the course semester information. In this particular university, courses are assigned by the curriculum creators to a *semester*, a number indicating the preferable order students should take the courses along the program. This order considers prerequisites and guarantees that students will have a place in the next-season courses, in case of passing

the current-season ones. In case of failing, students have to adequate their path to retaking courses and are subject to a lack of places and offerings. We used enhancement to annotate the log with financial cost information. For that, we used the work of Briskiewicz (2016), which defines the 2015 annual cost for each undergraduate program on the campus. The annual cost is defined considering labor, security, cleanliness, depreciation, electricity, water, telephone, and other expenses. We used the Brazilian consumer price index (IBGE, 2022) to infer other years' annual cost for each program (Frame 7.2). Then, we calculated the practiced hourly cost for each program based on the annual cost and the total number of hours of courses taken by students in each program. Results are shown in Frame 7.3[2].

Frame 7.2 – Yearly cost by program (Reals R$)

Year	Mn	Ag	Ac	Cp	Mc	Cv	El	Cm
1986	-	-	17,201.53	-	-	-	-	-
1987	-	78,007.08	17,364.68	-	-	-	-	-
1988	17,171.31	80,898.51	18,008.33	-	-	-	-	-
1989	19,078.75	89,884.96	20,008.75	-	-	-	-	-
1990	24,572.70	115,768.38	25,770.50	-	-	-	-	-
1991	32,188.17	151,646.90	33,757.20	-	-	-	-	-
1992	35,386.38	166,714.47	37,111.30	-	-	-	-	-
1993	46,270.07	217,990.40	48,525.53	-	-	-	-	-
1994	144,961.14	682,949.76	152,027.32	-	-	-	-	-
1995	687,387.57	3,238,462.34	720,894.55	-	-	-	-	-
1996	756,617.51	3,564,622.70	793,499.13	-	-	-	-	-
1997	790,584.47	3,724,649.90	829,121.82	-	-	-	-	-
1998	810,450.89	3,818,245.78	849,956.64	-	-	-	2,672,289.74	-
1999	816,939.84	3,848,816.94	856,761.90	-	-	-	2,693,685.69	-
2000	853,986.83	4,023,354.96	895,614.75	-	-	-	2,815,840.26	-
2001	880,655.77	4,148,999.30	923,583.68	-	-	3,117,758.45	2,903,775.43	2,543,157.39
2002	917,465.74	4,322,420.69	962,187.97	-	-	3,248,075.66	3,025,148.49	2,649,457.22
2003	984,704.70	4,639,201.01	1,032,704.52	-	-	3,486,119.69	3,246,854.70	2,843,629.87
2004	1,041,349.43	4,906,069.13	1,092,110.42	-	-	3,686,657.28	3,433,628.68	3,007,208.50
2005	1,092,717.36	5,148,076.82	1,145,982.29	-	3,437,109.81	3,868,513.55	3,603,003.49	3,155,548.77
2006	1,134,620.32	5,345,492.63	1,189,927.83	-	3,568,914.33	4,016,861.32	3,741,169.62	3,276,556.13
2007	1,159,150.12	5,461,058.91	1,215,653.35	-	3,646,072.08	4,103,703.41	3,822,051.42	3,347,393.27
2008	1,195,872.73	5,634,068.73	1,254,166.02	-	3,761,581.96	4,233,711.34	3,943,136.44	3,453,440.82
2009	1,248,183.30	5,880,517.47	1,309,026.48	4,066,834.97	3,926,123.29	4,418,904.83	4,115,619.42	3,604,503.25
2010	1,289,384.72	6,074,628.14	1,352,236.29	4,201,077.59	4,055,721.14	4,564,768.96	4,251,472.37	3,723,484.71
2011	1,350,513.10	6,362,619.89	1,416,344.39	4,400,246.28	4,247,998.63	4,781,179.87	4,453,030.23	3,900,011.22
2012	1,424,805.04	6,712,628.61	1,494,257.72	4,642,304.51	4,481,681.69	5,044,193.32	4,697,992.11	4,114,551.45
2013	1,498,886.78	7,061,647.04	1,571,950.61	4,883,677.89	4,714,703.61	5,306,462.63	4,942,260.93	4,328,484.68
2014	1,582,134.79	7,453,850.16	1,659,256.57	5,154,916.84	4,976,557.74	5,601,183.00	5,216,753.58	4,568,888.25
2015	1,683,549.63	7,931,641.96	1,765,614.92	5,485,347.01	5,295,555.09	5,960,218.83	5,551,147.48	4,861,753.99
2016	2,076,995.18	9,785,266.69	2,178,239.13	6,767,272.61	6,533,126.31	7,353,121.97	6,848,450.65	5,997,945.90
2017	2,126,659.89	10,019,250.12	2,230,324.77	6,929,090.34	6,689,345.19	7,528,948.43	7,012,209.50	6,141,367.64
2018	2,189,793.00	10,316,686.70	2,296,535.33	7,134,790.86	6,887,928.51	7,752,456.63	7,220,377.53	6,323,683.41
2019	2,262,353.99	10,658,540.47	2,372,633.33	7,371,209.31	7,116,166.93	8,009,342.06	7,459,631.98	6,533,225.01
2020	2,338,450.44	11,017,050.68	2,452,439.12	7,619,147.00	7,355,526.02	8,278,743.95	7,710,543.85	6,752,976.29
2021	2,507,815.53	11,814,973.86	2,630,059.98	8,170,972.91	7,888,258.86	8,878,341.97	8,268,989.29	7,242,068.74
2022	2,592,666.43	12,214,728.62	2,719,046.98	8,447,434.40	8,155,154.84	9,178,737.00	8,548,767.12	7,487,101.14

We used statistical analysis and PM techniques to evaluate several aspects of the dataset. First, we measured the correlation between certain variables and chosen outcomes. For instance, we calculate the correlation of admission score, year, age, season, and quota group concerning the enrollment situation, i.e., Graduated or Dropout. For measuring the likelihood of the null hypothesis (the observed difference is due to chance alone), we used

[2] Agronomy (Ag), Accounting (Ac), Chemistry (Cm), Civil Engr. (Cv), Computer Engr. (Cp), Electrical Engr. (El), Management (Mn), and Mechanical Engr. (Mc)

Frame 7.3 – Practiced hourly cost for each program, based on the annual cost and the total number of hours of courses taken by students in each program (Reals R$)

Year	Mn	Ag	Ac	Cp	Mc	Cv	El	Cm
1986	-	-	26.67	-	-	-	-	-
1987	-	573.58	6.60	-	-	-	-	-
1988	286.19	1189.68	7.93	-	-	-	-	-
1989	30.67	330.46	3.92	-	-	-	-	-
1990	22.18	851.24	0.92	-	-	-	-	-
1991	1.42	1115.05	0.88	-	-	-	-	-
1992	0.64	4.99	0.56	-	-	-	-	-
1993	0.54	4.05	0.50	-	-	-	-	-
1994	1.34	10.12	1.39	-	-	-	-	-
1995	5.35	32.45	5.76	-	-	-	-	-
1996	6.34	28.12	5.77	-	-	-	-	-
1997	7.46	26.08	6.33	-	-	-	-	-
1998	7.68	24.02	6.97	-	-	-	10479.57	-
1999	7.80	22.40	7.45	-	-	-	13813.77	-
2000	7.80	23.14	7.96	-	-	-	14440.21	-
2001	8.65	24.77	8.88	-	-	51962.64	38717.01	10596.49
2002	9.31	26.50	9.68	-	-	18044.86	100838.28	8410.98
2003	10.72	30.90	10.06	-	-	33201.14	72152.33	11151.49
2004	10.12	39.24	10.16	-	-	122888.58	8804.18	11792.97
2005	11.34	31.11	11.15	-	28642.58	16118.81	48040.05	42073.98
2006	10.96	33.47	11.20	-	12522.51	22315.90	11336.88	54609.27
2007	11.27	33.81	10.83	-	141.98	157.14	151.76	109.18
2008	11.58	33.55	11.14	-	68.33	67.25	65.12	51.84
2009	9.42	34.30	10.41	-	37.41	36.02	36.44	31.46
2010	12.54	35.41	11.81	29.96	14.81	13.16	12.36	23.71
2011	12.95	38.62	12.06	29.01	20.16	18.29	20.97	23.63
2012	14.68	40.12	12.63	27.74	20.16	17.61	20.45	26.65
2013	14.61	38.18	13.39	27.14	19.63	17.88	19.95	28.90
2014	16.65	35.26	14.26	28.11	19.67	17.89	18.99	34.78
2015	18.72	34.33	16.05	27.32	19.90	18.06	20.28	33.48
2016	21.69	39.17	18.88	33.61	22.95	20.73	24.25	48.41
2017	20.44	36.22	19.59	34.39	22.16	21.22	23.82	51.04
2018	21.69	34.94	20.53	34.98	21.86	22.71	26.43	50.13
2019	22.87	34.07	22.49	32.36	23.27	23.29	27.87	47.98
2020	26.28	36.98	22.03	35.11	27.20	26.36	32.48	56.61
2021	31.32	42.84	24.36	39.20	34.50	31.28	42.84	71.52
2022	32.07	42.96	26.36	37.73	34.20	34.52	49.23	80.65

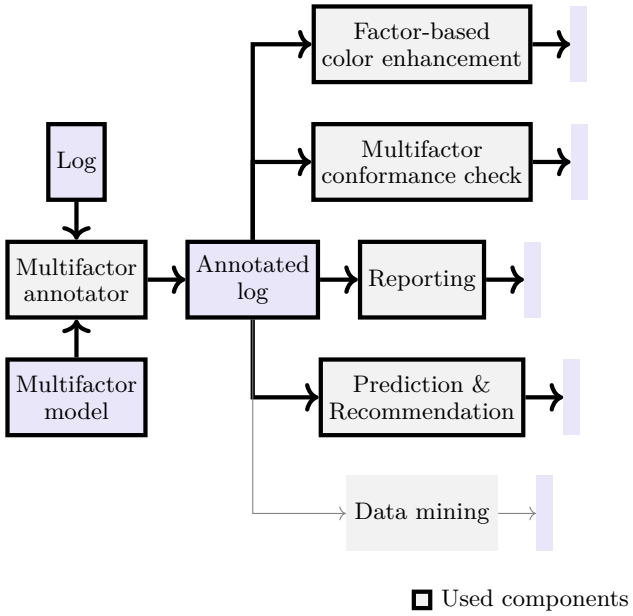
Student's t-test (BONEAU, 1960) and a chi-square test (CORDER; FOREMAN, 2014), with significance level $\alpha = 0.01$. For the variables whose null hypothesis is false, we used entropy test (MAIMON; ROKACH, 2014) to measure each variable's information gain (IG) with respect to the defined outcome. The tests were made to obtain insights about enrollment and students from all the programs. We used PM process discovery to calculate the student's path for each course, and we enhanced the models with grade information.

Then, we selected a specific undergraduate program to perform further exploration. We performed process discovery to obtain the process model for graduated and dropout students at both semester and course levels. Next, we enhanced the discovered process models with grades, attendances, and frequency of each course. Evaluation and suggestions for Process Improvement (steps 5 and 6 in Figure 7.1) are presented in Section 7.2.

7.1.1 Multifactor Framework

As depicted in Figure 7.2, to perform the analysis, the Multifactor Framework from Chapter 5 was used:

Figure 7.2 – Framework's used components in the education case study



- The **Multifactor annotator** was used to add cost information to students' transcripts. The time each student spent in the classroom was used to calculate the total cost of each student. The calculation was based on the practiced hourly cost for each program (Frame 7.3).
- The **Multifactor conformance check** component was used to split the input log into dropout log (non-conform) and undergraduate log (conform log). Each log was further explored to create process models and quantify costs.
- The **Factor-based color enhancement** component was used to create the process models based on grades, attendance, and frequency (Figures 7.7, 7.10, and 7.11).
- The **Reporting** component was used to create several reports in this case study (Figures 7.3, 7.4, 7.8, and 7.9). It was also used to perform descriptive analysis (Frame 7.1) and measure correlation and IG (Frames 7.4 and 7.5 and Figures 7.5, and 7.6).
- The **Prediction** component was used to predict the final GPA of students that are attending a specific course (Figure 7.12).

7.2 Results and Discussion

All the obtained results were discussed with specialists from the university with the goal of understanding the process. We used timeline plot from PM to see how data progresses over time (Figures 7.3 and 7.4). For numerical variables, the plot shows the mean and the 95% confidence interval around the mean; for categorical, a kernel density estimate (KDE) histogram (HÄRDLE et al., 2004) of each category of each variable.

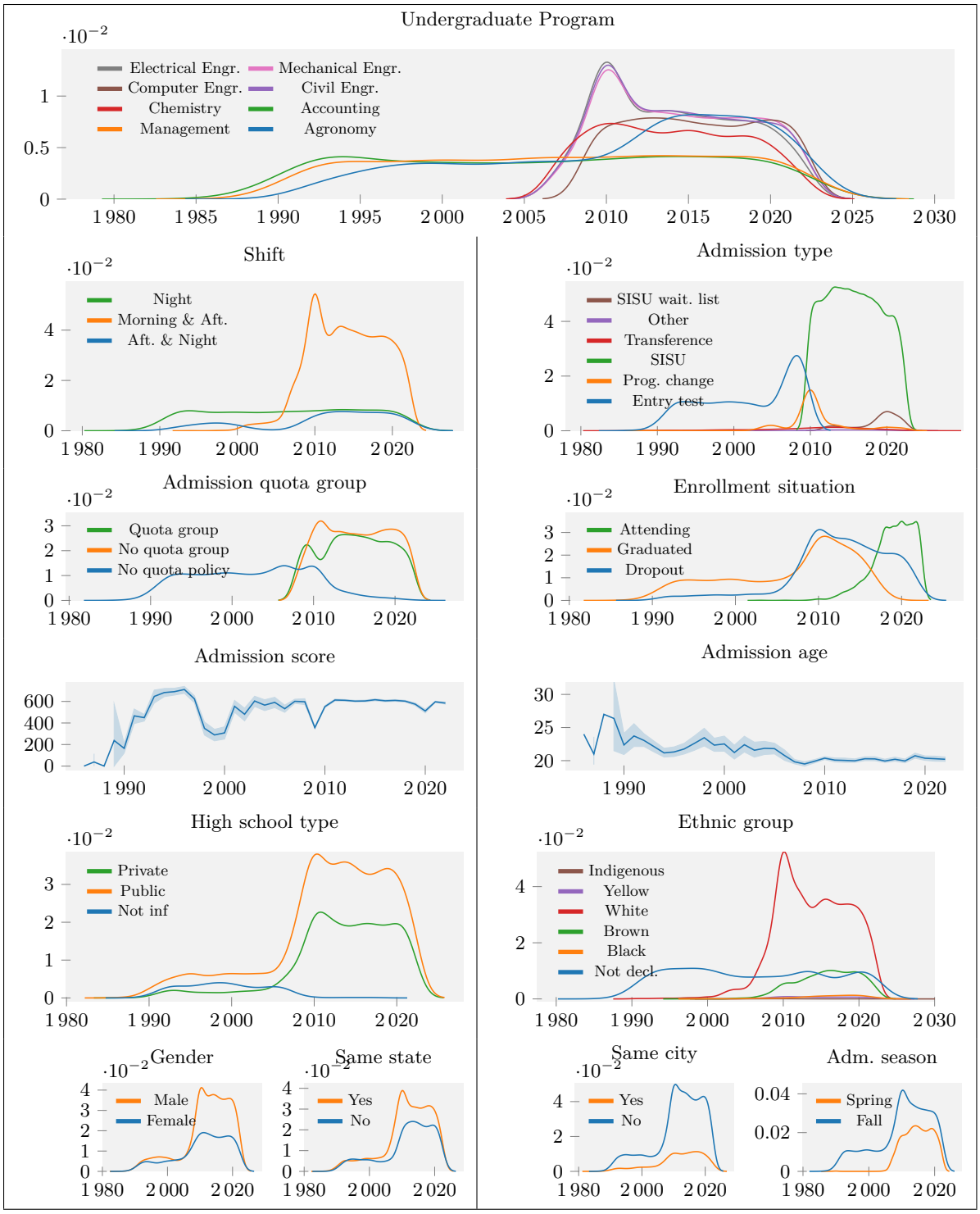
Three programs (Accounting, Management, and Agronomy) have been offered since 1986, while other programs started to be offered in the second decade of the 2000s. Here we observe a drift in the process that reflects in most variables. Around the same date, we can observe a change in Shift (Morning and Afternoon shifts started to be preeminent), Admission type (SISU started to be offered, while the entry test was discontinued), Admission quota (Quota policy was implemented), Admission Age (students started to enroll younger) and Total time (mean total time of courses was reduced). Also, High School Type and Ethnic Groups started to be informed with more frequency, possibly because of the quota policy. Furthermore, the proportion of dropout students surpasses the graduated after the drift, and the number of male students start to be twice the number of female students. Admission score and season, Brazilian, Same city, and Course Situation do not seem to be affected by this first drift. In 2020, another drift can be observed (the covid pandemic occurred). Grades reached the minimum in this case, and not completed courses reached the maximum.

Considering all undergraduate programs, the following correlations were measured for Enrollment variables (course, shift, and admission score, year, age, season, and quota group):

- Enrollment variables with respect to enrollment situation (*dropout* and *graduated*). We excluded the *attending* enrollment situation in this case.
- Enrollment variables concerning the duration of the program. In this case, we considered only the *graduated*. We split enrollments into two groups: long duration (students who took more than five years to graduate) and short duration (students who took five years or less to graduate).
- Enrollment variables concerning graduation GPA. In this university, the graduation GPA is a weighted average considering all taken courses and their total time. In this case, we considered only the *graduated*. We split enrollments into two groups: high GPA (students who scored more than 0.8) and low GPA (students who scored 0.8 or less).

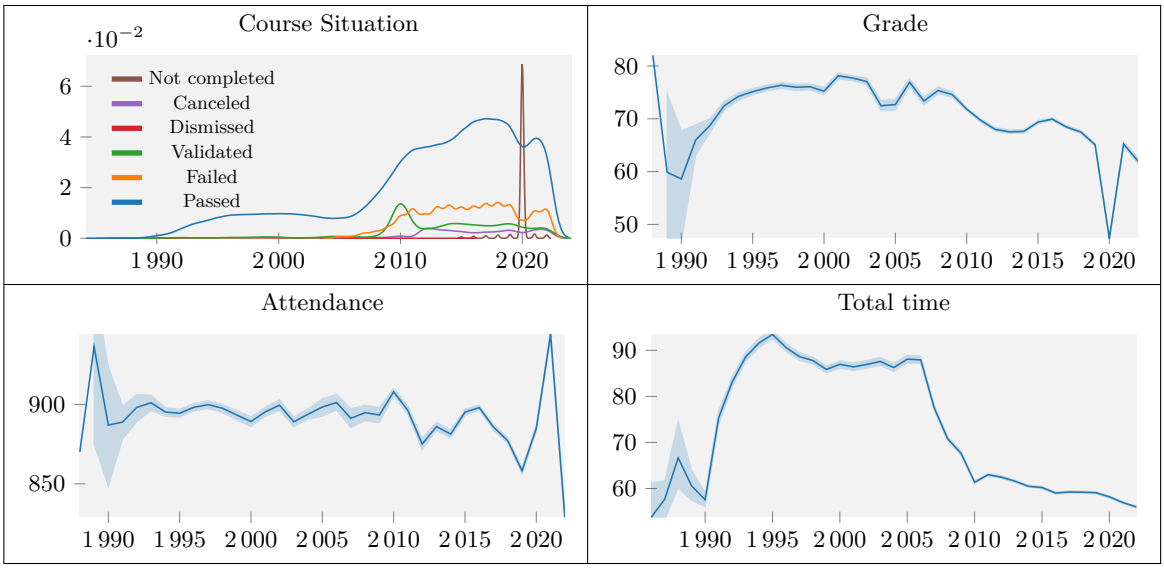
Frame 7.4 shows the results. Figure 7.5 shows the IG of each variable concerning the defined outcome.

Figure 7.3 – Enrollment variables progress over time



When testing the correlation of variables concerning enrollment situation and GPA, tests indicate that the *Admission Score* has the greater IG among the statistically significant variables ($p<0.01$). Enrollments with higher Admission scores correlate to graduated students and high GPAs, while lower with dropout and low GPAs. This result reinforces Voelkle and Sander (2008), indicating that admission score affects dropout through an influence on university grades. Students with lower admission scores perform

Figure 7.4 – Students variables progress over time



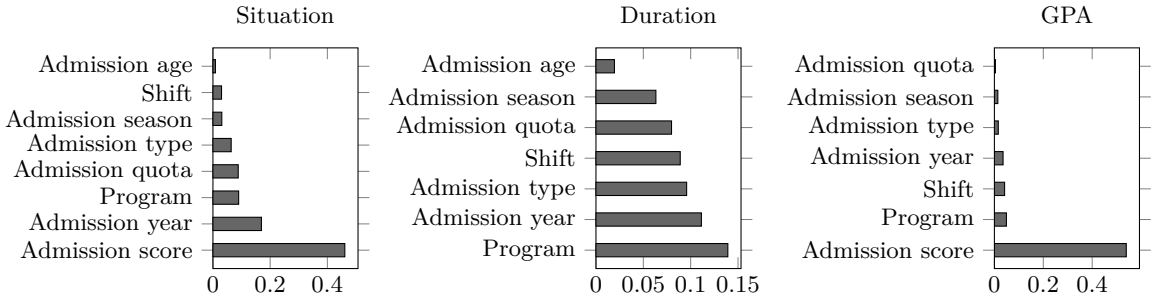
Frame 7.4 – Correlation measurement for enrollments (12,185 total).

	Enrollment situation			Graduation duration			GPA		
Variable	Dropout (5,415)	Graduated (4,266)	P	Short (2,750)	Long (1,516)	P	Low (3,070)	High (1,196)	P
Program			<.001			<.001			<.001
Agronomy	515 (9.51%)	937 (21.96%)		658 (23.93%)	279 (18.40%)		738 (24.04%)	199 (16.64%)	
Accounting	485 (8.96%)	880 (20.63%)		726 (26.40%)	154 (10.16%)		515 (16.78%)	365 (30.52%)	
Chemistry	684 (12.63%)	303 (7.10%)		192 (6.98%)	111 (7.32%)		225 (7.33%)	78 (6.52%)	
Civil Engr.	646 (11.93%)	552 (12.94%)		237 (8.62%)	315 (20.78%)		430 (14.01%)	122 (10.20%)	
Computer Engr.	768 (14.18%)	138 (3.23%)		24 (0.87%)	114 (7.52%)		108 (3.52%)	30 (2.51%)	
Electrical Engr.	907 (16.75%)	338 (7.92%)		126 (4.58%)	212 (13.98%)		301 (9.80%)	37 (3.09%)	
Management	617 (11.39%)	767 (17.98%)		667 (24.25%)	100 (6.60%)		444 (14.46%)	323 (27.01%)	
Mechanical Engr.	793 (14.64%)	351 (8.23%)		120 (4.36%)	231 (15.24%)		309 (10.07%)	42 (3.51%)	
Shift			<.001			<.001			<.001
Morning and aft.	3,482 (64.30%)	2,203 (51.64%)		1,102 (40.07%)	1,101 (72.63%)		309 (10.07%)	42 (3.51%)	
Aft. and night	831 (15.35%)	416 (9.75%)		255 (9.27%)	161 (10.62%)		335 (10.91%)	81 (6.77%)	
Only night	1,102 (20.35%)	1,647 (38.61%)		1,393 (50.65%)	254 (16.75%)		959 (31.24%)	688 (57.53%)	
Admission type			<.001			<.001			<.001
SISU	3,499 (64.62%)	1,769 (41.47%)		787 (28.62%)	982 (64.78%)		1,338 (43.58%)	431 (36.04%)	
Entry test	1,307 (24.14%)	1,897 (44.47%)		1,531 (55.67%)	366 (24.14%)		1,239 (40.36%)	658 (55.02%)	
Sisu waiting list	241 (4.45%)	14 (0.33%)		9 (0.33%)	5 (0.33%)		13 (0.42%)	1 (0.08%)	
Program change	277 (5.12%)	423 (9.92%)		286 (10.40%)	137 (9.04%)		354 (11.53%)	69 (5.77%)	
Transfer	70 (1.29%)	147 (3.45%)		124 (4.51%)	23 (1.52%)		117 (3.81%)	30 (2.51%)	
Other	21 (0.39%)	16 (0.38%)		13 (0.47%)	3 (0.20%)		9 (0.29%)	7 (0.59%)	
Admission score	540.22 (197.22)	584.28 (210.48)	<.001	579.25 (228.59)	593.39 (172.56)	.02	578.54 (211.13)	599.01 (208.19)	.004
Admission age	20.89 (5.14)	20.28 (3.60)	<.001	20.59 (3.81)	19.72 (3.09)	<.001	20.21 (3.45)	20.45 (3.95)	.06
Admission season			<.001			<.001			<.001
Fall	3,497 (64.58%)	3,536 (82.89%)		2,512 (91.35%)	1,024 (67.55%)		2,449 (79.77%)	1,087 (90.89%)	
Spring	1,918 (35.42%)	730 (17.11%)		238 (8.65%)	492 (32.45%)		621 (20.23%)	109 (9.11%)	
Admission quota			<.001			<.001			<.001
No quota policy	1,012 (18.69%)	2,203 (51.64%)		1,752 (63.71%)	451 (29.75%)		1,526 (49.71%)	677 (56.61%)	
Quota group	2,017 (37.25%)	1,007 (23.61%)		525 (19.09%)	482 (31.79%)		726 (23.65%)	281 (23.49%)	
No quota group	2,386 (44.06%)	1,056 (24.75%)		473 (17.20%)	583 (38.46%)		818 (26.64%)	238 (19.90%)	

poorly in the course exams leading to dropouts. Admission score has no statistically significant correlation to the Duration of the enrollment.

The *Program* chosen by the student also correlates to Enrollment Situation, Duration, and GPA (see Frame 7.4). Agronomy, Accounting, and Management programs have been offered since 1986 (see Figure 7.3) and are mostly offered only at night. Chemistry and the four Engineering programs started in the second decade of the 2000s. These programs are offered in the morning and afternoon or afternoon and night. These two groups of programs are characterized by variables *Admission year* and *Shift*. As we can see in Figure 7.5, these variables, along with *Program*, have, in general, a high IG considering the three

Figure 7.5 – Information Gain of each enrollment variable with respect to the defined outcome.



outcomes. In general, nightly programs (Agronomy, Accounting, and Management) have a higher proportion of graduated students, short duration, and high GPA enrollments, while daily programs (Chemistry and Engineering) have a higher proportion of dropouts, long duration, and low GPA. The difference in outcomes of these two groups of programs can be related to the programs' profiles. Chemistry and Engineering programs have complex math and physics curricula, making it more difficult for students who do not have a solid background in these fields to advance in the program (PAURA; ARHIPOVA, 2014).

Admission type, age, season, and quota also are correlated to Enrollment Situation, Duration, and GPA. Students who have been admitted via entry test have a high proportion of graduated, short duration, and high GPA, while students from SISU have a higher proportion of dropouts, long duration, and low GPA. Entry tests used to be used mostly when only Agronomy, Accounting, and Management programs were offered, which is correlated to graduated/short duration/high GPA students, as we said before. Transference and program change' students correlate to graduated, indicating that when a student goes from one program to another, they would not drop out. Almost all students admitted via SISU waiting list have dropped out. The waiting list is implemented so any students who have not gotten a high admission score can enter the university when there is a place. And, as we said, Admission score is highly correlated to Enrollment Situation. Older students have a higher proportion of dropouts but a lower proportion of long-duration enrollments. Students admitted in Spring normally are the ones that did not have admission scores high enough to be admitted in Fall. For that reason, Fall students have a higher proportion of graduates, while Spring students have a higher proportion of dropouts. Quota group students have a higher correlation with dropout.

Next, considering the students from all undergraduate programs, the following correlations were measured for Student variables (gender, birth country, state, and city, high school type, and ethnic group):

- Student variables with respect to situation (*dropout* and *graduated*, excluding *attending*).

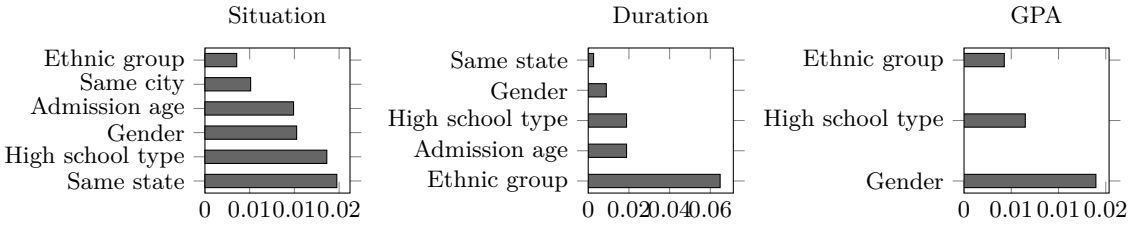
- Student variables concerning the duration of the program. We considered only the first completed graduation for students with more than one.
- Student variables concerning graduation grades. We considered only the first completed graduation.

Frame 7.5 summarizes the results. Figure 7.6 shows the IG of each variable concerning the defined outcome.

Frame 7.5 – Correlation measurement for students (11,290 total)

Variable	Enrollment situation			Graduation duration			GPA		
	Dropout (4,545)	Graduated (4,249)	P	Short (2,719)	Long (1,530)	P	Low (3,060)	High (1,189)	P
Gender			<.001			<.001			<.001
Female	1,409 (31.00%)	1,805 (42.48%)		1,267 (46.60%)	539 (35.23%)		1,169 (38.20%)	637 (53.57%)	
Male	3,136 (69.00%)	2,444 (57.52%)		1,452 (53.40%)	991 (64.77%)		1,891 (61.80%)	552 (46.43%)	
Brazilian	4,523 (99.52%)	4,239 (99.76%)	.08	2,711 (99.71%)	1,528 (99.87%)	.47	3,054 (99.80%)	1,185 (99.66%)	.62
Same state	2,354 (51.79%)	2,798 (65.85%)	<.001	1,861 (68.44%)	956 (62.48%)	<.001	2,037 (66.57%)	780 (65.60%)	.57
Same city	721 (15.86%)	955 (22.48%)	<.001	645 (23.72%)	312 (20.39%)	.001	688 (22.48%)	269 (22.62%)	.95
High school type			<.001			<.001			<.001
Not informed	461 (10.14%)	183 (4.31%)		123 (4.52%)	32 (2.09%)		129 (4.22%)	26 (2.19%)	
Private	1,538 (33.84%)	1,224 (28.81%)		662 (24.35%)	559 (36.54%)		942 (30.78%)	279 (23.47%)	
Public	2,546 (56.02%)	2,842 (66.89%)		1,934 (71.13%)	939 (61.37%)		1,989 (65.00%)	884 (74.35%)	
Ethnic group			<.001			<.001			<.001
Yellow	47 (1.03%)	30 (0.71%)		18 (0.66%)	11 (0.72%)		23 (0.75%)	6 (0.50%)	
White	2,293 (50.45%)	2,218 (52.20%)		1,201 (44.17%)	1,020 (66.67%)		1,636 (53.46%)	585 (49.20%)	
Indigenous	8 (0.18%)	1 (0.02%)		0 (0.00%)	1 (0.07%)		1 (0.03%)	0 (0.00%)	
Not declared	1,660 (36.52%)	1,645 (38.71%)		1,337 (49.17%)	306 (20.00%)		1,120 (36.60%)	523 (43.99%)	
Brown	494 (10.87%)	311 (7.32%)		140 (5.15%)	171 (11.18%)		246 (8.04%)	65 (5.47%)	
Black	43 (0.95%)	44 (1.04%)		23 (0.85%)	21 (1.37%)		34 (1.11%)	10 (0.84%)	

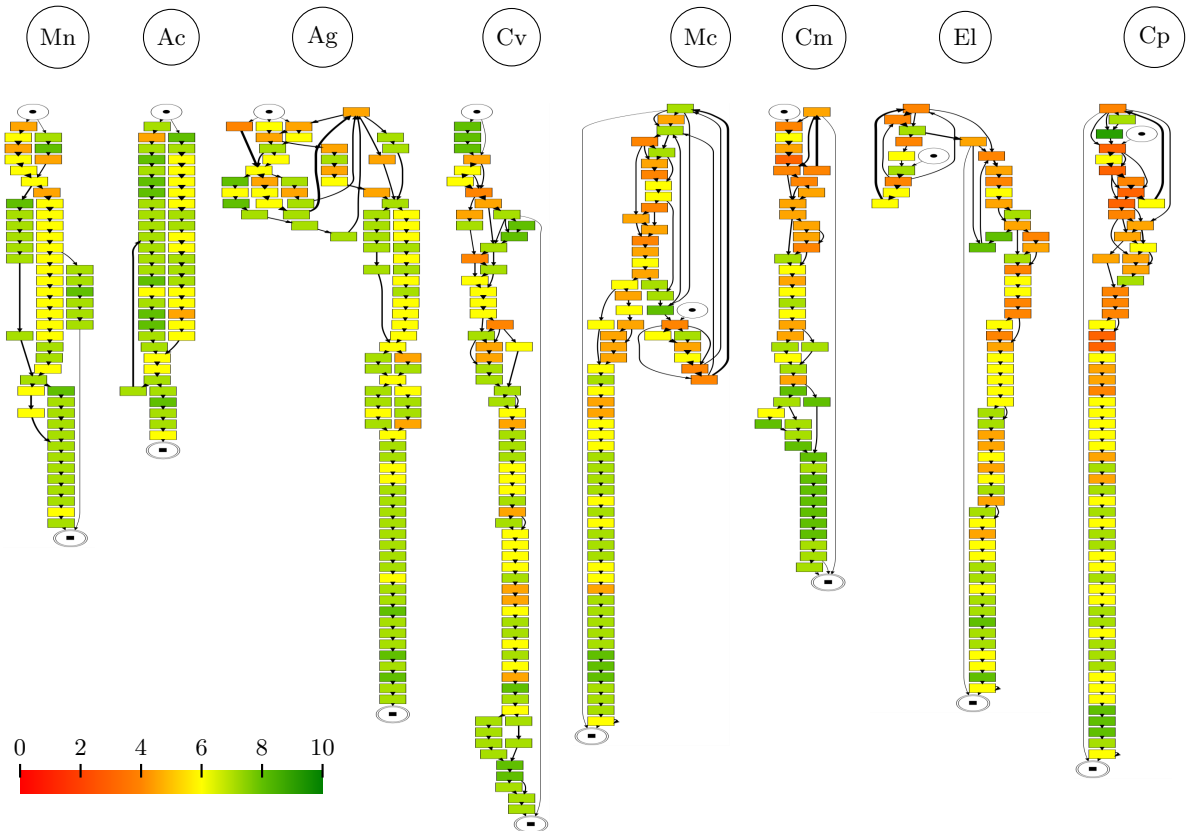
Figure 7.6 – Information Gain of each student variable with respect to the defined outcome.



Gender variable has the greater IG concerning grades. Male students have a higher proportion of dropouts, long duration, and low GPAs. Students from the same state and same city as the university have a higher proportion of graduated and short-duration enrollment. The same state variable has a higher IG considering duration. This reflects the difficulty of students who migrate from other cities or states living away from their families. High school type has the second higher IG considering all outcomes. Private school students have a higher proportion of dropouts, long duration, and low GPAs. Concerning Ethnic group, *not declared*, *white*, and *black* students have a higher proportion of graduated students. Furthermore, *not declared* students are associated with short duration and high GPA. In this case, *not declared* was the only option when only Agronomy, Accounting, and Management programs were offered, so the graduated, short duration, and high GPA correlation was expected.

Figure 7.7 shows the process models for the programs[3]. In the models, each node is colored according to average grade taken by students. Redder means closer to falling grade, while greener means closer to maximum grade. Ag, Ac, Mn, and Cv present more green and yellow colors, while Mc, Cm, El, and Cp have a considerable amount of orange and red. This reflects the results found in Frame 7.4 where Ag, Ac, Mn, and Cv have a higher proportion of Graduate than Dropout. Most programs present a straight-like path, but the older courses (Mn, Ac, and Ag) have parallel paths indicating the change of curricula. Furthermore, some models present spaghetti-like behavior at the start of the model, representing students that tend to retake courses.

Figure 7.7 – Process models for each program - colors represent grades

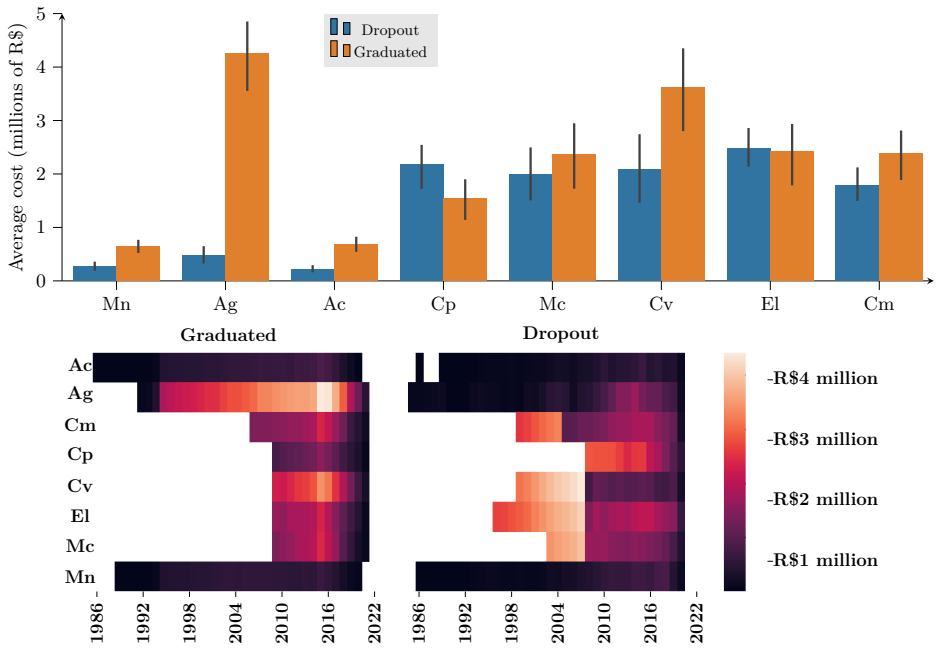


Regarding costs, we calculated the practiced hourly cost for each program based on the annual cost and the total number of hours of courses taken by students in each program. The results are shown in Frame 7.3. With the practiced hourly cost for each program, we split the cost for graduate and dropout students. Figure 7.8 (left) shows the average cost for each program, and (center and right) shows a cost heatmap for graduated and dropout students through the years.

Ag is the program with the higher average cost overall and for graduated students. Its heatmap shows an increasing cost peaking around 2016 for both graduates and dropouts.

[3] Agronomy (Ag), Accounting (Ac), Chemistry (Cm), Civil Engr. (Cv), Computer Engr. (Cp), Electrical Engr. (El), Management (Mn), and Mechanical Engr. (Mc)

Figure 7.8 – Graduates vs. Dropouts - Average cost by program and heatmaps by year



Since most AG students graduate, the average cost for dropouts is 9 times smaller than for graduates. Mn and Ac are the programs with the smaller costs overall, and the other programs (Cp, Mc, Cv, El, and Cm) have a similar cost level, being almost three times the cost of Mn and Ac. This difference can be explained in terms of the programs' curricula, where Mn and Ac do not require labs and have a smaller total time. Also, while Mn and Ac have at least twice the cost for graduates than for dropouts, Cp and El have the cost for dropouts surpassing the graduates.

Furthermore, the heatmaps indicate that Mc, El, Cv, Cp, and Cm have a higher cost with dropout students in the first years of the programs because the number of graduates in these years is low.

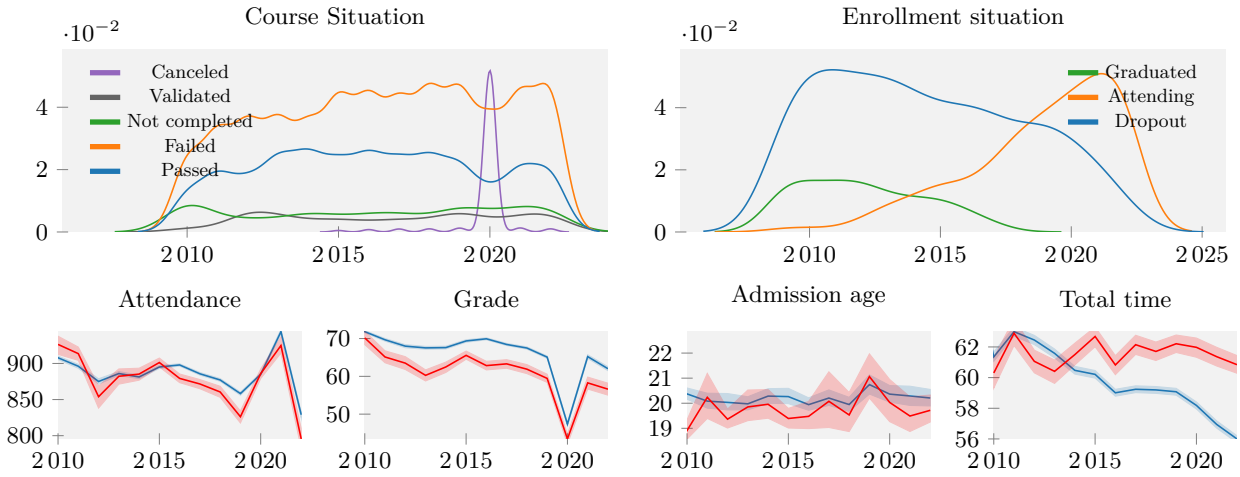
7.2.1 Specific program analysis

We selected a specific undergraduate program, Computer Engineering (Cp), to perform further exploration. For Enrollment variables, we measure the correlation concerning the same three outcomes from before (enrollment situation, duration, and GPA). The following IG were found for the statistically significant variables to enrollment situation: admission age (.027), type (.024), quota (.007), season (.001), and ethnic group (.038). We could not find any statistically significant variables related to Duration and GPA, possibly due to the sample size.

While the timeline plot of all programs (Figure 7.3) shows that the number of passed courses is greater than failed, it is possible to observe in Figure 7.9 that Cp students have more failed courses. In the same way, it is possible to observe that Cp students have

more than twice dropouts than graduated students and lower grades compared to all programs.

Figure 7.9 – Variables progress over time



Red lines correspond to the selected program and blue lines to all bachelor programs

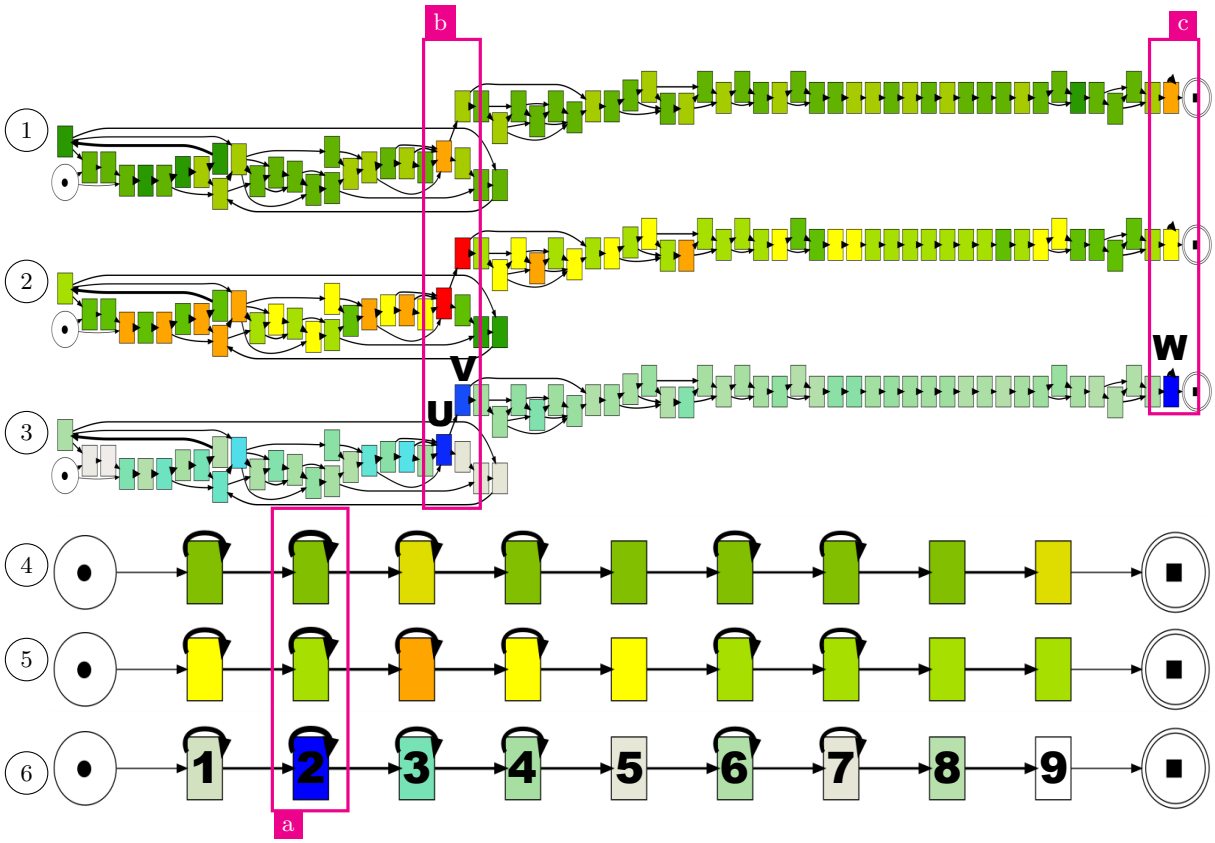
We performed Process Discovery to obtain the process models for graduated (Figure 7.10) and dropout (Figure 7.11) students' paths, using two levels of abstraction: by course (models 1, 2 and 3) and by semester (models 4, 5 and 6). In Figure 7.10, models 3 and 6 are colored based on frequency: bluer means higher frequency; models 1 and 4 are colored based on attendance, and models 2 and 5, on grades. In models 1, 2, 4, and 5, redder means closer to falling grade/attendance, while greener means closer to maximum grade/attendance. Models 1 to 6 are colored in the same way in Figure 7.11.

When analyzing the models in Figures 7.10 (graduated) and 7.11 (dropout), we observe that graduated students follow a more straight-forward path from semester 1 to 9, while dropout students tend to go from 1 to 3, or only attend to semester 1. This indicates students' difficulty in staying at the university, especially at the start of the path when courses that are more dependent on students' high school backgrounds are offered. For clarity purposes, we only show aliases instead of course names in this figures [4].

For graduated students (Figure 7.10), the most retaken semester is the second (a), but this not reflects lower grades/attendance. The lowest attendance can be observed in the third and last semesters, and the lowest grades around semesters 3-5 and semester 1. When analyzing models by courses, we found U course is the most retaken, and it has low grade/attendance, while V has also been retaken but with not so low grade/attendance (b). This reflects two courses where students behave differently: in U, students failed the course and have to retake it without going to classes (high frequency - low grade/attendance),

[4] Complex variables (U), Electric circuits analysis (V), Program conclusion work 2 (W), Integral and differential calculus 1 (X), Physics 1 (Y), Analytic geometry and linear algebra (Z)

Figure 7.10 – Process models for graduated students

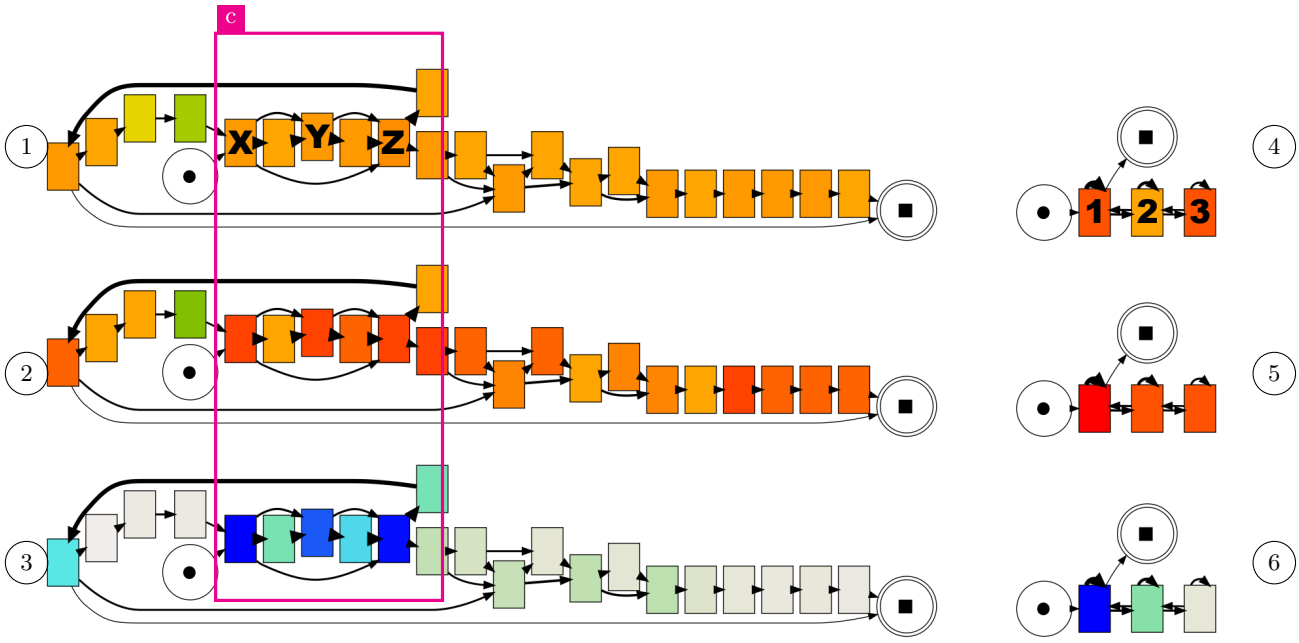


while in *V*, students go to classes and almost pass it (high frequency - low grade - high attendance). *W* also has a pattern of high frequency - low grade/attendance (c). *U*, *V*, and *W* courses should have been further explored to understand why this behavior occurs.

For dropout students (Figure 7.11), the most retaken semester is the first, and students tend to drop out in the first semester. Dropout students have bad attendance and grades in almost all courses and all semesters. The most retaken courses are *X*, *Y*, and *Z*, which, in general, have the lowest grades (c). This reinforces the importance of a good high school background, especially in Math and Physics subjects, since these courses depend on it.

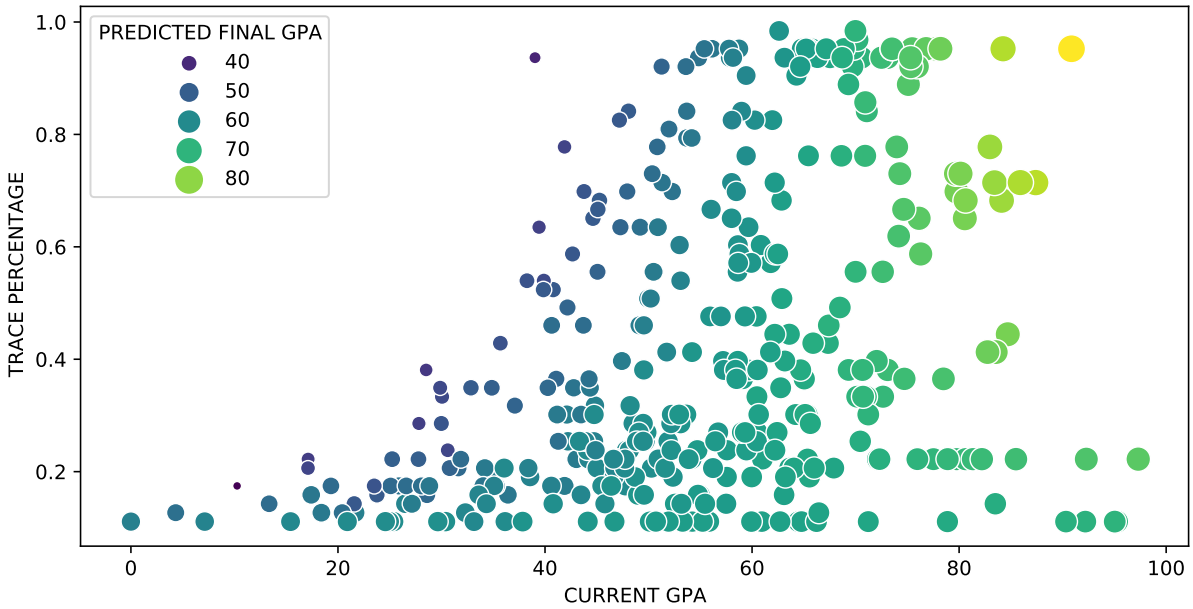
The dropout and graduated students' grades were used to create a predictive model. The attending students were used as partial traces, and the goal of the prediction was set as the final GPA for the attending students. Figure 7.12 shows the prediction result. On the vertical axis is shown the percentage of mandatory courses the attending student has completed. Some students are in the first semester and have completed 0% of the program, while others are close to graduation, with almost 100% of completion. On the horizontal axis, the current GPA is shown from 0 to 100. The dots' color and size represent the predicted final GPA. For students in the first semester, even with a low current GPA, the predicted final GPA is closer to 50. On the other hand, students that are closer to the end of the program will have their GPA less influenced by the upcoming courses, meaning

Figure 7.11 – Process models for dropout students



that the current and final GPAs are almost the same.

Figure 7.12 – Predicted final GPA for attending students



7.3 Final considerations

This study presents a methodology to perform curriculum mining, validating it in a case study of a Brazilian public university. The methodology can be used to help people understand: what are the variables with a higher correlation to the successful and less-successful students (in one program or overall); how data progress overtime and what

different paths graduated, and dropout students tend to take; how grades, attendance, and frequency progresses along these paths; and how costs are quantified for dropout and graduates. The results were discussed with educators and specialists from the Brazilian university. Specifically, in this case study, we found that admission scores, the program, high school type, gender, and location are the variables with a higher correlation to successful and less-successful students.

8 Healthcare

This case study[1] aims to apply computational methods to understand and optimize a hospital surgical center). The surgical center is the place that houses a significant part of the treatment of the patient who undergoes surgery, going through different stages, carried out in parallel or sequence, from screening, scheduling, preparation, and performance of surgery and postoperative. The complexity involved in this scenario is determined not only as a function of the number of stages, patients, different profiles of health professionals, and their achievements in parallel and/or in sequence but also the need to automate the collection and transformation of data into useful information targeting the optimization of the operating room. As a possible result of the phase of understanding this complexity involved, focusing on the respective optimization, it is possible to obtain an indication of the need for some reconfigurations, for example, the allocation of resources to reach a certain goal.

This way, the importance of a detailed and objective understanding of the scenario is perceived. However, when the search for such an understanding is carried out traditionally, a challenge is imposed that is not trivial nor even effective; that is, there is a risk of subjectivity—being oriented only from the point of view of the people involved in the process and not complemented by the vision from the data.

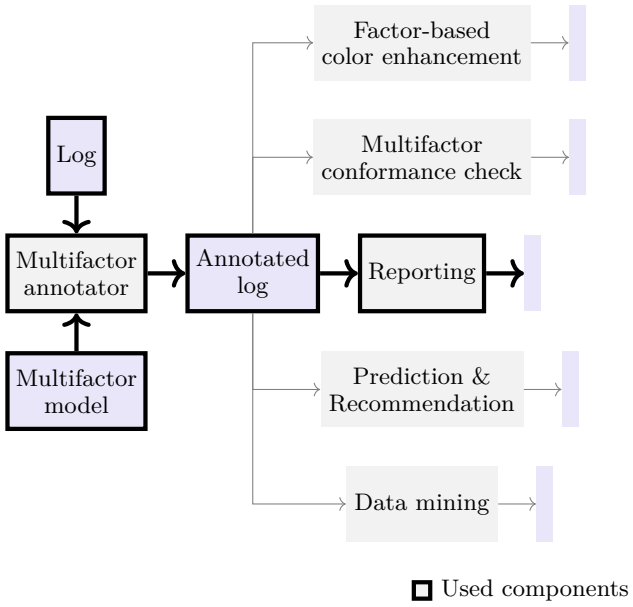
As an alternative to the traditional way, if computational tools correctly support such a search for understanding, it becomes more objective—performed by a machine; and instantaneous—performed by a machine; low cost—subject to one machine's processing time; and continuous, predictive, and prescriptive. Considering this, this study aims to build a computational model that allows an improved understanding of the scenario of the surgical center. This is done by characterizing each process and its resources (e.g., rooms) and how the variables involved in the processes progressed over time.

8.1 Method

As depicted in Figure 8.1, the **reporting** component of the Multifactor Framework from Chapter 5 was used to perform descriptive analysis and to understand how data progresses over time. The results were analyzed by a process specialist and validated. An event log extracted from the surgical center system was used. The log contains 259,991 events from 27,295 surgeries from 2019 to 2022, with 53 categorical variables. All sensitive information was replaced by pseudonyms. The income related to surgeries was extracted

[1] The study in this chapter was approved by the research ethics committee in 2022 (CAAE: 61432722.4.0000.0020) (Plataforma Brasil, 2023)

Figure 8.1 – Framework's used components in the healthcare case study



from the Brazilian Management System for the Unified Health System Tables (SIGTAP[2]) in January 2023 (BRASIL; SAÚDE, 2023). The **annotator** was used to include the income information in the annotated log.

8.2 Preliminary results

The description of some variables can be seen in Frame 8.1. Figure 8.2 shows how these variables progress over time. Figure 8.4 shows how income variables progress over time. Figure 8.3 shows the average duration, and 8.5 shows the average hospital income of each surgery plotted vs some selected variables.

Each surgery has an identifier, and the same patient can undergo more than one surgery. Most surgeries (70.83 %) are on adults (18 to 60 years), and 26.68% are on the elderly (over 60 years). The proportion of adults and elderly do not change over time. The duration of surgeries is greater for adults and the elderly than for people under 18 years old.

Most surgeries have an emergency classification (56.65%), while elective (26.98%) and urgency (16.18%) have a considerable part. Emergency surgeries have a greater average duration than the other types. While emergency and urgency have a constant proportion over time, elective surgeries seem to oscillate. Contamination info has been recorded in 4.42 % of the cases, being clean surgery the most frequent contamination type. On March 2022, a peak of potentially contaminated surgery was observed. Contaminated surgeries

[2] from Portuguese: *Sistema de Gerenciamento da Tabela de Procedimentos, Medicamentos, Órteses, próteses e meios auxiliares de locomoção do Sistema Único de Saúde*

Frame 8.1 – Surgical center dataset description

Variable	Description
Surgery Id	Identifier of surgery (27,295 surgeries).
Age	Less than 18 years (2.39%), 18 to 60 years (70.83 %), or over 60 years (26.68 %).
Classification	Emergency (56.65%), Elective (26.98%), Urgency (16.18%), or other (0.08%).
Contamination	Clean (3.05%), contaminated (0.47%), potentially contaminated (0.45%), or infected (0.45%).
Anesthetist	Sarah Brown (50.70%), Jordan Gaines (4.66%), Ana Brown (3.80%), Mark Little (3.68%), Melinda Hill (3.61%), or other 71 Anesthetist with smaller contributions.
Room	Room 02 (18.44%), Room 04 (15.89%), Room 03 (13.60%), Room 08 (12.57%), Room 09 (10.14%), Room H (7.50%), Room ME (5.11%), or Room 10 (4.67%).
Status	Performed (87.44%), Canceled (10.95%), Expected (0.93%), or Interrupted (0.58%).
Reason for interruption	Fasting (0.16%), abdominal distention (0.06%), patient refusal (0.04%), medicine reconciliation (0.04%), procedures (0.04%), exam (0.02%), hemodialysis (0.02%), oral diet (0.01%), vomiting (0.01%), no pain (0.01%), medical order (0.01%), stasis (0.00%), or other (0.00%).
Description	Description of the procedure (785 different descriptions).
Complexity	High (18.09%), middle (80.49%), or low (1.04%).
Surgeon	Antonio Weiss (4.29%), Adam Torres (3.63%), Mr. Thomas Collins (3.62%), Matthew Davis Jr. (3.14%), Stephanie Washington (3.08%), or other 223 surgeons with smaller contributions.
Specialty	Orthopedist and traumatologist (46.68%), general (16.66%), neurosurgery (8.07%), hand surgery (4.76%), vascular surgery (4.61%), radiology and imaging diagnosis (4.38%), urologist (3.71%), otorhinolaryngologist (2.48%), plastic (1.59%), coloproctologist (1.07%), thoracic (0.89%), dental - general (0.66%), anesthesiologist (0.23%), ophthalmologist (0.21%), neurologist (0.10%), cardiologist (0.09%), head and neck (0.06%), oral and maxillofacial traumatologist (0.05%), intensive medicine (0.02%), cardiovascular (0.01%), nephrologist (0.01%), or interventional cardiologist (0.00%).
Area	Knee (16.26%), hand (4.76%), hip (4.45%), foot and ankle (2.90%), elbow and shoulder (2.70%), column (1.98%), shoulder (0.59%), foot (0.16%), general and knee (0.01%).
Service type	Hospitalized (91.83%), external (2.47%), first aid (0.64%), and outpatient care (0.07%).

have a greater duration than other types.

One anesthetist (Sarah Brown) has more than half (50.70%) surgeries associated with it over the analyzed time window. The use of rooms has fairly equal distribution among all rooms over time. However, while the use of some rooms has increased (rooms 10, ME, H, and 9), others have decreased (4 and 8). Rooms 8, 4, and 3 have the greater average duration, while rooms H, ME, and 10 have the smaller.

87.44% of surgeries were performed, while 10.95% were canceled. This proportion does not seem to change over time. No fasting was the most frequent interruption reason in 2021 and 2022, while procedures were in 2020. Abdominal distension and patient refusal have increased from 2021 to 2022.

Most surgeries have middle complexity (80.49%) and high complexity (18.09%). Complexity does not change over time and does not seem to affect surgery duration. Regarding specialty, Orthopedist and traumatologist (46.68%) is the most frequent one. Ophthalmology, Cardiovascular, and Neurosurgery are the ones with greater average duration.

Knee is the most frequent surgery area (16.26%), and Knee & hip is the area with greater duration. The proportion of surgery area does not seem to change over time, except for foot & ankle (increased) and elbow & shoulder (decreased). Hospitalized is the most frequent service type (91.83%).

Figure 8.2 – Surgery variables progress over time

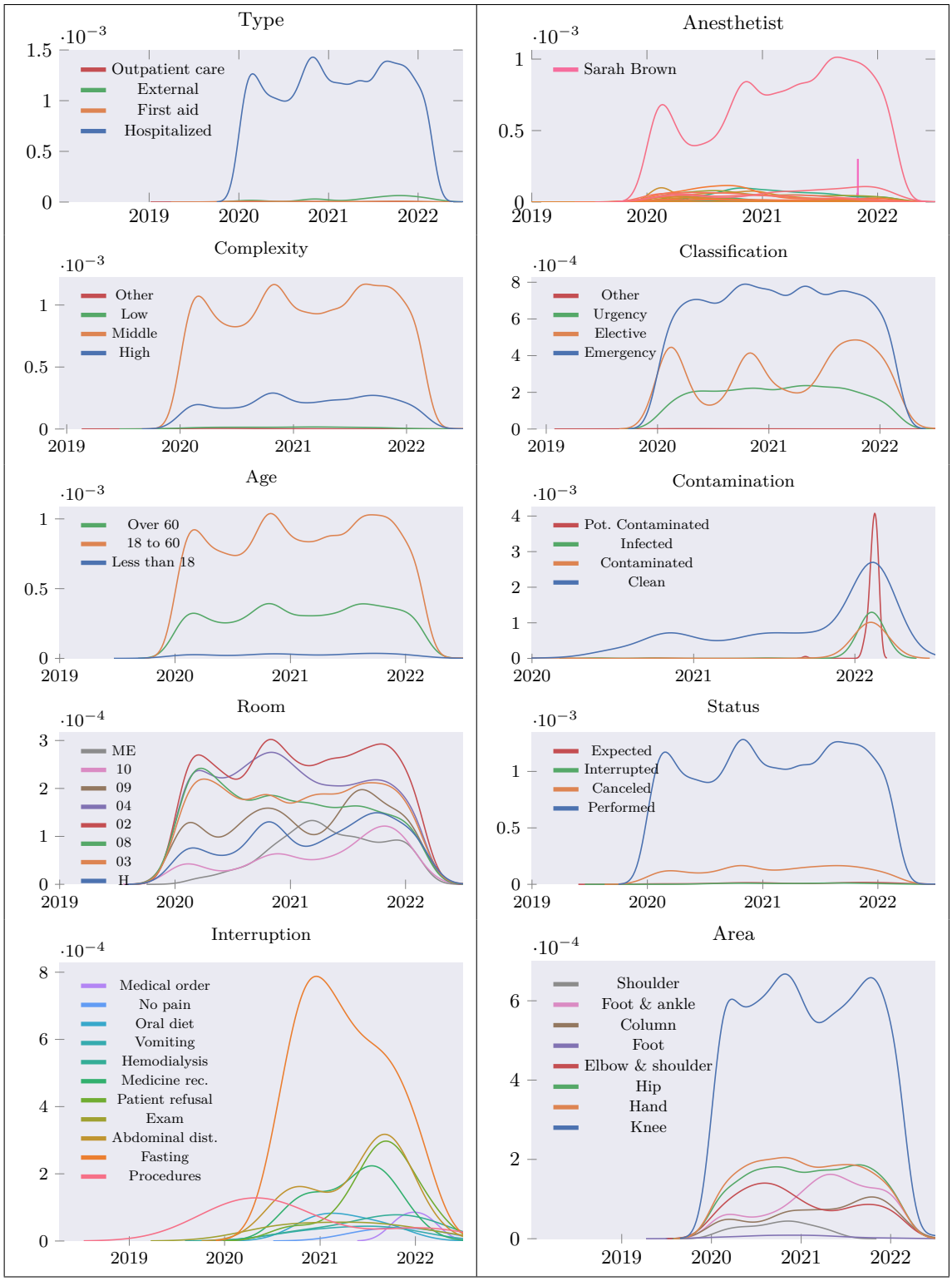


Figure 8.4 shows the average hospital income paid by the Brazilian government. Hospital income relates to daily rates, room rates, food, hygiene, patient support staff in bed, materials, medications, and therapeutic diagnostic support service (BRASIL; SAÚDE, 2023). Average hospital income varied from 600 to 800 reals for surgery, being the maximum value in November 2022.

Figure 8.3 – Surgery variables average duration

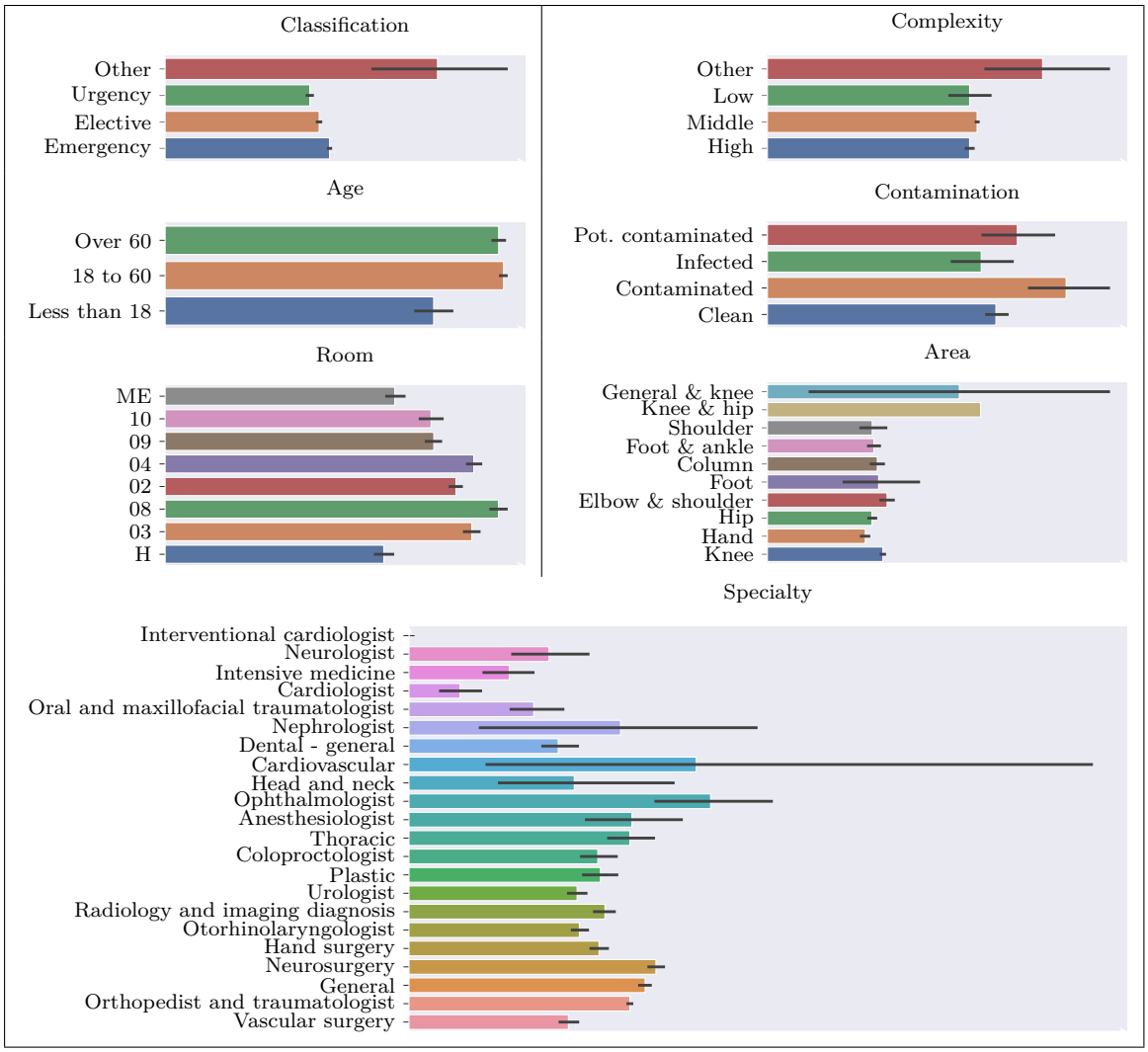


Figure 8.4 – Income variables progress over time

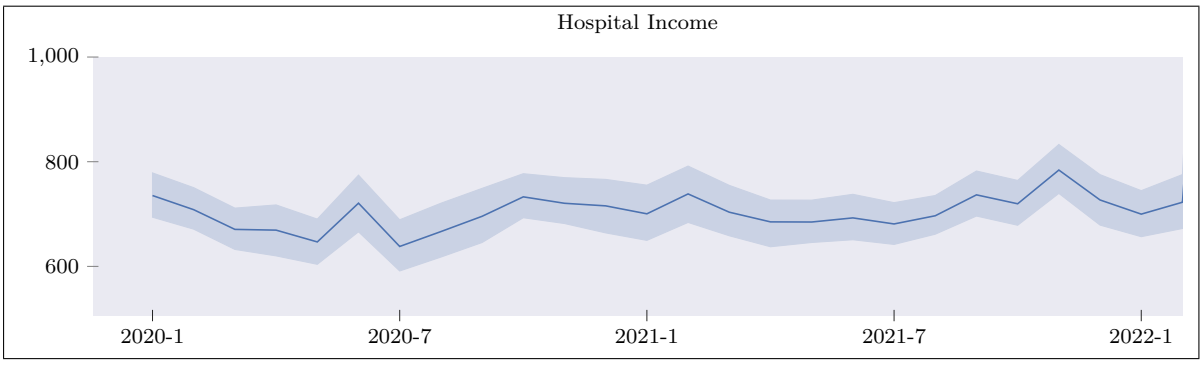
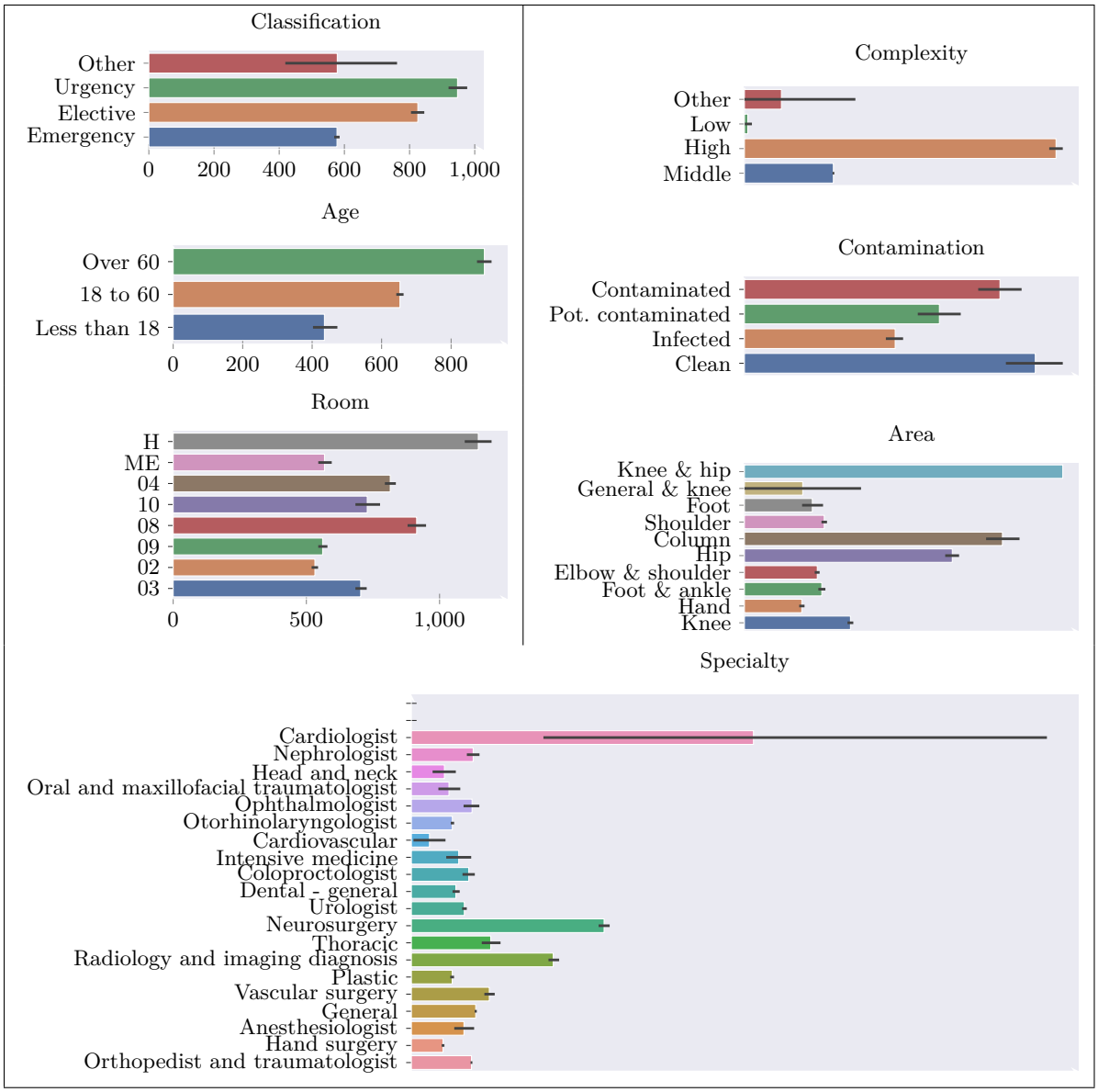


Figure 8.5 shows the average hospital income vs. surgery variables. Surgeries with high complexity, urgency classification, and patients over 60 years old have a greater average income. Surgeries with low complexity, emergency classification, and patients less than 18 years old have a smaller average income. The greater average income is observed in the cardiology and neurosurgery specialty, knee and hip, column and hip areas, and

Figure 8.5 – Surgery variables average hospital income



clean contamination type. Room H has the greater average income, while room 02 has the lower.

8.3 Considerations

A process specialist validated the results. It was identified that the surgery rooms were wrongly recorded in the healthcare system. Also, one anesthetist had half of the surgeries (Sarah Brown) because the staff recorded most of the time the name of the anesthetist in chief and not the in-room anesthetist. All other variables seem to have an expected behavior.

Considering this, the surgical center staff started recording the room and anesthetist's information correctly. The change is currently being implemented. For the next

steps, an interactive dashboard is proposed to be implemented.

9 Conclusion

This work has presented several aspects of process mining and its use in process mining activities, highlighting the cost aspects. The idea of expanding the cost dimension to a multifactor perspective contributes to the analysis of other factors in the process. A computational framework that enables multifactor process mining is presented.

XES standard has been used to represent event logs. A multifactor cost extension for XES is proposed. This extension can represent costs and other factors depending on the process domain. For modeling factors, an XML multifactor model is presented. A multifactor annotator is presented, enabling the user to model the multifactors and automatically annotate them in the event log.

A factor-based color enhancement was proposed to decorate a process model with selected factor information based on a color map. The conformance component creates a report based on structural conformance and factor-value conformance. It also can split the input log into the conform log and non-conform log.

The reporting component creates a report based on a given event log with multifactor information. The following types of reports are available: factor vs. numerical, factor vs. categorical, histograms, and statistical analysis. The statistical analysis report exports a descriptive analysis of the dataset. A categorical variable can be set as the outcome and the correlation between variables, and the outcome is measured by statistical tests. The information gain is shown.

The prediction/recommendation component is responsible for predicting a resulting variable for a partial trace. Its input is an event log to be used as historical data, and the prediction is made by historical information over the log. It is possible to set a goal, and based on the prediction, a recommendation for the next activity is made. The Data Mining component creates a data frame representation of the event log. The component can represent case and event attributes, including factors, in the data frame information.

Case study design science research method was used to validate the framework. The framework obtained an 85.7 usefulness score in a questionnaire applied to process mining researchers and students. Three case studies were used. First, a case study on an educational dataset from a Brazilian public university. The aim of the study was to understand how programs, courses, and students interact and what statistically significant and meaningful patterns lead students to different paths and outcomes. We compared students' outcomes and paths were compared, considering enrollment and course variables along with students' demographic information. Cost, grades, and attendance factors were used. Second, a case study on a real-world Brazilian telecommunication company was

performed. It explores the association between process mining and root cause analysis to identify causes for factors such as short and long-duration services, high rework rates, and activity repetition. The third case study explores a surgery center in a Brazilian hospital. The duration, frequency, cost, and revenue of surgery rooms were analyzed. Specialists in the domain validated all case studies. A 96.9 correctness score was given by the specialists.

The proposed framework can help consolidate multifactor process mining as a tool for business process analysis. However, this work presents some limitations. The factor-based color component can only accept one factor at a time. The prediction and recommendation component uses history annotation to predict, which could be improved by Machine Learning or other types of predictions. The reporting component accepts at most four factors (heatmap) on charts. The multifactor annotation is made offline and includes a computational overhead to the process mining process.

Also, this work will enable some future work. Since this work supports multifactors, multiple-criteria decision-making may be used to evaluate conflicting options: cost or price and other quality measures. Multicriteria analysis is a method of analyzing alternatives for solving problems that use several criteria related to the object of study, making it possible to identify priority alternatives for the object under consideration (FRANCISCO et al., 2007). High-level multiple-criteria decision-making may be used *along with* business process decision-making.

Furthermore, the prediction and recommendation component may be enhanced to include simulation models. It is possible to use multifactors and prediction techniques to generate a close-to-the-reality simulation model. These created models can be exploited to play out the behavior of the processes and run what-if analysis. Simulation can be used to explore different design alternatives and anticipate future performance problems (AALST, 2018). Feature selection approaches can be explored in the context of the data mining component.

Additionally, multifactor may be explored in the context of process concept drift. Concept drift is a problem in data mining, referring to an online supervised learning scenario when the relation between the input data and the target variable changes over time (SATO et al., 2021a). The data perspective in concept drift concerns the data produced or consumed by the process during the execution of its activities. A change in this perspective represents a change in the data related to the case or the event. It is possible to detect data drift considering a specific factor or multifactors.

Bibliography

AALST, W. M. V. D.; REIJERS, H. A.; SONG, M. Discovering social networks from event logs. *Computer Supported Cooperative Work (CSCW)*, Springer, v. 14, n. 6, p. 549–593, 2005. Cited in page 45.

AALST, W. M. Van der. The application of petri nets to workflow management. *Journal of circuits, systems, and computers*, World Scientific, v. 8, n. 01, p. 21–66, 1998. Cited in page 31.

AALST, W. M. Van der. Business process management: a comprehensive survey. *International Scholarly Research Notices*, Hindawi, v. 2013, 2013. Cited in page 19.

AALST, W. M. Van der. Data scientist: The engineer of the future. In: *Enterprise interoperability VI*. [S.l.]: Springer, 2014. p. 13–26. Cited 2 times in pages 30 and 40.

AALST, W. M. Van der. Extracting event data from databases to unleash process mining. In: *BPM-Driving innovation in a digital world*. [S.l.]: Springer, 2015. p. 105–128. Cited in page 171.

AALST, W. M. van der. Process mining and simulation: A match made in heaven! In: *SummerSim*. [S.l.: s.n.], 2018. p. 4–1. Cited in page 137.

AALST, W. M. van der; GUO, S.; GORISSEN, P. Comparative process mining in education: An approach based on process cubes. In: SPRINGER. *International symposium on data-driven process discovery and analysis*. [S.l.], 2013. p. 110–134. Cited in page 112.

AALST, W. M. Van der; MEDEIROS, A. K. A. de. Process mining and security: Detecting anomalous process executions and checking process conformance. *Electronic Notes in Theoretical Computer Science*, Elsevier, v. 121, p. 3–21, 2005. Cited in page 19.

AALST, W. M. Van der; PESIC, M.; SONG, M. Beyond process mining: From the past to present and future. In: SPRINGER. *International Conference on Advanced Information Systems Engineering*. [S.l.], 2010. p. 38–52. Cited in page 52.

AALST, W. M. Van der; SCHONENBERG, M. H.; SONG, M. Time prediction based on process mining. *Information systems*, Elsevier, v. 36, n. 2, p. 450–475, 2011. Cited 3 times in pages 35, 55, and 100.

AALST, W. V. D. Data science in action. In: *Process mining*. [S.l.]: Springer, 2016. p. 3–23. Cited 21 times in pages 29, 30, 31, 32, 35, 36, 37, 39, 40, 41, 42, 44, 45, 46, 47, 48, 49, 52, 53, 54, and 56.

AALST, W. V. D. et al. Process mining manifesto. In: SPRINGER. *International Conference on Business Process Management*. [S.l.], 2011. p. 169–194. Cited 2 times in pages 30 and 38.

AALST, W. V. D. et al. Process mining manifesto. In: SPRINGER. *International Conference on Business Process Management*. [S.l.], 2011. p. 169–194. Cited in page 161.

AALST, W. Van der; ADRIANSYAH, A.; DONGEN, B. van. Replaying history on process models for conformance checking and performance analysis. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, Wiley Online Library, v. 2, n. 2, p. 182–192, 2012. Cited in page 43.

AALST, W. Van der; WEIJTERS, T.; MARUSTER, L. Workflow mining: Discovering process models from event logs. *IEEE transactions on knowledge and data engineering*, IEEE, v. 16, n. 9, p. 1128–1142, 2004. Cited in page 39.

ABDULLATEEF, A. O.; MOKHTAR, S. S. M.; YUSOFF, R. Z. The mediating effects of first call resolution on call centers' performance. *Journal of Database Marketing & Customer Strategy Management*, Springer, v. 18, p. 16–30, 2011. Cited in page 20.

ADAMS, M. et al. Realisation of cost-informed process support within the yawl workflow environment. In: SPRINGER. *Asia-Pacific Conference on Business Process Management*. [S.l.], 2015. p. 3–18. Cited 3 times in pages 20, 22, and 24.

AGRAWAL, R.; SRIKANT, R. et al. Fast algorithms for mining association rules. In: SANTIAGO, CHILE. *Proc. 20th int. conf. very large data bases, VLDB*. [S.l.], 1994. v. 1215, p. 487–499. Cited in page 187.

AGUIRRE, S.; PARRA, C.; ALVARADO, J. Combination of process mining and simulation techniques for business process redesign: a methodological approach. In: SPRINGER. *International Symposium on Data-Driven Process Discovery and Analysis*. [S.l.], 2012. p. 24–43. Cited in page 105.

ALHARBI, A.; BULPITT, A.; JOHNSON, O. Improving pattern detection in healthcare process mining using an interval-based event selection method. In: SPRINGER. *International conference on business process management*. [S.l.], 2017. p. 88–105. Cited in page 175.

ANDREWS, R. et al. Quality-informed semi-automated event log generation for process mining. *Decision Support Systems*, Elsevier, p. 113265, 2020. Cited in page 173.

ANDREWS, R. et al. Towards event log querying for data quality. In: SPRINGER. *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. [S.l.], 2018. p. 116–134. Cited in page 180.

ANDREWS, R. et al. Leveraging data quality to better prepare for process mining: an approach illustrated through analysing road trauma pre-hospital retrieval and transport processes in queensland. *International journal of environmental research and public health*, Multidisciplinary Digital Publishing Institute, v. 16, n. 7, p. 1138, 2019. Cited 3 times in pages 161, 179, and 180.

ARAQUE, F.; ROLDÁN, C.; SALGUERO, A. Factors influencing university drop out rates. *Computers & Education*, Elsevier, v. 53, n. 3, p. 563–574, 2009. Cited in page 112.

AYUTAYA, N. S. N.; PALUNGSUNTIKUL, P.; PREMCHAISWADI, W. Heuristic mining: Adaptive process simplification in education. In: IEEE. *2012 tenth international conference on ict and knowledge engineering*. [S.l.], 2012. p. 221–227. Cited in page 112.

BAIER, T.; MENDLING, J.; WESKE, M. Bridging abstraction layers in process mining. *Information Systems*, Elsevier, v. 46, p. 123–139, 2014. Cited in page 174.

Batista, E.; Solanas, A. Process mining in healthcare: A systematic review. In: *2018 9th International Conference on Information, Intelligence, Systems and Applications (IISA)*. [S.l.: s.n.], 2018. p. 1–6. Cited in page 162.

BAUER, M. et al. Elpaas: Event log privacy as a service. In: CEUR WORKSHOP PROCEEDINGS [UNIVERSITY PUBLISHER]. [S.l.], 2019. Cited in page 181.

BAYOMIE, D. et al. Deducing case ids for unlabeled event logs. In: SPRINGER. *International Conference on Business Process Management*. [S.l.], 2016. p. 242–254. Cited in page 178.

BECKER, T.; INTOYOAD, W. Context aware process mining in logistics. *Procedia Cirp*, Elsevier, v. 63, p. 557–562, 2017. Cited in page 19.

BEEST, N. V.; MARUSTER, L. A process mining approach to redesign business processes-a case study in gas industry. In: IEEE. *Ninth international symposium on symbolic and numeric algorithms for scientific computing (SYNASC 2007)*. [S.l.], 2007. p. 541–548. Cited 2 times in pages 19 and 161.

BERGENTHUM, R. et al. Process mining based on regions of languages. In: SPRINGER. *International Conference on Business Process Management*. [S.l.], 2007. p. 375–383. Cited in page 39.

BEZERRA, F.; WAINER, J. Algorithms for anomaly detection of traces in logs of process aware information systems. *Information Systems*, Elsevier, v. 38, n. 1, p. 33–44, 2013. Cited in page 176.

BLACK, J.; HASHIMZADE, N.; MYLES, G. *A dictionary of economics*. [S.l.]: Oxford university press, 2012. Cited in page 72.

BODENHEIMER, T.; SINSKY, C. From triple to quadruple aim: care of the patient requires care of the provider. *The Annals of Family Medicine*, Annals Family Med, v. 12, n. 6, p. 573–576, 2014. Cited in page 20.

BOGARÍN, A.; CEREZO, R.; ROMERO, C. A survey on educational process mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, Wiley Online Library, v. 8, n. 1, p. e1230, 2018. Cited 2 times in pages 19 and 112.

BOGARÍN, A. et al. Clustering for improving educational process mining. In: *Proceedings of the fourth international conference on learning analytics and knowledge*. [S.l.: s.n.], 2014. p. 11–15. Cited in page 30.

BOLT, A. et al. Exploiting process cubes, analytic workflows and process mining for business process reporting: A case study in education. *SIMPDA*, v. 1527, p. 33–47, 2015. Cited in page 112.

BOLT, A. et al. Business process reporting using process mining, analytic workflows and process cubes: a case study in education. In: SPRINGER. *International symposium on data-driven process discovery and analysis*. [S.l.], 2015. p. 28–53. Cited in page 112.

BONEAU, C. A. The effects of violations of assumptions underlying the t test. *Psychological Bulletin*, American Psychological Association, v. 57, n. 1, p. 49, 1960. Cited 2 times in pages 98 and 116.

BORREGO, D.; BARBA, I. Conformance checking and diagnosis for declarative business process models in data-aware scenarios. *Expert Systems with Applications*, Elsevier, v. 41, n. 11, p. 5340–5352, 2014. Cited in page 68.

BOSE, R. J. C.; AALST, W. M. Van der. Abstractions in process mining: A taxonomy of patterns. In: SPRINGER. *International Conference on Business Process Management*. [S.l.], 2009. p. 159–175. Cited in page 174.

BOSE, R. J. C.; MANS, R.; AALST, W. M. van der. Wanna improve process mining results?: it's high time we consider data quality issues seriously. *BPM reports*, BPMcenter. org, v. 1302, 2013. Cited 2 times in pages 179 and 180.

BOSE, R. J. C.; MANS, R. S.; AALST, W. M. van der. Wanna improve process mining results? In: IEEE. *2013 IEEE symposium on computational intelligence and data mining (CIDM)*. [S.l.], 2013. p. 127–134. Cited in page 161.

BRASIL. Lei nº 12.711, de 29 de agosto de 2012. dispõe sobre o ingresso nas universidades federais e nas instituições federais de ensino técnico de nível médio e dá outras providências. *Diário Oficial da República Federativa do Brasil*, Brasília, DF, 2012. ISSN 1677-7042. Available from Internet: <http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2012/lei/l12711.htm>. Cited in page 114.

BRASIL; SAÚDE, M. da. *SIGTAP-Sistema de Gerenciamento da Tabela de Procedimentos, Medicamentos e OPM do SUS*. Ministério da Saúde Brasília (DF), 2023. Available from Internet: <http://sigtap.datasus.gov.br/>. Cited 2 times in pages 130 and 132.

BRISKIEWICZ, L. B. *Identificação dos gastos dos cursos de graduação da Universidade Tecnológica Federal do Paraná Câmpus Pato Branco e mensuração do custo ideal por aluno*. Dissertação (B.S. thesis) — Universidade Tecnológica Federal do Paraná, 2016. Cited in page 115.

BROOKE, J. et al. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, London, England, v. 189, n. 194, p. 4–7, 1996. Cited in page 77.

BRZYCHCZY, E.; TRZCIONKOWSKA, A. Creation of an event log from a low-level machinery monitoring system for process mining purposes. In: SPRINGER. *International Conference on Intelligent Data Engineering and Automated Learning*. [S.l.], 2018. p. 54–63. Cited in page 173.

BRZYCHCZY, E.; TRZCIONKOWSKA, A. Process-oriented approach for analysis of sensor data from longwall monitoring system. In: SPRINGER. *International Conference on Intelligent Systems in Production Engineering and Maintenance*. [S.l.], 2018. p. 611–621. Cited in page 176.

BURATTIN, A.; CONTI, M.; TURATO, D. Toward an anonymous process mining. In: IEEE. *2015 3rd International Conference on Future Internet of Things and Cloud*. [S.l.], 2015. p. 58–63. Cited in page 181.

CAIRNS, A. H. et al. Towards custom-designed professional training contents and curriculums through educational process mining. In: *The fourth international conference on advances in information mining and management*. [S.l.: s.n.], 2014. p. 53–58. Cited in page 112.

CALVANESE, D. et al. Obda for log extraction in process mining. In: SPRINGER. *Reasoning Web International Summer School*. [S.l.], 2017. p. 292–345. Cited in page 172.

CALVANESE, D. et al. Ontology-based data access for extracting event logs from legacy data: the onprom tool and methodology. In: SPRINGER. *International Conference on Business Information Systems*. [S.l.], 2017. p. 220–236. Cited in page 172.

CALVANESE, D. et al. Ontology-driven extraction of event logs from relational databases. In: SPRINGER. *International Conference on Business Process Management*. [S.l.], 2016. p. 140–153. Cited 2 times in pages 170 and 172.

CAO, Y. et al. Prediction of medical expenses for gastric cancer based on process mining. *Concurrency and Computation: Practice and Experience*, Wiley Online Library, v. 33, n. 15, p. e5694, 2021. Cited 3 times in pages 22, 23, and 24.

CARRASQUEL, J.; CHUBUROV, S.; LOMAZOVA, I. Pre-processing network messages of trading systems into event logs for process mining. In: _____. [S.l.: s.n.], 2021. p. 88–100. ISBN 978-3-030-71471-0. Cited in page 174.

CERDEIRA, J. M. et al. Predictors of student success in higher education: Secondary school internal scores versus national exams. *Higher Education Quarterly*, Wiley Online Library, v. 72, n. 4, p. 304–313, 2018. Cited in page 112.

CEREZO, R. et al. Process mining for self-regulated learning assessment in e-learning. *Journal of Computing in Higher Education*, Springer, v. 32, n. 1, p. 74–88, 2020. Cited in page 112.

CHARTRAND, G. *Introductory graph theory*. [S.l.]: Courier Corporation, 1977. Cited in page 34.

CHENG, H.-J.; KUMAR, A. Process mining on noisy logs—can log sanitization help to improve performance? *Decision Support Systems*, Elsevier, v. 79, p. 138–149, 2015. Cited in page 176.

CHIUDINELLI, L. et al. Mining post-surgical care processes in breast cancer patients. *Artificial Intelligence in Medicine*, Elsevier, p. 101855, 2020. Cited in page 161.

CHYUNG, S. Y.; BARKIN, J. R.; SHAMSY, J. A. Evidence-based survey design: The use of negatively worded items in surveys. *Performance Improvement*, Wiley Online Library, v. 57, n. 3, p. 16–25, 2018. Cited in page 78.

CNPQ. *Programa Mai/Dai*. 2023. Access date: 28 dec. 2022. Available from Internet: <https://www.gov.br/cnpq/pt-br/acesso-a-informacao/acoes-e-programas/programas/programa-mai-dai>. Cited in page 27.

COMBI, C. et al. Seamless conceptual modeling of processes with transactional and analytical data. *Data & Knowledge Engineering*, Elsevier, v. 134, p. 101895, 2021. Cited in page 173.

CONFORTI, R. et al. Automatic repair of same-timestamp errors in business process event logs. In: SPRINGER. *International Conference on Business Process Management*. [S.l.], 2020. p. 327–345. Cited in page 179.

CONFORTI, R.; ROSA, M. L.; HOFSTEDE, A. H. ter. Filtering out infrequent behavior from business process event logs. *IEEE Transactions on Knowledge and Data Engineering*, IEEE, v. 29, n. 2, p. 300–314, 2016. Cited in page 177.

CONFORTI, R.; ROSA, M. L.; HOFSTEDE, A. T. Timestamp repair for business process event logs. 2018. Cited in page 179.

CONFORTI, R.; ROSA, M. L.; HOFSTEDE, A. ter. Noise filtering of process execution logs based on outliers detection. 2015. Cited in page 177.

CORDER, G. W.; FOREMAN, D. I. *Nonparametric statistics: A step-by-step approach*. [S.l.]: John Wiley & Sons, 2014. Cited 2 times in pages 98 and 116.

CRUCHTEN, R. R. van; WEIGAND, H. H. Process mining in logistics: The need for rule-based data abstraction. In: IEEE. *2018 12th International Conference on Research Challenges in Information Science (RCIS)*. [S.l.], 2018. p. 1–9. Cited in page 176.

DAKIC, D. et al. Business process mining application: A literature review. *Annals of DAAAM & Proceedings*, v. 29, 2018. Cited in page 19.

DAKIC, D. et al. Event log extraction for the purpose of process mining: A systematic literature review. In: *Innovation in Sustainable Management and Entrepreneurship*. [S.l.: s.n.], 2019. p. 126–136. Cited in page 164.

DEOKAR, A. V.; TAO, J. Semantics-based event log aggregation for process mining and analytics. *Information Systems Frontiers*, Springer, v. 17, n. 6, p. 1209–1226, 2015. Cited in page 175.

DIAMANTINI, C. et al. Discovering mobility patterns of instagram users through process mining techniques. In: IEEE. *2017 IEEE International Conference on Information Reuse and Integration (IRI)*. [S.l.], 2017. p. 485–492. Cited in page 172.

DIŠEK, M.; ŠPERKA, R.; KOLESÁR, J. Conversion of real data from production process of automotive company for process mining analysis. In: SPRINGER. *KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications*. [S.l.], 2017. p. 223–233. Cited in page 172.

DIXIT, P. M. et al. Detection and interactive repair of event ordering imperfection in process logs. In: SPRINGER. *International Conference on Advanced Information Systems Engineering*. [S.l.], 2018. p. 274–290. Cited in page 178.

DOGAN, O.; FERNANDEZ-LLATAS, C.; OZTAYSI, B. Process mining application for analysis of customer's different visits in a shopping mall. In: SPRINGER. *International Conference on Intelligent and Fuzzy Systems*. [S.l.], 2019. p. 151–159. Cited in page 30.

DONGEN, B. F. V.; MEDEIROS, A. A. D.; WEN, L. Process mining: Overview and outlook of petri net discovery algorithms. *transactions on petri nets and other models of concurrency II*, Springer, p. 225–242, 2009. Cited in page 39.

DONGEN, B. van. *BPI Challenge 2019*. 2019. Available from Internet: <https://doi.org/10.4121/uuid:d06aff4b-79f0-45e6-8ec8-e19730c248f1>. Cited in page 177.

DUNZER, S. et al. Conformance checking: a state-of-the-art literature review. In: *Proceedings of the 11th international conference on subject-oriented business process management*. [S.l.: s.n.], 2019. p. 1–10. Cited in page 40.

ECK, M. L. v. et al. Pm²: a process mining project methodology. In: SPRINGER. *International conference on advanced information systems engineering*. [S.l.], 2015. p. 297–313. Cited in page 113.

ECK, M. L. V.; SIDOROVA, N.; AALST, W. M. Van der. Enabling process mining on sensor data from smart products. In: IEEE. *2016 IEEE Tenth International Conference on Research Challenges in Information Science (RCIS)*. [S.l.], 2016. p. 1–12. Cited 2 times in pages 170 and 172.

EIGLSPERGER, M.; SIEBENHALLER, M.; KAUFMANN, M. An efficient implementation of sugiyama's algorithm for layered graph drawing. In: SPRINGER. *International Symposium on Graph Drawing*. [S.l.], 2004. p. 155–166. Cited in page 196.

EILI, M. Y.; REZAEENOUR, J.; SANI, M. F. A systematic literature review on process-aware recommender systems. *arXiv preprint arXiv:2103.16654*, 2021. Cited in page 57.

EMAMJOME, F. et al. Alohomora: Unlocking data quality causes through event log context. In: ASSOCIATION FOR INFORMATION SYSTEMS. *Proceedings of the 28th European Conference on Information Systems (ECIS2020)*. [S.l.], 2020. p. 1–16. Cited in page 181.

ENGEL, R. et al. Mining inter-organizational business process models from edi messages: A case study from the automotive sector. In: SPRINGER. *International Conference on Advanced Information Systems Engineering*. [S.l.], 2012. p. 222–237. Cited 2 times in pages 171 and 172.

ENGEL, R. et al. Ediminer: A toolset for process mining from edi messages. In: CITESEER. *CAiSE Forum*. [S.l.], 2013. p. 146–153. Cited 2 times in pages 171 and 172.

ENGEL, R. et al. Towards edi-based business activity monitoring. In: IEEE. *2013 17th IEEE International Enterprise Distributed Object Computing Conference Workshops*. [S.l.], 2013. p. 158–162. Cited in page 171.

ENGEL, R. et al. Process mining for electronic data interchange. In: SPRINGER. *International Conference on Electronic Commerce and Web Technologies*. [S.l.], 2011. p. 77–88. Cited 2 times in pages 170 and 171.

ENGEL, R. et al. Analyzing inter-organizational business processes. *Information Systems and e-Business Management*, Springer, v. 14, n. 3, p. 577–612, 2016. Cited in page 172.

ENGELEN, K. K.; BAKKERS, F.; ENERGIEVERLENING, P. *Towards Improved Decision-Making through the Integration of Financial Information in BPMN Models*. [S.l.]: Eindhoven University of Technology, 2015. Cited 3 times in pages 22, 23, and 24.

ERDEM, S.; DEMIRÖRS, O.; RABHI, F. Systematic mapping study on process mining in agile software development. In: SPRINGER. *International Conference on Software Process Improvement and Capability Determination*. [S.l.], 2018. p. 289–299. Cited in page 163.

ERDOGAN, T. G.; TARHAN, A. Systematic mapping of process mining studies in healthcare. *IEEE Access*, IEEE, v. 6, p. 24543–24567, 2018. Cited in page 163.

ESTER, M. et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In: *kdd*. [S.l.: s.n.], 1996. v. 96, n. 34, p. 226–231. Cited in page 46.

FAHLAND, D.; AALST, W. M. van D. Model repair—aligning process models to reality. *Information Systems*, Elsevier, v. 47, p. 220–243, 2015. Cited in page 44.

FAHRENKROG-PETERSEN, S. A.; AA, H. V. D.; WEIDLICH, M. Pretsa: event log sanitization for privacy-aware process discovery. In: IEEE. *2019 International Conference on Process Mining (ICPM)*. [S.l.], 2019. p. 1–8. Cited in page 181.

FERRONATO, J. J. et al. Analyzing process mining for short-term simulation in operational decision making: a systematic literature review. in press. Cited in page 185.

FISCHER, D. A. et al. Enhancing event log quality: detecting and quantifying timestamp imperfections. In: SPRINGER. *International Conference on Business Process Management*. [S.l.], 2020. p. 309–326. Cited 2 times in pages 179 and 181.

FLUXICON. *Disco Process Mining*. 2023. Available from Internet: <https://fluxicon.com/book/read/reference/>. Cited in page 91.

FOLINO, F.; PONTIERI, L. Pushing more ai capabilities into process mining to better deal with low-quality logs. In: SPRINGER. *International Conference on Business Process Management*. [S.l.], 2019. p. 5–11. Cited in page 179.

FOX, F. et al. A data quality framework for process mining of electronic health record data. In: IEEE. *2018 IEEE International Conference on Healthcare Informatics (ICHI)*. [S.l.], 2018. p. 12–21. Cited 2 times in pages 179 and 180.

FRANCESCOMARINO, C. D. et al. Predictive process monitoring methods: Which one suits me best? In: SPRINGER. *International Conference on Business Process Management*. [S.l.], 2018. p. 462–479. Cited 2 times in pages 54 and 56.

FRANCISCO, C. E. d. S. et al. Espacialização de análise multicriterial em sig: prioridades para recuperação de áreas de preservação permanente. *Simpósio Brasileiro de Sensoriamento Remoto*, v. 13, p. 2643–2650, 2007. Cited in page 137.

FRANK, E.; HALL, M.; WITTEN, I. *The WEKA Workbench*. Morgan Kaufmann, 2016. Available from Internet: <https://books.google.com.br/books?id=4-FZuwEACAAJ>. Cited in page 190.

GARCIA, C. d. S. et al. Process mining techniques and applications – a systematic mapping study. *Expert Systems with Applications*, Elsevier, v. 133, p. 260–295, 2019. Cited 6 times in pages 19, 39, 57, 112, 161, and 163.

GHASEMI, M.; AMYOT, D. Data preprocessing for goal-oriented process discovery. In: IEEE. *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*. [S.l.], 2019. p. 200–206. Cited in page 161.

GHASEMI, M.; AMYOT, D. From event logs to goals: a systematic literature review of goal-oriented process mining. *Requirements Engineering*, v. 25, 2020. Cited in page 163.

GHAZAL, M. A.; IBRAHIM, O.; SALAMA, M. A. Educational process mining: a systematic literature review. In: IEEE. *2017 European Conference on Electrical Engineering and Computer Science (EECS)*. [S.l.], 2017. p. 198–203. Cited 2 times in pages 112 and 162.

GHIONNA, L. et al. Outlier detection techniques for process mining applications. In: SPRINGER. *International symposium on methodologies for intelligent systems*. [S.l.], 2008. p. 150–159. Cited in page 176.

GOTTLIEB, A. et al. A method for inferring medical diagnoses from patient similarities. *BMC medicine*, BioMed Central, v. 11, n. 1, p. 1–10, 2013. Cited in page 192.

GREEN, D. *What Is Quality in Higher Education?*. [S.l.]: ERIC, 1994. Cited in page 20.

GRIGOROVA, K.; MALYSHEVA, E.; BOBROVSKIY, S. Application of data mining and process mining approaches for improving e-learning processes. In: *3rd International Conference on Information Technology and Nanotechnology*. [S.l.: s.n.], 2017. p. 25–27. Cited in page 112.

GÜNTHER, C.; VERBEEK, H. Xes standard definition. www. xes-standard. org (2009). *Cited on*, v. 72, 2009. Cited 3 times in pages 37, 38, and 63.

GÜNTHER, C. W.; AALST, W. M. van der. A generic import framework for process event logs. In: SPRINGER. *International Conference on Business Process Management*. [S.l.], 2006. p. 81–92. Cited 2 times in pages 170 and 171.

GÜNTHER, C. W.; ROZINAT, A.; AALST, W. M. V. D. Activity mining by global trace segmentation. In: SPRINGER. *International Conference on Business Process Management*. [S.l.], 2009. p. 128–139. Cited in page 174.

GUPTA, M.; SEREBRENIK, A.; JALOTE, P. Improving software maintenance using process mining and predictive analytics. In: IEEE. *2017 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. [S.l.], 2017. p. 681–686. Cited in page 19.

HAND, D. J. Principles of data mining. *Drug safety*, Springer, v. 30, n. 7, p. 621–622, 2007. Cited in page 55.

HÄRDLE, W. et al. *Nonparametric and semiparametric models*. [S.l.]: Springer, 2004. v. 1. Cited 2 times in pages 97 and 118.

HEE, K. M. van; LIU, Z.; SIDOROVA, N. Is my event log complete?—a probabilistic approach to process mining. In: IEEE. *2011 FIFTH INTERNATIONAL CONFERENCE ON RESEARCH CHALLENGES IN INFORMATION SCIENCE*. [S.l.], 2011. p. 1–12. Cited in page 180.

HERAVIZADEH, M.; MENDLING, J.; ROSEMANN, M. Root cause analysis in business processes. 2008. Cited in page 105.

HERNANDEZ, S. et al. Analysis of users' behavior in structured e-commerce websites. *IEEE Access*, IEEE, v. 5, p. 11941–11958, 2017. Cited in page 173.

HIRSCH, R. *Exploring colour photography: a complete guide*. [S.l.]: Laurence King Publishing, 2004. Cited in page 91.

HOFSTEDE, A. H. T. et al. *Modern Business Process Automation: YAWL and its support environment*. [S.l.]: Springer Science & Business Media, 2009. Cited in page 31.

HONG, T. T. B. Process mining-driven performance analysis in manufacturing process: Cost and quality perspective. Graduate School of UNIST, 2016. Cited 4 times in pages 20, 22, 23, and 24.

HOUDT, G. V.; DEPAIRE, B.; MARTIN, N. Root cause analysis in process mining with probabilistic temporal logic. In: SPRINGER. *International Conference on Process Mining*. [S.l.], 2022. p. 73–84. Cited in page 105.

HUANG, H.; JIN, T.; WANG, J. Extracting clinical-event-packages from billing data for clinical pathway mining. In: SPRINGER. *International Conference on Smart Health*. [S.l.], 2016. p. 19–31. Cited in page 172.

HUANG, Z. et al. On mining latent treatment patterns from electronic medical records. *Data mining and knowledge discovery*, Springer, v. 29, n. 4, p. 914–949, 2015. Cited 2 times in pages 170 and 171.

HUGHES, M. D.; BARTLETT, R. M. The use of performance indicators in performance analysis. *Journal of sports sciences*, Taylor & Francis, v. 20, n. 10, p. 739–754, 2002. Cited in page 20.

HWANG, I.; JANG, Y. J. Process mining to discover shoppers' pathways at a fashion retail store using a wifi-base indoor positioning system. *IEEE Transactions on Automation Science and Engineering*, IEEE, v. 14, n. 4, p. 1786–1792, 2017. Cited 2 times in pages 161 and 172.

IBGE. *IPCA - Índice Nacional de Preços ao Consumidor Amplo*. 2022. Available from Internet: <https://www.ibge.gov.br/estatisticas/economicas/precos-e-custos/9256-indice-nacional-de-precos-ao-consumidor-amplo.html?=&t=resultados>. Cited in page 115.

IBGE. *ProM Tools*. 2023. Available from Internet: <https://promtools.org/>. Cited in page 91.

IEEE. IEEE 1849-2016 XES Standard. 1999. Available from Internet: <http://xes-standard.org/>. Cited 2 times in pages 63 and 64.

IEEE. Ieee standard for extensible event stream (xes) for achieving interoperability in event logs and event streams. *IEEE Std 1849-2016*, p. 1–50, 2016. Cited in page 37.

IGLESIAS, J. A. et al. Creating evolving user behavior profiles automatically. *IEEE Transactions on Knowledge and Data Engineering*, IEEE, v. 24, n. 5, p. 854–867, 2011. Cited in page 161.

INGVALDSEN, J. E.; GULLA, J. A. Preprocessing support for large scale process mining of sap transactions. In: SPRINGER. *International Conference on Business process management*. [S.l.], 2007. p. 30–41. Cited 2 times in pages 170 and 171.

INTAYOAD, W.; KAMYOD, C.; TEMDEE, P. Process mining application for discovering student learning paths. In: IEEE. *2018 International Conference on Digital Arts, Media and Technology (ICDAMT)*. [S.l.], 2018. p. 220–224. Cited in page 112.

ISO 9241-11. *Ergonomics of human-system interaction - Part 11: Usability: Definitions and Concepts*. 2018. Cited in page 77.

JAMOLIDDINOVICH, U. B. Fundamentals of education quality in higher education. *INTERNATIONAL JOURNAL OF SOCIAL SCIENCE & INTERDISCIPLINARY RESEARCH ISSN: 2277-3630 Impact factor: 7.429*, v. 11, n. 01, p. 149–151, 2022. Cited in page 20.

JANS, M. et al. A business process mining application for internal transaction fraud mitigation. *Expert Systems with Applications*, Elsevier, v. 38, n. 10, p. 13351–13359, 2011. Cited in page 161.

JENSEN, K.; KRISTENSEN, L. M. *Coloured Petri nets: modelling and validation of concurrent systems*. [S.l.]: Springer Science & Business Media, 2009. Cited 2 times in pages 31 and 33.

JLAILATY, D.; GRIGORI, D.; BELHAJJAME, K. A framework for mining process models from emails logs. *arXiv preprint arXiv:1609.06127*, 2016. Cited 2 times in pages 170 and 172.

JLAILATY, D.; GRIGORI, D.; BELHAJJAME, K. Business process instances discovery from email logs. In: IEEE. *2017 IEEE International Conference on Services Computing (SCC)*. [S.l.], 2017. p. 19–26. Cited in page 172.

JLAILATY, D.; GRIGORI, D.; BELHAJJAME, K. Mining business process activities from email logs. In: IEEE. *2017 IEEE International Conference on Cognitive Computing (ICCC)*. [S.l.], 2017. p. 112–119. Cited in page 172.

JLAILATY, D.; GRIGORI, D.; BELHAJJAME, K. Email business activities extraction and annotation. In: SPRINGER. *International Workshop on Information Search, Integration, and Personalization*. [S.l.], 2018. p. 69–86. Cited in page 173.

JUHAŇÁK, L.; ZOUNEK, J.; ROHLÍKOVÁ, L. Using process mining to analyze students' quiz-taking behavior patterns in a learning management system. *Computers in Human Behavior*, Elsevier, v. 92, p. 496–506, 2019. Cited in page 112.

KABICHER, S.; RINDERLE-MA, S. Human-centered process engineering based on content analysis and process view aggregation. In: SPRINGER. *International Conference on Advanced Information Systems Engineering*. [S.l.], 2011. p. 467–481. Cited 2 times in pages 170 and 171.

KAPLAN, R. S.; ANDERSON, S. R. *Time-driven activity-based costing: a simpler and more powerful path to higher profits*. [S.l.]: Harvard business press, 2007. Cited in page 61.

KAPLAN, R. S.; ATKINSON, A. A. *Advanced management accounting*. [S.l.]: PHI Learning, 1998. Cited 4 times in pages 59, 60, 61, and 62.

KELLER, R. M. Formal verification of parallel programs. *Communications of the ACM*, ACM New York, NY, USA, v. 19, n. 7, p. 371–384, 1976. Cited in page 35.

KHERBOUCHE, M. O.; LAGA, N.; MASSE, P.-A. Towards a better assessment of event logs quality. In: IEEE. *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*. [S.l.], 2016. p. 1–8. Cited in page 180.

KHOVRICHEV, M. et al. Intelligent approach for heterogeneous data integration: Information processes analysis engine in clinical remote monitoring systems. *Procedia Computer Science*, Elsevier, v. 156, p. 134–141, 2019. Cited in page 173.

KIREMIRE, A. R. The application of the pareto principle in software engineering. *Consulted January*, v. 13, p. 2016, 2011. Cited in page 39.

KITCHENHAM, B. A.; BUDGEN, D.; BRERETON, P. *Evidence-based software engineering and systematic reviews*. [S.l.]: CRC press, 2015. v. 4. Cited 2 times in pages 164 and 167.

KNOLL, D.; WALDMANN, J.; REINHART, G. Developing an internal logistics ontology for process mining. *Procedia CIRP*, Elsevier, v. 79, p. 427–432, 2019. Cited in page 173.

KRAJSIC, P.; FRANCZYK, B. Lambda architecture for anomaly detection in online process mining using autoencoders. In: SPRINGER. *International Conference on Computational Collective Intelligence*. [S.l.], 2020. p. 579–589. Cited in page 177.

KRAJSIC, P.; FRANCZYK, B. Semi-supervised anomaly detection in business process event data using self-attention based classification. *Procedia Computer Science*, Elsevier, v. 192, p. 39–48, 2021. Cited in page 177.

KRATHU, W. et al. A framework for inter-organizational performance analysis from edi messages. In: IEEE. *2014 IEEE 16th Conference on Business Informatics*. [S.l.], 2014. v. 1, p. 17–24. Cited in page 161.

KURNIATI, A. P. et al. The assessment of data quality issues for process mining in healthcare using medical information mart for intensive care iii, a freely available e-health record database. *Health informatics journal*, SAGE Publications Sage UK: London, England, v. 25, n. 4, p. 1878–1893, 2019. Cited in page 180.

LAROSE, D. T.; LAROSE, C. D. *Discovering knowledge in data: an introduction to data mining*. [S.l.]: John Wiley & Sons, 2014. v. 4. Cited in page 106.

LASI, H. et al. Industry 4.0. *Business & information systems engineering*, Springer, v. 6, n. 4, p. 239–242, 2014. Cited in page 19.

LEE, S. W.; RINE, D. C. Case study methodology designed research in software engineering methodology validation. In: *SEKE*. [S.l.: s.n.], 2004. p. 117–122. Cited in page 77.

LEEMANS, S. J. et al. Stochastic process mining: Earth movers' stochastic conformance. *Information Systems*, Elsevier, p. 101724, 2021. Cited in page 40.

LEEMANS, S. J.; FAHLAND, D.; AALST, W. M. van der. Discovering block-structured process models from event logs containing infrequent behaviour. In: SPRINGER. *International conference on business process management*. [S.l.], 2013. p. 66–78. Cited in page 39.

LEEMANS, S. J. et al. Process mining for healthcare decision analytics with micro-costing estimations. *Artificial Intelligence in Medicine*, Elsevier, p. 102473, 2022. Cited 4 times in pages 20, 22, 23, and 24.

LEHTO, T.; HINKKA, M.; HOLLMÉN, J. Focusing business improvements using process mining based influence analysis. In: SPRINGER. *International Conference on Business Process Management*. [S.l.], 2016. p. 177–192. Cited 2 times in pages 105 and 106.

LEONARDI, G. et al. Leveraging semantic labels for multi-level abstraction in medical process mining and trace comparison. *Journal of biomedical informatics*, Elsevier, v. 83, p. 10–24, 2018. Cited in page 175.

LEONI, M. D.; AALST, W. M. V. D. Aligning event logs and process models for multi-perspective conformance checking: An approach based on integer linear programming. In: *Business process management*. [S.l.]: Springer, 2013. p. 113–129. Cited in page 68.

LEONI, M. D.; AALST, W. M. V. D.; DONGEN, B. F. V. Data-and resource-aware conformance checking of business processes. In: SPRINGER. *International Conference on Business Information Systems*. [S.l.], 2012. p. 48–59. Cited in page 68.

LEONI, M. D.; MAGGI, F. M.; AALST, W. M. van der. An alignment-based framework to check the conformance of declarative process models and to preprocess event-log data. *Information Systems*, Elsevier, v. 47, p. 258–277, 2015. Cited in page 176.

LIKERT, R. A technique for the measurement of attitudes. *Archives of psychology*, 1932. Cited in page 78.

LIU, C. Automatic discovery of behavioral models from software execution data. *IEEE Transactions on Automation Science and Engineering*, IEEE, v. 15, n. 4, p. 1897–1908, 2018. Cited in page 173.

LIU, C. et al. Proactive workflow modeling by stochastic processes with application to healthcare operation and management. In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. [S.l.: s.n.], 2014. p. 1593–1602. Cited 2 times in pages 170 and 171.

LIU, C. et al. Logrank: An approach to sample business process event log for efficient discovery. In: SPRINGER. *International Conference on Knowledge Science, Engineering and Management*. [S.l.], 2018. p. 415–425. Cited in page 177.

LORENZ, R. et al. Using process mining to improve productivity in make-to-stock manufacturing. *International Journal of Production Research*, Taylor & Francis, v. 59, n. 16, p. 4869–4880, 2021. Cited in page 19.

LOW, W. Z. *Towards cost model-driven log-based business process improvement*. Tese (Doutorado) — Queensland University of Technology, 2016. Cited 3 times in pages 22, 23, and 24.

LOW, W. Z. et al. Revising history for cost-informed process improvement. *Computing*, Springer, v. 98, n. 9, p. 895–921, 2016. Cited 3 times in pages 22, 23, and 24.

LU, X.; FAHLAND, D. A conceptual framework for understanding event data quality for behavior analysis. In: *ZEUS*. [S.l.: s.n.], 2017. p. 11–14. Cited in page 180.

LU, X. et al. Handling duplicated tasks in process discovery by refining event labels. In: SPRINGER. *International Conference on Business Process Management*. [S.l.], 2016. p. 90–107. Cited in page 175.

LU, X. et al. Discovering interacting artifacts from erp systems. *IEEE Transactions on Services Computing*, IEEE, v. 8, n. 6, p. 861–873, 2015. Cited in page 171.

MADAKAM, S. et al. Internet of things (iot): A literature review. *Journal of Computer and Communications*, Scientific Research Publishing, v. 3, n. 05, p. 164, 2015. Cited in page 19.

MAHMOOD, T.; SHAIKH, G. M. Adaptive automated teller machines. *Expert Systems with Applications*, Elsevier, v. 40, n. 4, p. 1152–1169, 2013. Cited in page 161.

MAIMON, O. Z.; ROKACH, L. *Data mining with decision trees: theory and applications*. [S.l.]: World scientific, 2014. v. 81. Cited 2 times in pages 98 and 116.

MANNHARDT, F. et al. Privacy-preserving process mining. *Business & Information Systems Engineering*, Springer, v. 61, n. 5, p. 595–614, 2019. Cited in page 181.

MANNHARDT, F. et al. Balanced multi-perspective checking of process conformance. *Computing*, Springer, v. 98, n. 4, p. 407–437, 2016. Cited in page 71.

MANNHARDT, F. et al. From low-level events to activities-a pattern-based approach. In: SPRINGER. *International conference on business process management*. [S.l.], 2016. p. 125–141. Cited in page 175.

MANNHARDT, F.; PETERSEN, S. A.; OLIVEIRA, M. F. Privacy challenges for process mining in human-centered industrial environments. In: IEEE. *2018 14th International Conference on Intelligent Environments (IE)*. [S.l.], 2018. p. 64–71. Cited in page 181.

MANS, R. S.; AALST, W. M. Van der; VANWERSCH, R. J. *Process mining in healthcare: evaluating and exploiting operational healthcare processes*. [S.l.]: Springer, 2015. Cited 4 times in pages 32, 34, 37, and 54.

MANS, R. S. et al. Process mining in healthcare: Data challenges when answering frequently posed questions. In: *Process Support and Knowledge Representation in Health Care*. [S.l.]: Springer, 2012. p. 140–153. Cited in page 30.

MEDEIROS, A. K. A. D. et al. Process mining based on clustering: A quest for precision. In: SPRINGER. *International Conference on Business Process Management*. [S.l.], 2007. p. 17–29. Cited in page 161.

MEDEIROS, A. K. A. de; WEIJTERS, A. J.; AALST, W. M. van der. Genetic process mining: an experimental evaluation. *Data Mining and Knowledge Discovery*, Springer, v. 14, n. 2, p. 245–304, 2007. Cited in page 39.

MEDEIROS, R. W. de; ROSA, N. S.; PIRES, L. F. A metamodel for modeling cost behavior in service composition. In: IEEE. *2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA)*. [S.l.], 2014. p. 84–91. Cited 3 times in pages 21, 22, and 24.

METSKER, O. et al. Pattern-based mining in electronic health records for complex clinical process analysis. *Procedia computer science*, Elsevier, v. 119, p. 197–206, 2017. Cited in page 172.

MODEL, O. B. P. Notation (bpmn). object management group, dtc. 2010. Cited 2 times in pages 31 and 33.

MOORE, G. E. et al. *Cramming more components onto integrated circuits*. [S.l.]: McGraw-Hill New York, NY, USA:, 1965. Cited in page 30.

MOSTAFAEE, D. K. et al. Mining process evaluation in discovering the semi-automatic processes of the banking industry (the case: Bank guarantee issuance process). Journal of Industrial Management Studies, 2019. Cited in page 30.

MUELLER-WICKOP, N.; SCHULTZ, M. Erp event log preprocessing: timestamps vs. accounting logic. In: SPRINGER. *International Conference on Design Science Research in Information Systems*. [S.l.], 2013. p. 105–119. Cited in page 171.

MUKALA, P. et al. Learning analytics on coursera event data: A process mining approach. In: *SIMPDA*. [S.l.: s.n.], 2015. p. 18–32. Cited in page 161.

MURILLAS, E. G. L. de; AALST, W. M. van der; REIJERS, H. A. Process mining on databases: Unearthing historical data from redo logs. In: SPRINGER. *International Conference on Business Process Management*. [S.l.], 2016. p. 367–385. Cited in page 172.

MURILLAS, E. G. L. de; REIJERS, H. A.; AALST, W. M. V. D. Connecting databases with process mining: A meta model and toolset. In: *BMMDS/EMMSAD*. [S.l.: s.n.], 2016. p. 231–249. Cited in page 173.

MYERS, D. et al. Anomaly detection for industrial control systems using process mining. *Computers & Security*, Elsevier, v. 78, p. 103–125, 2018. Cited in page 53.

NADERIFAR, V.; SAHRAN, S.; SHUKUR, Z. A review on conformance checking technique for the evaluation of process mining algorithms. *TEM Journal*, UIKTEN-Association for Information Communication Technology Education and . . . , v. 8, n. 4, p. 1232, 2019. Cited in page 40.

NAKATUMBA, J.; AALST, W. M. van der. Analyzing resource behavior using process mining. In: SPRINGER. *International Conference on Business Process Management*. [S.l.], 2009. p. 69–80. Cited in page 105.

NAUTA, W. Towards cost-awareness in process mining. *Eindhoven University of Technology*, 2011. Cited 8 times in pages 21, 22, 24, 61, 65, 66, 67, and 88.

NEUMANN, H. et al. Data quality assessment to apply process mining in production processes. In: SPRINGER. *Congress of the German Academic Association for Production Technology*. [S.l.], 2021. p. 515–524. Cited in page 181.

NGUYEN, H. T. C. et al. Autoencoders for improving quality of process event logs. *Expert Systems with Applications*, Elsevier, v. 131, p. 132–147, 2019. Cited in page 179.

NOLLE, T. et al. Analyzing business process anomalies using autoencoders. *Machine Learning*, Springer, v. 107, n. 11, p. 1875–1893, 2018. Cited in page 177.

NOLLE, T.; SEELIGER, A.; MÜHLHÄUSER, M. Unsupervised anomaly detection in noisy business process event logs using denoising autoencoders. In: SPRINGER. *International conference on discovery science*. [S.l.], 2016. p. 442–456. Cited in page 177.

NOROUZI, M.; FLEET, D. J.; SALAKHUTDINOV, R. R. Hamming distance metric learning. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2012. p. 1061–1069. Cited in page 46.

PAJIĆ, A.; BEČEJSKI-VUJAKLIJA, D. Metamodel of the artifact-centric approach to event log extraction from erp systems. *International Journal of Decision Support System Technology (IJDSST)*, IGI Global, v. 8, n. 2, p. 18–28, 2016. Cited 2 times in pages 170 and 171.

PARTINGTON, A. et al. Process mining for clinical processes: a comparative analysis of four australian hospitals. *ACM Transactions on Management Information Systems (TMIS)*, ACM New York, NY, USA, v. 5, n. 4, p. 1–18, 2015. Cited in page 30.

PAURA, L.; ARHIPOVA, I. Cause analysis of students' dropout rate in higher education study program. *Procedia - Social and Behavioral Sciences*, v. 109, p. 1282–1286, 2014. ISSN 1877-0428. 2nd World Conference on Business, Economics and Management. Available from Internet: <https://www.sciencedirect.com/science/article/pii/S1877042813052646>. Cited in page 121.

PEFFERS, K. et al. A design science research methodology for information systems research. *Journal of management information systems*, Taylor & Francis, v. 24, n. 3, p. 45–77, 2007. Cited in page 76.

PÉREZ-CASTILLO, R. et al. Assessing event correlation in non-process-aware information systems. *Software & Systems Modeling*, Springer, v. 13, n. 3, p. 1117–1139, 2014. Cited 2 times in pages 170 and 171.

PETERSEN, K. et al. Systematic mapping studies in software engineering. In: *12th International Conference on Evaluation and Assessment in Software Engineering (EASE) 12*. [S.l.: s.n.], 2008. p. 1–10. Cited 3 times in pages 164, 166, and 168.

PIKA, A. et al. Towards privacy-preserving process mining in healthcare. In: SPRINGER. *International Conference on Business Process Management*. [S.l.], 2019. p. 483–495. Cited 2 times in pages 162 and 181.

PIKA, A. et al. Privacy-preserving process mining in healthcare. *International journal of environmental research and public health*, Multidisciplinary Digital Publishing Institute, v. 17, n. 5, p. 1612, 2020. Cited in page 182.

Plataforma Brasil. *Plataforma Brasil*. 2023. Access date: 28 dec. 2022. Available from Internet: <https://plataformabrasil.saude.gov.br/>. Cited 2 times in pages 113 and 129.

PM group of Fraunhofer FIT. PM4Py. 2021. Available from Internet: <https://pm4py.fit.fraunhofer.de/about-us>. Cited 2 times in pages 91 and 190.

QAFARI, M. S.; AALST, W. M. van der. Feature recommendation for structural equation model discovery in process mining. *Progress in Artificial Intelligence*, Springer, p. 1–25, 2022. Cited in page 105.

QAFARI, M. S.; AALST, W. v. d. Root cause analysis in process mining using structural equation models. In: SPRINGER. *International Conference on Business Process Management*. [S.l.], 2020. p. 155–167. Cited in page 105.

QUINLAN, J. R. Simplifying decision trees. *International journal of man-machine studies*, Elsevier, v. 27, n. 3, p. 221–234, 1987. Cited in page 50.

RAFIEI, M.; WAGNER, M.; AALST, W. M. van der. Tlkc-privacy model for process mining. In: SPRINGER. *International Conference on Research Challenges in Information Science*. [S.l.], 2020. p. 398–416. Cited in page 162.

RAFIEI, M.; WALDTHAUSEN, L. von; AALST, W. M. van der. Ensuring confidentiality in process mining. *SIMPDA*, v. 18, p. 3–17, 2018. Cited in page 181.

RAFIEI, M.; WALDTHAUSEN, L. von; AALST, W. M. van der. Supporting confidentiality in process mining using abstraction and encryption. In: *Data-Driven Process Discovery and Analysis*. [S.l.]: Springer, 2018. p. 101–123. Cited 2 times in pages 162 and 182.

RAGGIO, O. The myth of prometheus: Its survival and metamorphoses up to the eighteenth century. *Journal of the Warburg and Courtauld Institutes*, The University of Chicago Press, v. 21, n. 1-2, p. 44–62, 1958. Cited in page 6.

RELIJVELD, S. *Analysis of the cost of care for colorectal cancer patients using Australian multi-centre linked data*. Dissertação (Mestrado) — University of Twente, 2021. Cited 3 times in pages 22, 23, and 24.

RINNER, C. et al. Process mining and conformance checking of long running processes in the context of melanoma surveillance. *International journal of environmental research and public health*, Multidisciplinary Digital Publishing Institute, v. 15, n. 12, p. 2809, 2018. Cited in page 173.

ROGGE-SOLTI, A. et al. Improving documentation by repairing event logs. In: SPRINGER. *IFIP Working Conference on The Practice of Enterprise Modeling*. [S.l.], 2013. p. 129–144. Cited in page 178.

ROGGE-SOLTI, A. et al. *Repairing event logs using stochastic process models*. [S.l.]: Universitätsverlag Potsdam, 2013. v. 78. 705–708 p. Cited in page 178.

ROJAS, E. et al. Process mining in healthcare: A literature review. *Journal of biomedical informatics*, Elsevier, v. 61, p. 224–236, 2016. Cited 2 times in pages 19 and 30.

ROMERO, C. et al. Educational process mining: A tutorial and case study using moodle data sets. *Data mining and learning analytics: Applications in educational research*, Wiley Online Library, p. 1–28, 2016. Cited in page 112.

ROMERO, C. et al. Predicting students' final performance from participation in on-line discussion forums. *Computers & Education*, Elsevier, v. 68, p. 458–472, 2013. Cited in page 112.

ROZINAT, A.; AALST, W. M. Van der. Conformance testing: Measuring the fit and appropriateness of event logs and process models. In: SPRINGER. *International conference on business process management*. [S.l.], 2005. p. 163–176. Cited in page 112.

ROZINAT, A.; AALST, W. M. van der. Decision mining in prom. In: SPRINGER. *International Conference on Business Process Management*. [S.l.], 2006. p. 420–425. Cited 2 times in pages 49 and 105.

ROZINAT, A. et al. Process mining applied to the test process of wafer scanners in asml. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, IEEE, v. 39, n. 4, p. 474–479, 2009. Cited in page 105.

ROZINAT, A. et al. Discovering simulation models. *Information systems*, Elsevier, v. 34, n. 3, p. 305–327, 2009. Cited in page 52.

RUBIN, V. A. et al. Process mining can be applied to software too! In: *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. [S.l.: s.n.], 2014. p. 1–8. Cited in page 161.

SADEGHIANASL, S. et al. A contextual approach to detecting synonymous and polluted activity labels in process event logs. In: SPRINGER. *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. [S.l.], 2019. p. 76–94. Cited 2 times in pages 179 and 180.

SANI, M. F.; ZELST, S. J. v.; AALST, W. M. van der. Repairing outlier behaviour in event logs. In: SPRINGER. *International Conference on Business Information Systems*. [S.l.], 2018. p. 115–131. Cited in page 179.

SANI, M. F.; ZELST, S. J. van; AALST, W. M. van der. Improving process discovery results by filtering outliers using conditional behavioural probabilities. In: SPRINGER. *International Conference on Business Process Management*. [S.l.], 2017. p. 216–229. Cited in page 177.

SANI, M. F.; ZELST, S. J. van; AALST, W. M. van der. Repairing outlier behaviour in event logs using contextual behaviour. *Enterprise Modelling and Information Systems Architectures (EMISAJ)*, v. 14, p. 5–1, 2019. Cited in page 179.

SATO, D. M. V. et al. A survey on concept drift in process mining. *ACM Computing Surveys (CSUR)*, ACM New York, NY, v. 54, n. 9, p. 1–38, 2021. Cited in page 137.

SATO, D. M. V. et al. A survey on concept drift in process mining. *ACM Comput. Surv.*, Association for Computing Machinery, New York, NY, USA, v. 54, n. 9, oct 2021. ISSN 0360-0300. Available from Internet: <https://doi.org/10.1145/3472752>. Cited in page 185.

SCHEER, A.-W. *Business process engineering: reference models for industrial enterprises*. [S.l.]: Springer Science & Business Media, 2012. Cited in page 32.

SCHULTE, J. et al. Large scale predictive process mining and analytics of university degree course data. In: *Proceedings of the Seventh International Learning Analytics & Knowledge Conference*. [S.l.: s.n.], 2017. p. 538–539. Cited in page 112.

SEDGWICK, P. Pearson's correlation coefficient. *Bmj*, British Medical Journal Publishing Group, v. 345, 2012. Cited in page 46.

SIDOROVA, N.; STAHL, C.; TRČKA, N. Soundness verification for conceptual workflow nets with data: Early detection of errors with the most precision possible. *Information Systems*, Elsevier, v. 36, n. 7, p. 1026–1043, 2011. Cited in page 70.

SIGUENZA-GUZMAN, L. et al. Recent evolutions in costing systems: A literature review of time-driven activity-based costing. *Review of Business and Economic Literature*, Intersentia, v. 58, n. 1, p. 34–64, 2013. Cited 4 times in pages 59, 60, 61, and 62.

SIM, S.; BAE, H.; CHOI, Y. Likelihood-based multiple imputation by event chain methodology for repair of imperfect event logs with missing data. In: IEEE. *2019 International Conference on Process Mining (ICPM)*. [S.l.], 2019. p. 9–16. Cited 2 times in pages 178 and 179.

SIMOVIĆ, A. P.; BABAROGIĆ, S.; PANTELIĆ, O. A domain-specific language for supporting event log extraction from erp systems. In: IEEE. *2018 7th International Conference on Computers Communications and Control (ICCCC)*. [S.l.], 2018. p. 12–16. Cited in page 173.

SISU. Sistema de seleção unificada. *Ministério da Educação*, 2022. Available from Internet: <https://acessounico.mec.gov.br/sisu>. Cited in page 114.

SLYWOTZKY, A. J.; MORRISON, D.; WEBER, K. *How digital is your business?* [S.l.]: Currency, 2001. Cited in page 19.

SMIRNOV, S.; REIJERS, H. A.; WESKE, M. From fine-grained to abstract process models: A semantic approach. *Information Systems*, Elsevier, v. 37, n. 8, p. 784–797, 2012. Cited in page 174.

SONG, M.; AALST, W. M. Van der. Towards comprehensive support for organizational mining. *Decision support systems*, Elsevier, v. 46, n. 1, p. 300–317, 2008. Cited in page 45.

SONG, S.; CAO, Y.; WANG, J. Cleaning timestamps with temporal constraints. *Proceedings of the VLDB Endowment*, VLDB Endowment, v. 9, n. 10, p. 708–719, 2016. Cited in page 178.

SONG, W. et al. Heuristic recovery of missing events in process logs. In: IEEE. *2015 IEEE International Conference on Web Services*. [S.l.], 2015. p. 105–112. Cited in page 178.

SOUTHAVILAY, V.; YACEF, K.; CALLVO, R. A. Process mining to support students' collaborative writing. In: *Educational data mining 2010*. [S.l.: s.n.], 2010. Cited in page 112.

SOUTHIER, L. F. P. 2022. Available from Internet: <https://drive.google.com/drive/folders/12suvNg76C57OH4XkjKMuy4BP39jgPLz6?usp=sharing>. Cited 2 times in pages 166 and 167.

SURIADI, S. et al. Event log imperfection patterns for process mining: Towards a systematic approach to cleaning event logs. *Information Systems*, Elsevier, v. 64, p. 132–150, 2017. Cited in page 178.

TAN, P.-N.; STEINBACH, M.; KUMAR, V. *Introduction to data mining*. [S.l.]: Pearson Education India, 2016. Cited in page 191.

TAX, N. et al. On generation of time-based label refinements. *arXiv preprint arXiv:1609.03333*, 2016. Cited 3 times in pages 174, 175, and 176.

TAX, N. et al. Generating time-based label refinements to discover more precise process models. *Journal of Ambient Intelligence and Smart Environments*, IOS Press, v. 11, n. 2, p. 165–182, 2019. Cited 2 times in pages 174 and 176.

TAX, N.; SIDOROVA, N.; AALST, W. M. van der. Discovering more precise process models from event logs by filtering out chaotic activities. *Journal of Intelligent Information Systems*, Springer, v. 52, n. 1, p. 107–139, 2019. Cited in page 177.

TAX, N. et al. Event abstraction for process mining using supervised learning techniques. In: SPRINGER. *Proceedings of SAI Intelligent Systems Conference*. [S.l.], 2016. p. 251–269. Cited in page 175.

TAX, N. et al. Log-based evaluation of label splits for process models. *Procedia Computer Science*, Elsevier, v. 96, p. 63–72, 2016. Cited in page 175.

TAX, N. et al. Mining process model descriptions of daily life through event abstraction. In: SPRINGER. *Proceedings of SAI intelligent systems conference*. [S.l.], 2016. p. 83–104. Cited in page 175.

TERRAGNI, A.; HASSANI, M. Analyzing customer journey with process mining: From discovery to recommendations. In: IEEE. *2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud)*. [S.l.], 2018. p. 224–229. Cited in page 173.

THABET, D. et al. Towards context-aware business process cost data analysis including the control-flow perspective: A process mining-based approach. In: SPRINGER. *Intelligent Systems Design and Applications: 19th International Conference on Intelligent Systems Design and Applications (ISDA 2019) held December 3-5, 2019 19*. [S.l.], 2021. p. 193–204. Cited 2 times in pages 22 and 24.

THABET, D.; GHANNOUCHI, S. A.; GHÉZALA, H. H. B. Towards business process model extension with cost perspective based on process mining. In: *Proceedings of the 16th International Conference on Enterprise Information Systems-Volume 3*. [S.l.: s.n.], 2014. p. 335–342. Cited 4 times in pages 22, 23, 24, and 25.

THABET, D.; GHANNOUCHI, S. A.; GHÉZALA, H. H. B. Petri net model cost extension based on process mining. In: *Proceedings of the 17th International Conference on Enterprise Information Systems-Volume 3*. [S.l.: s.n.], 2015. p. 268–275. Cited 3 times in pages 22, 23, and 24.

THABET, D.; GHANNOUCHI, S. A.; GHEZALA, H. H. B. A general solution for business process model extension with cost perspective based on process mining. *ICSEA 2016*, p. 251, 2016. Cited 3 times in pages 22, 23, and 24.

THABET, D.; GHANNOUCHI, S. A.; GHEZALA, H. H. B. A process mining-based solution for business process model extension with cost perspective context-based cost data analysis and case study. In: SPRINGER. *IFIP International Conference on Computer Information Systems and Industrial Management*. [S.l.], 2018. p. 434–446. Cited 3 times in pages 20, 22, and 24.

THABET, D.; GHANNOUCHI, S. A.; GHEZALA, H. H. B. Towards business cost mining: Considering business process reliability. In: SPRINGER. *Advances in Systems Engineering: Proceedings of the 28th International Conference on Systems Engineering, ICSEng 2021, December 14-16, Wrocław, Poland 28*. [S.l.], 2022. p. 127–137. Cited 3 times in pages 20, 22, and 24.

TRCKA, N.; PECHENIZKIY, M.; AALST, W. van der. Process mining from educational data. *Handbook of educational data mining*, Chapman & Hall/CRC, p. 123–142, 2010. Cited in page 112.

TU, T. B. H.; SONG, M. Analysis and prediction cost of manufacturing process based on process mining. In: IEEE. *2016 International Conference on Industrial Engineering, Management Science and Application (ICIMSA)*. [S.l.], 2016. p. 1–5. Cited 3 times in pages 22, 23, and 24.

UMER, R. et al. On predicting academic performance with process mining in learning analytics. *Journal of Research in Innovative Teaching & Learning*, Emerald Publishing Limited, v. 10, n. 2, p. 160–176, 2017. Cited in page 112.

UPFLUX. *UpFlux Process Mining*. 2023. Access date: 28 dec. 2022. Available from Internet: <https://upflux.net/>. Cited 2 times in pages 27 and 107.

VASILYEV, E.; FERREIRA, D. R.; IIJIMA, J. Using inductive reasoning to find the cause of process delays. In: IEEE. *2013 IEEE 15th Conference on Business Informatics*. [S.l.], 2013. p. 242–249. Cited 2 times in pages 105 and 106.

VOELKLE, M. C.; SANDER, N. University dropout: A structural equation approach to discrete-time survival analysis. *Journal of Individual Differences*, Hogrefe & Huber Publishers, v. 29, n. 3, p. 134, 2008. Cited in page 119.

VOIGT, S. N. von et al. Quantifying the re-identification risk of event logs for process mining. In: SPRINGER. *International Conference on Advanced Information Systems Engineering*. [S.l.], 2020. p. 252–267. Cited 2 times in pages 162 and 182.

WALICKI, M.; FERREIRA, D. R. Sequence partitioning for process mining with unlabeled event logs. *Data & Knowledge Engineering*, Elsevier, v. 70, n. 10, p. 821–841, 2011. Cited in page 178.

WANG, J. et al. Cleaning structured event logs: A graph repair approach. In: IEEE. *2015 IEEE 31st International Conference on Data Engineering*. [S.l.], 2015. p. 30–41. Cited in page 178.

WANG, J. et al. Efficient recovery of missing events. *Proceedings of the VLDB Endowment*, VLDB Endowment, v. 6, n. 10, p. 841–852, 2013. Cited in page 178.

WANG, R.; ZAÏANE, O. R. Discovering process in curriculum data to provide recommendation. In: *EDM*. [S.l.: s.n.], 2015. p. 580–581. Cited in page 112.

WEIJTERS, A.; RIBEIRO, J. Flexible heuristics miner (fhm). In: IEEE. *2011 IEEE symposium on computational intelligence and data mining (CIDM)*. [S.l.], 2011. p. 310–317. Cited in page 39.

WERNER, M.; WIESE, M.; MAAS, A. Embedding process mining into financial statement audits. *International Journal of Accounting Information Systems*, Elsevier, v. 41, p. 100514, 2021. Cited in page 19.

WYNN, M. et al. A framework for cost-aware process management: cost reporting and cost prediction. *Journal of Universal Computer Science*, Technische Universitat Graz from Austria, v. 20, n. 3, p. 406–430, 2014. Cited 9 times in pages 21, 22, 24, 25, 65, 66, 67, 68, and 69.

WYNN, M. et al. Cost-aware business process management: a research agenda. In: RMIT UNIVERSITY. *Proceedings of the 24th Australasian Conference on Information Systems (ACIS)*. [S.l.], 2013. p. 1–10. Cited 5 times in pages 19, 21, 22, 24, and 25.

WYNN, M. T. et al. Cost-informed operational process support. In: SPRINGER. *International Conference on Conceptual Modeling*. [S.l.], 2013. p. 174–181. Cited 4 times in pages 20, 21, 22, and 24.

WYNN, M. T.; SADIQ, S. Responsible process mining-a data quality perspective. In: SPRINGER. *International Conference on Business Process Management*. [S.l.], 2019. p. 10–15. Cited in page 180.

XU, X. et al. Tcpm: topic-based clinical pathway mining. In: IEEE. *2016 IEEE First International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*. [S.l.], 2016. p. 292–301. Cited in page 172.

YANG, H. et al. On global completeness of event logs. *BPM Center Report BPM-10-09*, 2010. Cited in page 179.

YANG, H.; WEN, L.; WANG, J. An approach to evaluate the local completeness of an event log. In: IEEE. *2012 IEEE 12th International Conference on Data Mining*. [S.l.], 2012. p. 1164–1169. Cited in page 180.

YASMIN, F. A.; BUKHSH, F. A.; SILVA, P. D. A. Process enhancement in process mining: a literature review. In: RHEINISCH WESTFÄLISCHE TECHNISCHE HOCHSCHULE. *CEUR workshop proceedings*. [S.l.], 2018. v. 2270, p. 65–72. Cited in page 45.

YOUSFI, A.; WESKE, M. Discovering commute patterns via process mining. *Knowledge and Information Systems*, Springer, v. 60, n. 2, p. 691–713, 2019. Cited in page 173.

ZELST, S. J. van et al. Event abstraction in process mining: literature review and taxonomy. *Granular Computing*, Springer, p. 1–18, 2020. Cited in page 164.

ZELST, S. J. van et al. Event abstraction in process mining: literature review and taxonomy. *Granular Computing*, Springer, v. 6, n. 3, p. 719–736, 2021. Cited in page 174.

ZELST, S. J. van et al. Filtering spurious events from event streams of business processes. In: SPRINGER. *International Conference on Advanced Information Systems Engineering*. [S.l.], 2018. p. 35–52. Cited in page 177.

ZELST, S. J. van et al. Detection and removal of infrequent behavior from event streams of business processes. *Information Systems*, Elsevier, v. 90, p. 101451, 2020. Cited in page 177.

ZHAO, W.; ZHAO, X. Process mining from the organizational perspective. In: *Foundations of intelligent systems*. [S.l.]: Springer, 2014. p. 701–708. Cited in page 35.

ZHU, R. et al. Automatic real-time mining software process activities from svn logs using a naive bayes classifier. *IEEE Access*, IEEE, v. 7, p. 146403–146415, 2019. Cited in page 173.

Appendix

APPENDIX A – Systematic Mapping Review on Log Preparation for Process Mining

In modern times, organizations need to adjust their business processes along with the changing environments to maintain a competitive advantage (BEEST; MARUSTER, 2007), and they are thus challenged with tracking and optimizing organizational processes to support their businesses. Process mining (PM) is a research discipline that helps effective process management that employs a detailed analysis of the behavior of operational processes (AALST et al., 2011b). The idea of the PM is to discover, monitor, and improve real processes by extracting knowledge from event logs readily available in today's (information) systems (IGLESIAS et al., 2011). To this end, activities need to be executed, and they can be grouped into three groups: discovery, conformance checking, and enhancement. Discovery refers to using event logs generated by information systems to discover process models. Conformance refers to detecting and analyzing misalignments between two models, be it a real and a discovered model, or between two extracted models. Enhancement refers to extending a process model by adding information about different aspects or repairing to fix some problems (MEDEIROS et al., 2007). Several PM applications have been described in the literature. PM has been applied to internal transaction fraud mitigation (JANS et al., 2011), usability analysis of automated teller machines (MAHMOOD; SHAIKH, 2013), business activities and transactions using electronic data interchange (EDI) (KRATHU et al., 2014), software engineering (RUBIN et al., 2014), education (MUKALA et al., 2015), discovering shoppers' pathways (HWANG; JANG, 2017), healthcare (CHIUDINELLI et al., 2020), and other areas (GARCIA et al., 2019).

In conventional PM, event logs are the starting point for the PM activities. They are characterized by attributes such as a case identifier, a timestamp, and an activity name. They are often supplemented with information about the transaction type (start, complete), resources, associated costs, and so on (GHASEMI; AMYOT, 2019). Because different systems generate event logs with different formats, they need to be prepared to be used in PM activities because the quality of the event logs used as input is critical for the success of any PM effort (BOSE; MANS; AALST, 2013b). As event log preparation, we consider acquiring or extracting the log from the data sources, preprocessing, evaluating, merging, enriching, filtering, correcting, and any other action that need to be performed so the event log is prepared to be given as input to PM activities.

Several advantages can be identified in the literature for preparing event logs. In (ANDREWS et al., 2019), the authors cite: (i) early identification of errors in the event log; (ii) early identification of errors in discovered process models and performance analyses;

(iii) improved understanding of the data being used to support the PM study; and (iv) opportunity to revise data extraction and event log generation (in the light of quality assessments) to minimize the risk of erroneous analysis (and consequent rework) caused by poor data. Another use of the event log preparation is for security and privacy issues (RAFIEI; WAGNER; AALST, 2020) (PIKA et al., 2019). As described by Voigt et al. (2020), event logs often contain sensitive information that can be related to individual process stakeholders through background information and cross-correlation. At this point, some techniques such as filtering and modifying the input (i.e., event logs), encryption, and making times relative can be used to hide any relevant information that can be used to identify sensitive data from event logs (RAFIEI; WALDTHAUSEN; AALST, 2018b).

The motivation for this research was to identify the problems related to event log preparation and to provide a map of the state-of-the-art solutions applied in this context. Given the variety of data sources, many event logs are generated and come with formats that challenge anyone trying to use them in PM. Thus, it is unclear what method, technique, or approach can be used in the event log preparation step and in which situation. In this context, a systematic mapping review study may be useful to clarify this point for documenting event log preparation problems and solutions. Furthermore, the description of the quality issues identified in the literature may help other researchers and organizations to upgrade the preparation of event logs to a more automatic step, thereby decreasing the manual efforts involved.

A.0.1 Our Contributions

- Identification and mapping of problems related to the preparation of event logs in PM;
- Description of the methods, techniques, and frameworks applied to solve this type of problem;
- Description of application and findings of the identified studies.

A.1 Related Studies

Ghazal, Ibrahim and Salama (2017) examined the literature on case studies conducted and related to the application of PM in education from 2009 to 2016. Their objective was to highlight the use of PM to improve educational processes and identify 37 key studies to collect information regarding methodologies applied in the domain.

Batista and Solanas (2018) presented the results of a review of the PM literature applied to the healthcare domain. The authors analyzed 55 articles that identified the objectives of the PM analysis, PM technique types applied, the perspective of mining

(organizational, performance, and time-related perspectives), algorithms and tools, medical facilities, medical fields, and process types. For process discovery, they analyzed medical data preparation techniques. They identified that the data is preprocessed by removing inaccurate or inconsistent events and dealing with the complexity of the so-called "spaghetti processes" using filters and clusters. Further, privacy issues, such as anonymization or pseudonymization, were examined in connection with the integration and standardization of activities.

A systematic review that aimed to determine the PM usage areas related to the agile development of software, commonly used algorithms, data sources, mechanisms for obtaining data, analysis techniques, and tools was conducted by Erdem, Demirörs and Rabhi (2018). Their study showed that PM is used in Agile software development, especially to discover models of process instances from event traces obtained from task-tracking applications.

Erdogan and Tarhan (2018) showed that despite healthcare data and technique-related challenges, the PM application is rapidly growing and is open for further research and practice. The authors analyzed 172 relevant papers concerning various aspects, including research and contribution type, application context and healthcare specialty, process modeling type and notations, PM techniques, and demographic and bibliometric analysis. They identified many studies that proposed methods and processes, tools, metrics, and models. Further, they found that most researchers applied the PM to discover a healthcare process, and only a few used PM to enhance healthcare processes. For the application area, the authors identified that most studies analyzed a single department or a single hospital instead of multiple departments or hospitals. They explain that this limitation is attributed to data integration and availability problems, data confidentially, or different physical locations. In addition, oncology was the most studied healthcare specialty.

Another study (GARCIA et al., 2019) identified research topics on PM regarding discovery, conformance checking and process enhancement, and PM discovery algorithms and their applications. They examined 1278 articles and noted that the most active research topics were associated with process discovery algorithms, conformance checking, and architecture and tools improvements. In addition, in application domains, the segments with major case studies are healthcare, followed by information and communication technology, manufacturing, education, finance, and logistics. The authors concluded with suggestions on the growth of the PM research area and its relevance for future research.

Ghasemi and Amyot (2020) argued that there are clear advantages in exploring the goals of existing processes, although goal-oriented approaches that consider event logs during model construction are rare. The authors presented results from a review of 24 articles, reporting on three main categories of studies: goal modeling, requirements elicitation, intention mining, and performance indicators. The results presented by the

authors: a) do not suggest a coherent line of research on PM in association with goals, and b) there is sparse research on performance indicators associated with PM.

Dakic et al. (2019) argued that various event log extraction techniques, approaches, and tools are being developed to be specific and generic. They presented the results of an SLR conducted to answer questions about the generality of the approaches, applicability by non-experts, and feasibility of the developed tools. Most approaches were developed for event log extraction from enterprise resource planning (ERP) systems with few applications applied to non-experts. Further, various tools and plug-ins were developed to automate the processes, such as event log extraction: XES Mapper, Xtract (v1, v2, and v3), OpenSlex, PADAS, and others. Yet another of the findings was the problems identified while performing event log extraction from ERP systems: convergence, which occurs when the same activity is executed on multiple process instances at once, and divergence, which occurs for one process instance when the same activity is performed multiple times.

Zelst et al. (2020) argued that preprocessing techniques that allow the abstraction of event data into the right granularity level is vital for PM's successful application. The authors presented an SLR in which they assessed state-of-the-art techniques for applying event abstraction in the PM field. The results were used to build an ontology that represents their findings. Despite the useful findings, there are some limitations to their research. Using specific search terms that are not sufficiently broad could have led to misconceptions regarding the state-of-the-art techniques for preparing event logs. Further, they did not include the IEEE Explore Academic Search Engine in their research.

Only two studies (DAKIC et al., 2019; ZELST et al., 2020) directly related to log preparation. Most related studies do not concern log preparation, indicating the need for this mapping study.

A.2 Systematic Literature Review Process

A Systematic Mapping Review aims to gather the available knowledge about a topic, synthesizing this knowledge by categorization of studies that could perhaps form the basis of a fuller review (KITCHENHAM; BUDGEN; BRERETON, 2015). This study follows the method of systematic mapping review in the software engineering proposed by Petersen et al. (2008). This method consists of three main phases: planning the review (as described in Section A.2.1), conducting the review process (as described in Section A.2.2) and presenting the results (as presented in Section A.3 and discussed in Section A.4).

A.2.1 Planning the Review

This paper aims to understand the current state-of-the-art techniques for the preparation of event logs for PM. To that end, the study goal is to gather knowledge about the issues

found in event logs preparation and the techniques used to solve these issues. This study can contribute to the knowledge of the topic, leading to available solutions in the literature. That means that PM researchers can use the results of this paper to identify what studies are available that will help them to solve the issues in their event log, making it ready for PM techniques. The following research questions (RQs) were used for this purpose:

- RQ1: What types of problems are involved in the preparation of event logs for PM? Here, we aim to identify the different issues that stimulate research to prepare event logs. One or more issues can be present in an event log and can be related to obtaining the event log, such as extracting it from a database or performing necessary operations to make the extracted log suitable for PM.
- RQ2: What approaches are available to solve these problems identified in RQ1? The solution proposed by researchers to solve the issues are identified here. The solutions can be presented as a specific method, technique, or framework containing several steps.
- RQ3: What is the research trend related to preparing event logs? What are the most frequent authors and venues? How is the research trend over the years? The goal of RQ3 is to map the bibliographical information related to event log preparation.

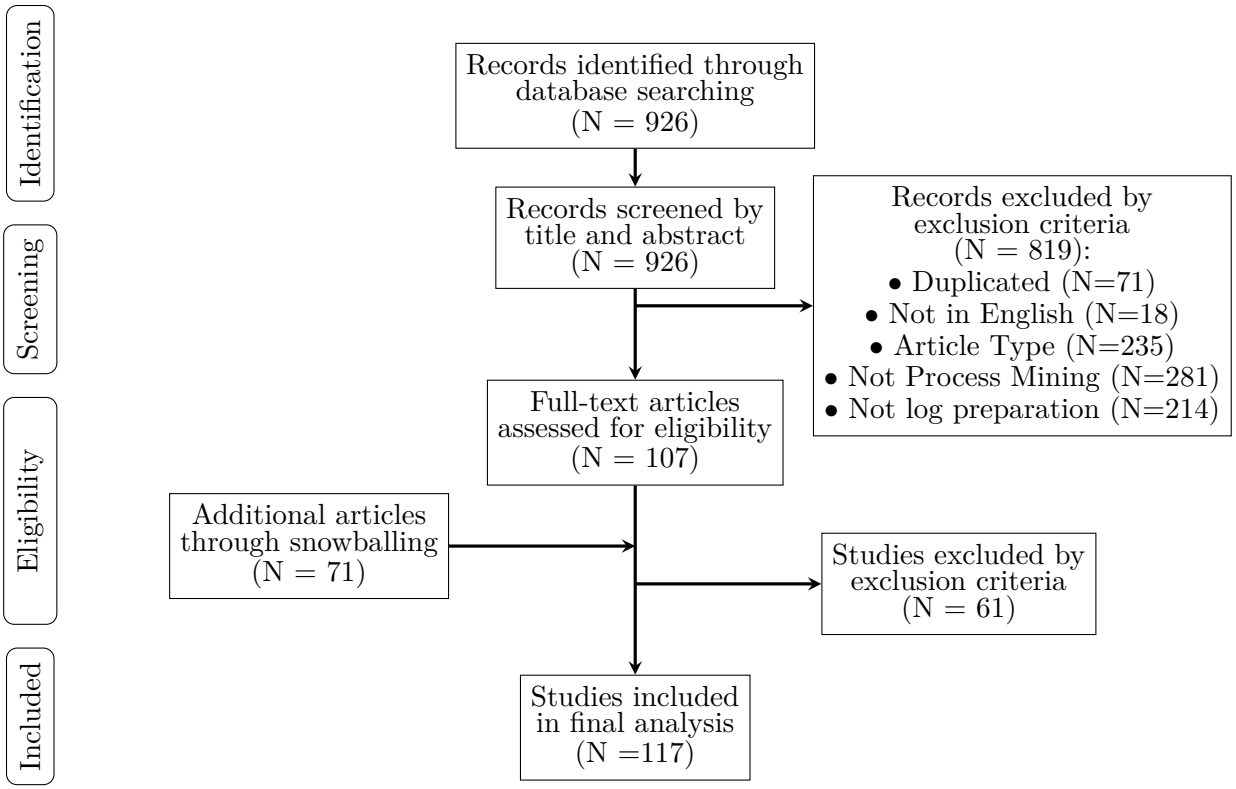
A.2.2 Conducting the Review

This step was first conducted by identifying relevant primary studies. Five digital libraries were used for this purpose: ACM Digital Library, IEEE Xplore, PubMed, Science Direct, and Springer Link. The digital libraries were selected based on the related studies. The process of identifying the studies is described below and is illustrated in Figure A.1.

A.2.2.1 Searching in digital libraries

An automated search in the five digital libraries was conducted using search strings, as listed in Table A.1. We considered the available results without specifying a time restriction. Therefore, all results available until December 2021—when the search was conducted—were included. As shown in Table A.1, the total number of automated retrieved papers is 926: 132 papers from ACM; 486, Springer Link; 126, IEEE Xplore; 176, Science Direct; and 6, PubMed. Our systematic mapping aims to identify issues (RQ1) and solutions (RQ2) present in the preparation of event logs for process mining. Therefore, only PM-related articles are included using the search strings terms “process mining” and “workflow mining.” Further, researchers can use several terms to describe the preparation of event logs. For the preparation of the event logs, our search strings included the terms “data pre-processing/preparation,” “event correction/preparation,” “log preparation”, and “cleaning event log.” All terms were considered the singular and plural forms. Since Science

Figure A.1 – PRISM chart of the systematic mapping review



Direct and PubMed already include the plural of terms in the results, we used wildcards in the strings on the other sources. Furthermore, because Science Direct restricts the number of operators in the search, the search string was divided into two, and the results were united.

The 926 article information was organized in a spreadsheet containing the following for each article: the database that it was retrieved from, title, authors, year of publication, abstract, type of publication, venue, URL, DOI, keywords, and address (when available). The spreadsheet can be found in (SOUTHIER, 2022).

A.2.2.2 Records eliminated by exclusion criteria

The 926 retrieved studies were screened by title and abstract by the first author. For quality purposes, exclusion criteria were used in studies that were not relevant to answering the RQs (PETERSEN et al., 2008). 71 studies were excluded because they were duplicated. 18 studies were not written in English, and therefore, were excluded. 235 records were excluded because of the article type, such as no peer-reviewed articles, academic thesis, book chapters, and short papers. Furthermore, studies that have not addressed PM (281 studies), and studies that have not covered log preparation (214 studies) were eliminated. 107 papers were selected in this step to be assessed by their full texts. These 107 studies were assessed by two different authors by their full texts. In case of disagreement, a third author decided if the article was relevant or not. Of the 107 articles, 46 were considered

Frame A.1 – Search strings by digital libraries

Digital library	Search string	#
ACM Dig Library	AllField:(("process* mining" OR "workflow* mining")) AND AllField:(("data-preprocessing" OR "data preparation" OR "data preprocessing" OR "data pre processing" OR "data pre-processing" OR "cleaning event log*" OR "event* correction" OR "event* preparation" OR "log* preparation"))	132
IEEE eXplore	("Full Text .AND. Metadata": "process* mining" OR "workflow* mining") AND ("Full Text .AND. Metadata": "data-preprocessing" OR "data preparation" OR "data pre-processing" OR "data pre processing" OR "data pre-processing" OR "cleaning event log*" OR "event* correction" OR "event* preparation" OR "log* preparation")	126
Springer Link	("process* mining" OR "workflow* mining") AND ("data-preprocessing" OR "data preparation" OR "data preprocessing" OR "data pre processing" OR "data pre-processing" OR "cleaning event log*" OR "event* correction" OR "event* preparation" OR "log* preparation")	486
Science Direct	("process mining" OR "workflow mining") AND ("data-preprocessing" OR "data preparation" OR "data preprocessing" OR "data pre processing" OR "data pre-processing" OR "cleaning event log" OR "event correction") ("process mining" OR "workflow mining") AND ("event preparation" OR "log preparation")	161 15
PubMed	((("process mining") OR ("workflow mining")) AND (("data-preprocessing") OR ("data preparation") OR ("data preprocessing") OR ("data pre processing") OR ("data pre-processing") OR ("cleaning event log") OR ("event correction") OR ("event preparation") OR ("log preparation"))	6

relevant studies. The details of the assessment can be found in (SOUTHIER, 2022).

A.2.2.3 Snowballing

Backward snowballing, also referred to as citation analysis (KITCHENHAM; BUDGEN; BRERETON, 2015), was executed to complete the research. Citation analysis was applied to all 46 relevant articles by screening their references by abstract and title. New articles were found relevant, they were assessed by two authors, and the citation analysis was applied to them as well. This process iterated until the citation analysis of all papers did not result in new papers that met the inclusion criteria. A total of 71 relevant papers were retrieved iteratively by snowballing: 60 in the first iteration, 10 in the second, and 1 in the third. No relevant new papers were found in the fourth iteration. As shown in Figure A.1, 117 relevant papers (46 from the automated search and 71 from snowballing) were selected in this research. The details of the citation analysis can be found in (SOUTHIER, 2022).

A.2.2.4 Data extraction

The data were extracted and tabulated from each relevant paper. Spreadsheets were used to record the information. Table A.2 shows the data extracted from the selected studies and the corresponding RQs related to the data. In Section A.3, we summarize and explain how the tabulated data answer each RQ. The spreadsheets with the extracted information can be found in (SOUTHIER, 2022).

In Section A.3, we discuss the last step of the systematic mapping review method

Frame A.2 – Extracted data from relevant papers

Extracted data	Relevant RQ
Title	RQ3
Authors	RQ3
Year of publication	RQ3
Country of the first author	RQ3
Keywords	RQ3
Venue	RQ3
Type of venue	RQ3
DOI	RQ3
Category of addressed problem	RQ1 and RQ2
Subcategory of addressed problem	RQ1 and RQ2
Description of approaches (methods, frameworks, or techniques) and corresponding problem that was addressed	RQ1 and RQ2
Application used for testing, validating, exemplifying, or illustrating the proposed approach	RQ1 and RQ2
Main indings and results	RQ1 and RQ2

in software engineering (PETERSEN et al., 2008): the results are presented.

A.3 Results

In this section, we describe the results obtained from this study. First, in section A.3.1, we provide the quantitative results for bibliographical information to give readers an idea of what the result set looks like. In section A.3.2, we present the classification of retrieved studies according to the review.

A.3.1 Quantitative results

This section shows the quantitative result related to the research and its trend, i.e., the distribution of selected studies per year, grouped by publication type, most frequent authors, countries, keywords, and venues. Figure A.2 shows the distribution of studies over time, grouped by the type of publication (conference or journal). A total of 50 studies were published in journals and 67 were published in conferences. The examined studies were distributed in a crescent trend since 2006; the highest number of publications were observed in 2018.

Figure A.2 – Number of Studies over time

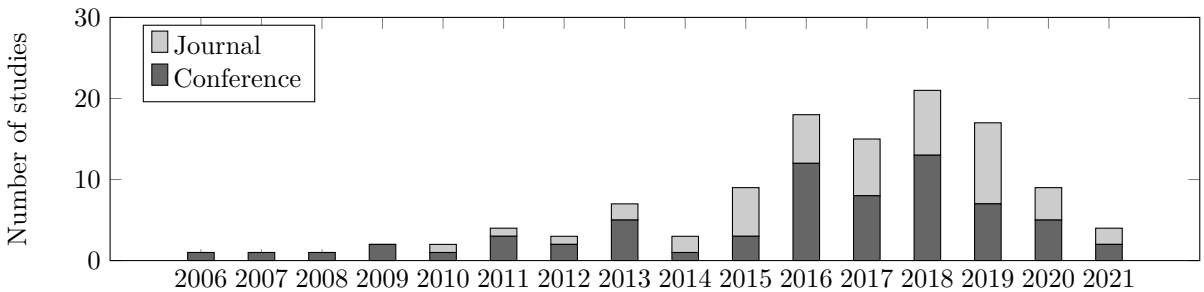
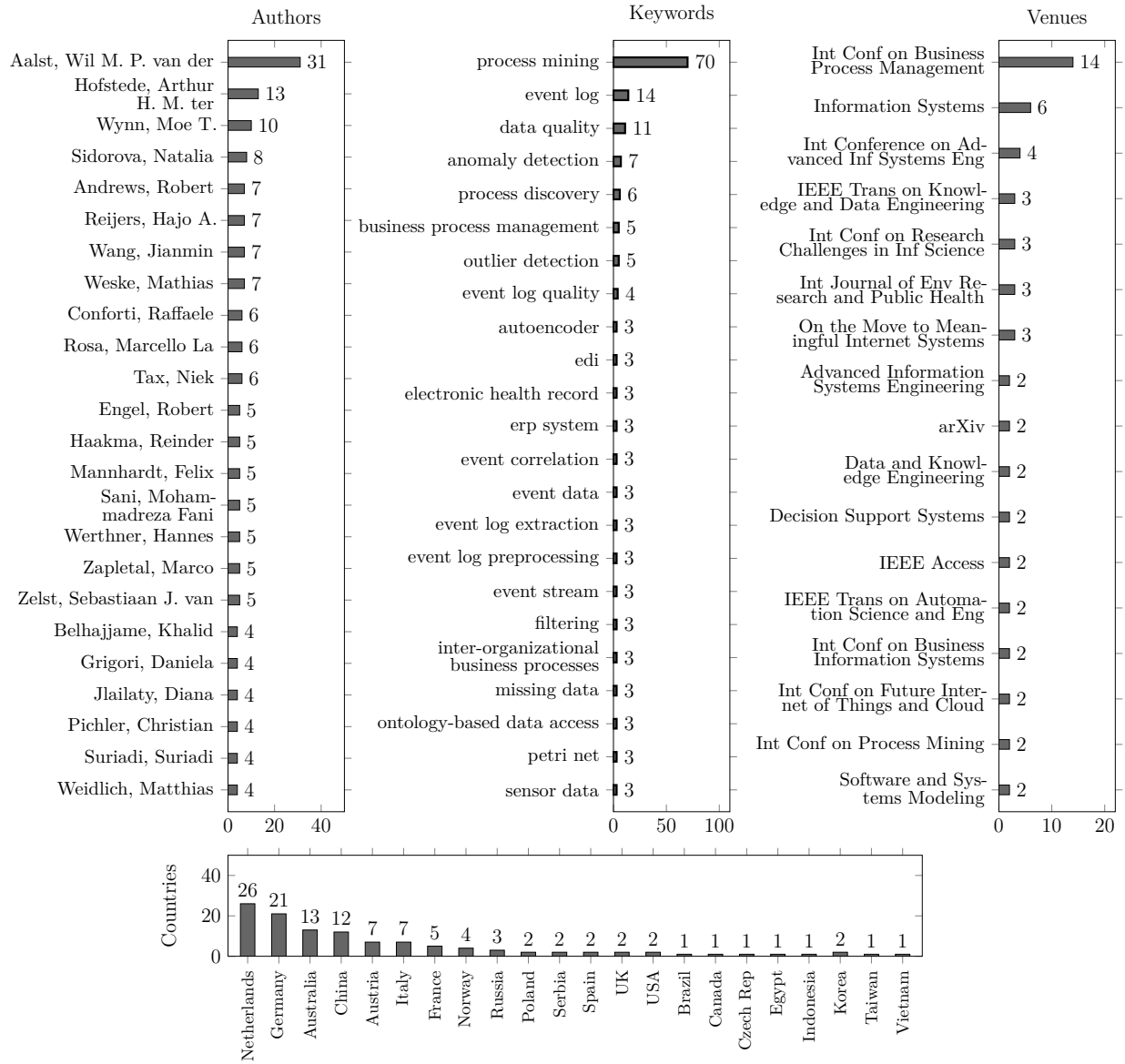


Figure A.3 shows the frequency of authors, keywords, venues, and countries in the examined studies. For clarity purposes, less frequent entries are omitted from this

Figure. Wil van der Aalst (31 studies), Arthur ter Hofstede (13 studies), and Moe T. Wynn (10 studies) are the people who most often appeared as authors in works related to data preprocessing in PM. "Process Mining" is the most frequent keyword with 70 studies, followed by "Event Log" (14 studies), and "Data Quality" (11 studies). The most frequent venues are the International Conference on Business Process Management (14 studies), Information Systems (6 studies), and International Conference on Advanced Information Systems Engineering (4 studies). Countries with the most frequent first authors are the Netherlands (26 studies), Germany (21 studies), and Australia (13 studies).

Figure A.3 – Most frequent authors, keywords, venues, and countries

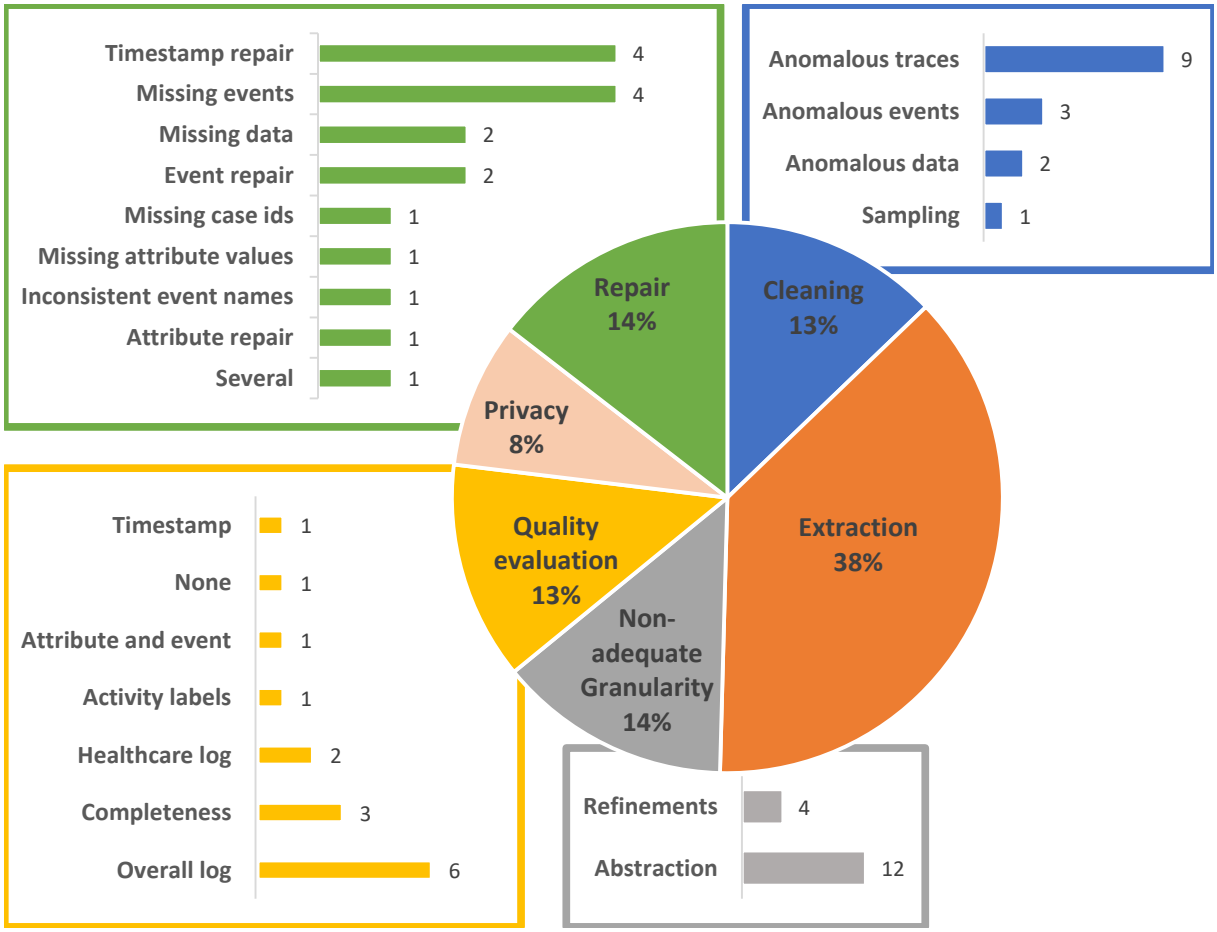


A.3.2 Classification of log preparation studies

The studies were classified according to the problems and solutions presented. Six categories for classification of the studies were identified and used: Extraction, Non-adequate

Granularity, Quality evaluation, Privacy, Repair, and Cleaning. With exception of Extraction and Privacy, the type of approach is shown in details on Figure A.4 along with the numbers of studies in each category/type. The details of each category are explained next.

Figure A.4 – Classification of the selected studies



A.3.2.1 Extraction

Here we tabulate the studies that focus on data acquisition or data extraction. The main goal of each article is to describe how to obtain the event log for specific data sources such as PAIS (Process-aware information systems) (GÜNTHER; AALST, 2006), ERP (Enterprise resource planning) systems (INGVALDSEN; GULLA, 2007), natural language (KABICHER; RINDERLE-MA, 2011), EDI (Electronic data interchange (EDI) messages (ENGEL et al., 2011), non-PAIS (PÉREZ-CASTILLO et al., 2014), real-time location system (LIU et al., 2014), relational databases (CALVANESE et al., 2016), EMR (Electronic medical records) (HUANG et al., 2015), emails (JLAILATY; GRIGORI; BELHAJJAME, 2016), sensor data (ECK; SIDOROVA; AALST, 2016), transaction log (PAJIĆ; BEČEJSKI-VUJAKLIJA, 2016), etc. Our research identified 44 studies related to extraction when searching for preprocessing techniques. Table A.1 presents these studies.

Table A.1 – Extraction

Study	Description	Application	Findings
Günther and Aalst (2006)	The ProM Import Framework is presented. It is designed for the extraction of event log data from any given PAIS implementation	Seven PAIS	Its flexible and extensible architecture makes it a versatile contribution to the general BPI community
Ingvaldsen and Gulla (2007)	An ERP log analysis system is described. It allows one to define at a meta-level how events, resources and their relations are stored and transformed for use in PM.	None	Large scale PM in SAP became feasible
Engel et al. (2011)	Extract process information from EDI message exchanges	ProM plugin	A two-staged technical architecture is proposed for ProM
Kabicher and Rinderle-Ma (2011)	A method to extract a process model from natural language in written form is presented.	12 interviews at the University of Vienna.	The method was successfully performed
Engel et al. (2012)	Follow-up of Engel et al. (2011). A case study of EDI message extraction is presented.	410 EDIFACT messages supplied by an automotive supplier company	Inter-organizational business process models can be derived by analyzing EDI messages
Mueller-Wickop and Schultz (2013)	It extracts activity sequence from accounting data.	Data from a big news-agency	A case study demonstrates the effectiveness of the presented approach
Engel et al. (2013a)	Follow-up of Engel et al. (2012). The toolset EDIminer is proposed.	Real-world data of purchase order process.	The utility of EDIminer is demonstrated.
Engel et al. (2013b)	Follow-up of Engel et al. (2013a). The study extracts Key Performance Indicators from EDI messages.	None	It enables organizations to monitor activities and events which are not modeled in their ERP systems.
Liu et al. (2014)	It provides a study of workflow modeling by the integrated analysis of a real-time location system	A hospital environment	The framework can effectively model the workflow patterns, and have managerial applications in workflow monitoring, auditing, and inspection of workflow compliance.
Pérez-Castillo et al. (2014)	It supports event log collection from non-process-aware information systems by correlating events in the appropriate business process instance	Author management system and healthcare information system	The results show that the technique is able to obtain event logs from traditional systems.
Aalst (2015)	An approach is described that scopes, binds, and classifies data from a database to create event logs.	None	The paper only conceptualized the different ideas.
Lu et al. (2015)	A semi-automatic, end-to-end approach is presented for extracting event data in a plain database of an ERP system into an artifact-centric process model.	Two case studies	The approach allowed to successfully analyze processes of ERP systems
Huang et al. (2015)	A probabilistic topic model is proposed, to link patient features and treatment behaviors together to mine treatment patterns hidden in electronic medical records.	985 EMRs from a Chinese hospital	The approach can effectively identify meaningful treatment patterns from EMRs
Pajić and Bečejski-Vujaklija (2016)	The paper captures the knowledge of existing approaches and tools in converting the data from transaction logs to event logs.	Dynamics NAV ERP system	The paper results in the creation of an ontological metamodel.

Jlailaty, Grigori and Belhajjame (2016)	A method is proposed for mining process models from email logs that leverage unsupervised machine learning techniques with little human involvement	Real-world dataset	A use case demonstrates the usefulness of the proposed solution
Calvanese et al. (2016)	A framework is proposed that supports domain experts in the extraction of XES event log from relational databases	None	The framework enables the possibility of maintaining logs virtual and fetch log-related information on-demand
Murillas, Aalst and Reijers (2016)	The paper proposes an approach that exploits database redo logs.	Synthetical data	The approach demonstrates its potential to answer a range of business questions.
Eck, Sidorova and Aalst (2016)	The paper addresses the challenge of applying PM to sensor data.	Case study performed at Philips	The case study demonstrates that the use of PM can add value to the smart product design process.
Engel et al. (2016)	Follow-up of Engel et al. (2012) and Engel et al. (2013a). The EDImine Framework is proposed.	Real-world EDI data of a German consumer goods manufacturing company	The applicability of the approach is shown by means of a case study
Xu et al. (2016)	A topic-based clinical pathway mining approach is proposed, which is concise, interpretable and of sequential information.	Real-world data	The experiments shows the effectiveness and practicability of the approach
Diamantini et al. (2017)	It extracts events from mobility patterns of social media users geotagged posts.	Instagram user posts from exposition	Results shows the capability of the proposed methodology to support the analysis of mobility patterns and to derive interesting insights
Jlailaty, Grigori and Belhajjame (2017a)	A method to discover business process instances from email logs is proposed.	250 emails	Experimental results prove the approach contributions.
Jlailaty, Grigori and Belhajjame (2017b)	Follow-up of Jlailaty, Grigori and Belhajjame (2017a). A method to discover business process activities from email logs is proposed.	250 emails	Experimental results are detailed to prove the approach contributions.
Calvanese et al. (2017a)	The paper proposes ontology-based data access for extracting event logs from legacy information systems	Examples	The method automates the extraction of event logs, manipulating and reasoning over mappings and annotations.
Calvanese et al. (2017b)	Follow-up of Calvanese et al. (2017a).	Real-world case	The study illustrates the limitations of manual extraction and shows the features of the approach
Huang, Jin and Wang (2016)	It extracts clinical-event-packages from billing data	Billing data of 240 patients	The experiment that the approach is a good way of generating more comprehensible clinical process.
Hwang and Jang (2017)	Extraction of WiFi-based positioning system is presented, for pathway analysis in an off-line store	WiFi signal-capturing device in a retail store of a fashion brand in South Korea	The customers' pathway in the store could be analysed in two scenarios.
Dišek, Šperka and Kolesár (2017)	It extracts real data from the raw format from different information systems to the event log files in a large automotive company	Automotive company data	Experimental results show the success of the case study
Metsker et al. (2017)	The paper proposes the use of text mining methods to extract events from electronic health records.	Russian language medical data of Acute coronary syndrome) patients	The method is demonstrated in the selected data set, and the extracted data is used in PM

Hernandez et al. (2017)	A linear-temporal logic model checking approach is proposed for the analysis of structured e-commerce Web logs.	Real case study of a Spanish e-commerce website	The results have identified interesting findings that have made it possible to propose some improvements in the website design to increase its efficiency
Terragni and Hassani (2018)	This paper contributes a novel approach for applying process mining techniques to weblog customer journey analysis	Real-life case	The study could describe the user behavior, find useful insights, and propose personalized recommendations
Simović, Babarogić and Pantelić (2018)	An abstract syntax of domain-specific language (DSL) is presented, for extraction of event logs from ERP systems	None	The basic concepts of the language as well as principles are discussed in the paper
Rinner et al. (2018)	The paper shows how existing clinical data collected during melanoma surveillance can be prepared and preprocessed to be reused for process mining.	Melanoma event log	The presented approach enables the use of PM techniques on the cited event log.
Liu (2018)	The paper proposes an approach to extract a hierarchical software event log from a software event log by recursively applying a method calling relation detection	one real-life and two synthetic software event logs	The approach can help visualize actual software runtime behavior in an easy-to-understand manner
Murillas, Reijers and Aalst (2016)	The paper proposes a meta-model to integrate process and data perspectives, to generate different views from the database at any moment.	Database redo logs, in-table version storage, and SAP-style change tables	The technique's applicability has been demonstrated in real-life environments. Also, an implementation of the solution has been developed and tested.
Brzychczy and Trzcionkowska (2018a)	The paper presents event log extraction from a low-level machinery monitoring system used in an underground mine.	Real industrial data sets	The approach is tested on a selected example from the longwall monitoring system
Yousfi and Weske (2019)	This paper integrates methods from ubiquitous computing for extracting events for business process management.	Ten test subjects with smartphones	The main contribution is to integrate location-based services within the process flow
Jlailaty, Grigori and Belhajjame (2018)	The paper goal is to recast emails into business activity-centric resources.	Public email dataset	The approach is validated in a real-world case
Knoll, Waldmann and Reinhart (2019)	This paper extends internal logistics ontology focusing on the process perspective. Existing internal ontologies are reviewed, compared, and merged.	42 relevant ontologies	The main contribution is a systematic approach to re-use existing ontologies.
Khovrichev et al. (2019)	The paper proposes the extraction of events from detection systems of systolic/diastolic pressure and heart rate	219 patients	The approach enables the extraction and integration of devices to analyze chronic diseases
Zhu et al. (2019)	The proposed approach extracts activity from the software configuration management systems log	Two real-world software development process logs, ArgoUML and jEdit	The experimental results show that the approach validates the approach
Andrews et al. (2020)	The paper presents the RDB2Log, a quality-aware, semi-automated approach for extracting event logs from relational data.	Real-world case	The evaluation shows that RDB2Log is understandable, relevant in current research, and supports process mining in practice.
Combi et al. (2021)	The proposed approach allows the designer to model the connection between business processes and database models	Controlled experiment with students and academics	The proposed approach improves the comprehension of integrated processes and data

Carrasquel, Chuburov and Lomazova (2021)	An approach to extract event logs from Financial Information Exchange protocol messages is proposed	Examples	The approach is illustrated in examples and a program is created

A.3.2.2 Non-adequate Granularity

Abstractions and Refinements are solutions related to the problem of the non-adequate granularity of event data. *Refinements* are used to bring more context to event labels, avoiding overgeneralizations. In this case, one event label can be split into two or more refined event labels to adequately model the actual behavior (TAX et al., 2016). The problem of overgeneralization of events normally occurs when a triggering sensor is used as the label for sensor events (TAX et al., 2019). On the other hand, *abstractions* deal with the problem of different or too-fine-grained levels of granularity. According to (ZELST et al., 2021), PM techniques assume that the event data are of the same and appropriate level of granularity, but in practice, the data are extracted from different systems at different granularity levels. Our research identified 16 studies related to non-adequate granularity when searching for preprocessing techniques. Table A.2 presents the findings.

Table A.2 – Non-adequate granularity

Study	**Type**	**Description**	**Application**	**Findings**
Bose and Aalst (2009)	Abstraction	Commonly used constructs in the event log are characterized, and pattern definitions to capture this constructs and to create abstractions over them are proposed. This is done by an interactive method of transformation of traces.	A set of 1372 event traces of a real health care system	The process model mined from the abstracted log is more comprehensible. The abstractions were formed over activities that are related by a functionality
Günther, Rozinat and Aalst (2009)	Abstraction	An activity mining approach based on global trace segmentation is proposed. The goal is to group low-level events into clusters, which represent higher-level activity.	A real-life event log from ASML's test process	A simplified log was obtained and used to discover better process models
Smirnov, Reijers and Weske (2012)	Abstraction	It extends beyond the existing works that analyze the structure of process models. It uses semantic information to decide on which activities belong together.	A set of business process models from a large telecommunication service provider	The experimental validation provides strong support for the applicability and effectiveness of the presented ideas
Baier, Mendling and Weske (2014)	Abstraction	It exploits the difficulty that automated abstraction approaches have in capturing the required domain knowledge. An approach is proposed that aims to abstract an event log in a semi-automatic way that uses domain knowledge extracted from existing process documentation	Two case studies with a German IT outsourcing company	The abstraction approach was able to deal with n:m relations between events and activities and also supports concurrency

Deokar and Tao (2015)	Abstraction	It presents an overall computational framework for event log abstractions. The abstractions are created hierarchically, and the aggregation of events is done by identifying phrase-based semantic similarities between normalized event names	Synthetic event data	A prototype was created. The experiment conducted shows promising results with practical implications
Tax et al. (2016a)	Abstraction	It presents an approach to generate feature vector representations of events an to abstract events with Condition Random Fields using the features. A metric to evaluate supervised event abstraction results was proposed as well.	Both real and synthetic event data	Supervised event abstraction can be used to discover smaller, more comprehensible, high-level process models
Mannhardt et al. (2016b)	Abstraction	A method based on behavioral activity patterns is proposed. By aligning activity patterns and the low-level event log, it is possible to obtain the abstracted event log.	The method was evaluated with domain experts of a Norwegian hospital using an event log from their digital whiteboard system	Process mining methods provide valuable insights on the usage of the system when using the abstracted event log but fail when using the original lower level event log
Lu et al. (2016)	Refinements	An approach for refining labels based on their context is proposed.	Controlled setting and a Dutch hospital log	A 42% improvement in the quality of the discovered models.
Tax et al. (2016b)	Refinements	A statistical evaluation method to determine the usefulness of the refinements is presented.	Van Kasteren smart home environment data set	The method was able to select two label refinements out of a set of candidates that had a positive effect on model precision
Tax et al. (2016)	Refinements	The lack of an automated approach to create refinements is explored. A framework for automatic label refinements based on time perspective is proposed. Fuzzy clustering is used to identify dense areas in time-space for each label.	Van Kasteren smart home environment data set)	Refinements of event labels enable discovery of more precise and insightful process models
Alharbi, Bulpitt and Johnson (2017)	Abstraction	A method is proposed for reduction of outlier events. It uses interval-based patterns to determine outlier thresholds based on the time of events occurring and the distinctive attribute of observed events.	The MIMIC-II open access medical dataset	The method has improved model precision conformance without reducing model fitness
Leonardi et al. (2018)	Abstraction	A framework for abstracting event logs into higher level concepts based on domain knowledge is presented. The approach can manage interleaved actions or delays between two actions abstracting to the same concept. A trace comparison approach is available, considers abstraction phase penalties, and deals with quantitative and qualitative temporal constraints in abstracted traces.	A stroke care event log	It is easier to identify common behaviors in abstracted traces while preserving outliers
Tax et al. (2016c)	Abstraction	Follow-up of Tax et al. (2016a) including the same approach	Van Kasteren event log, MIT household A and B event logs	Process models discovered after abstraction are more precise

Cruchten and Weigand (2018)	Abstraction	Several data preparation methods for abstraction are proposed. They apply logistic domain knowledge for process mining the material movements within an organization. Furthermore, an adapted process mining project methodology is presented.	A material movements log	Process Mining techniques are applied to identify the performance and compliance of materials.
Brzychczy and Trzcionkowska (2018b)	Abstraction	An approach is proposed requiring the creation of high-level event logs based on low-level events from the longwall monitoring system.	Example log	The approach enabled PM technique on low-level sensor data from machinery monitoring system
Tax et al. (2019)	Refinements	Follow-up article of Tax et al. (2016). Four strategies are presented to create combinations of multiple-label refinements.	Van Kasteren smart home environment data set	It allows the discovery of more insightful and behaviorally more specific process models.

A.3.2.3 Cleaning

The cleaning category is related to the problem of removing part of the event data. Generally, the studies in this category are concerned with noisy or anomalous information presented in the original event log. This information can severally impact the results of PM techniques (LEONI; MAGGI; AALST, 2015). The type of cleaning can be related to filtering cases that are presented as *anomalous traces* (BEZERRA; WAINER, 2013). Some studies also focus on filtering *anomalous events* within the trace. Also, recent techniques focus on cleaning incorrect traces, events, and attributes, i.e., *Anomalous data* in a general way. There is also a study focusing on filtering traces with the goal of *sampling*, that is, choosing a representative subset of cases in the log. Our research identified 15 studies related to cleaning when searching for preprocessing techniques. Table A.3 presents the findings.

Table A.3 – Cleaning

Study	**Type**	**Description**	**Application**	**Findings**
Ghionna et al. (2008)	Anomalous traces	An approach is proposed to detect anomalous evolutions within a set of process traces, which takes into account both statistical properties of the log and the constraints associated with the process model	Synthetic trace with a configurable percentage of outliers	Quasi-optimal values when the percentage of outliers is under 9%
Bezerra and Wainer (2013)	Anomalous traces	Four algorithms for detecting anomalies (frequency, threshold, sampling, and iterative) are compared	1500 artificial logs	The sampling algorithm proved to be the most effective
Leoni, Maggi and Aalst (2015)	Anomalous traces	The paper focuses on aligning event logs and predefined declarative process models to clean and repair the event log	Synthetic and real-life logs	The alignment-based approach has been implemented and evaluated
Cheng and Kumar (2015)	Anomalous traces	A technique is proposed to clean noisy logs by using a classifier is proposed on a subset of the log.	Synthetic logs from benchmark models	Mined models produced from such preprocessed logs are superior on several evaluation metrics

Conforti, Rosa and Hofstede (2015)	Anomalous traces	It presents an automated technique to the removal of infrequent behavior from event logs	Eight noisy synthetical logs and four real-life logs	A significant improvement over fitness, precision and complexity is shown, without a negative effect on generalization
Nolle, Seeliger and Mühlhäuser (2016)	Anomalous traces	A neural network system is proposed that is able to deal with noise in the log	Five different event logs	The approach results on 97.2% F1-score in detecting anomalous traces
Conforti, Rosa and Hofstede (2016)	Anomalous traces	Follow-up of Conforti, Rosa and Hofstede (2015). An automated technique to the removal of infrequent behavior from event logs is proposed.	Eight noisy synthetical logs and four real-life logs	The technique significantly improves the quality of the discovered process models. It scales well to large datasets.
Sani, Zelst and Aalst (2017)	Anomalous traces	A filtering method is proposed that exploits observed conditional probabilities between sequences of activities.	Real and synthetic event data	The proposed method accurately removes irrelevant behaviour and improves process discovery results.
Nolle et al. (2018)	Anomalous traces	A follow-up article of Nolle, Seeliger and Mühlhäuser (2016). A method using autoencoders for detecting anomalies is proposed.	700 different datasets.	The approach reached an F1 score of 0.87. Also, it can be used to detect which event causes the anomaly.
Tax, Sidorova and Aalst (2019)	Anomalous events	A technique to filter out chaotic activities from event logs is proposed.	Seventeen real-life event logs	The filtering method enables the discovery of more behaviorally specific process models.
Zelst et al. (2018)	Anomalous events	A event processor is proposed to filter out spurious events from a live event stream.	Synthetic and real-life datasets	High filtering accuracy for different instantiations of the filter was obtained.
Liu et al. (2018)	Sampling	A graph-based ranking model for event log sampling and an approach to measure the quality of the sample are proposed.	Synthetic and real-life event logs	The experimental analyses show that the sampling approach provides an effective solution to improve process discovery efficiency and ensure the high quality of the discovered model.
Zelst et al. (2020)	Anomalous events	Follow-up of Zelst et al. (2018)	Synthetic and real-life datasets	High filtering accuracy for different instantiations of the filter was obtained.
Krajsic and Franczyk (2020)	Anomalous data	It presents the lambda architecture in which an autoencoder is used for anomaly detection of incorrect traces, events, and attributes.	BPIC19 dataset Dongen (2019)	The conducted experiment showed success in detecting anomalies in event data
Krajsic and Franczyk (2021)	Anomalous data	A semi-supervised classification model is presented that takes into account different developments in deep learning, time series analysis, and sequence processing	BPIC19 dataset Dongen (2019)	It is able to filter activity-related and time-related anomalies from the event data

A.3.2.4 Repair

The repair category includes studies related to fixing some aspect of event data. *Missing data* (SIM; BAE; CHOI, 2019) is one of the most common problems found in this category. It can present as *missing attribute values* (WALICKI; FERREIRA, 2011), *missing case ids* (BAYOMIE et al., 2016), and *missing events* (ROGGE-SOLTI et al., 2013a), for example. Also, wrong or erroneous information can be repaired. Examples of this are *attribute repair*, *event repair*, *timestamp repair* and *inconsistent event names repair*. Our research identified 17 studies on repairing erroneous or missing data when searching for preprocessing techniques. Table A.4 presents the findings.

Table A.4 – Repair

Study	Type	Description	Application	Findings
Walicki and Ferreira (2011)	Missing attribute values	A method based on autoencoders is proposed for detecting anomalous values and reconstructing missing values at the level of attributes in event logs.	Real-world and artificially-generated event logs	The proposed approach shows remarkable performance
Rogge-Solti et al. (2013a)	Missing events	Follow up of Rogge-Solti et al. (2013b)	Synthetic data and real event data from a Dutch hospital	The evaluations indicate that the method can effectively repair if noise is limited
Rogge-Solti et al. (2013b)	Missing events	The paper uses stochastic Petri nets, alignments, and Bayesian networks to repair missing entries in the logs	None	The method is proposed and formally defined
Wang et al. (2015)	Inconsistent event names	The paper proposes a graph repair approach for repairing inconsistent event name.	Real and synthetic event data set	Experiments demonstrate the effectiveness and efficiency of proposed methods
Song et al. (2015)	Missing events	The paper uses the technique of process decomposition and presents heuristics to efficiently prune the unqualified sub-processes that fail to generate the minimum recovery.	Real-world processes	The experimental results demonstrate that the approach achieves high accuracy
Song, Cao and Wang (2016)	Timestamp repair	The paper proposes a method for repairing inconsistent timestamps that do not conform to the required temporal constraints	Real datasets	Experiments demonstrate the efficiency of the approach
Bayomie et al. (2016)	Missing case ids	The paper proposes an approach to deduce case ID for the unlabeled event log depending on the knowledge about the process model	Synthetical log	The approach can handle noise and incompleteness, but not cyclic models
Wang et al. (2013)	Missing events	The paper studies the efficient techniques for recovering missing events	Real data set	The experimental results demonstrate that the approach achieves high accuracy
Suriadi et al. (2017)	Several	The paper shows that a patterns-based approach is applicable to documenting event log quality issues, describing them as patterns, and repairing them.	Case study	This paper directly addresses several of the challenges facing PM
Dixit et al. (2018)	Timestamp repair	The paper describes a set of timestamp-based indicators for detecting event ordering imperfection issues in a log and repair them using domain knowledge	Two publicly available logs	The experiments were able to show that the approach can detect anomalies and repaired them

Sani, Zelst and Aalst (2018)	Event repair	An event data repair method is proposed, that tries to detect and repair outlier behavior within the given event data using a probabilistic method of frequency of activities in specific contexts.	Artificial and real event data	The approach was able to detect and modify most types of outlier behavior in the event data
Conforti, Rosa and Hofstede (2018)	Timestamp repair	The approach reorders events with erroneous timestamps and assigns an estimated timestamp to each such event	Synthetic and real-life logs	The experiments show that the approach significantly reduces the number of incorrect timestamps, while the reordering of events scales well to large and complex datasets
Nguyen et al. (2019)	Attribute repair	A method based on autoencoders is proposed to detect anomalous values and reconstruct missing values at the level of attributes in event logs.	Artificial and real-life event logs	Process models discovered from reconstructed event logs have lower variability of allowed behavior
Sim, Bae and Choi (2019)	Missing data	A likelihood-based Multiple Imputation by Event Chain method is proposed for dealing with imperfect event logs with missing data.	Sample event logs and a real steel manufacturing event log	The method repaired the event log to a high-level
Folino and Pontieri (2019)	Missing data	The paper exploits auxiliary AI tasks to deal with incomplete data	None	The approach is only proposed
Sani, Zelst and Aalst (2019)	Event repair	Follow-up of Sani, Zelst and Aalst (2018). A data preprocessing method is proposed that detects and subsequently repairs outlier behavior in event data.	Real and synthetic event logs	The evaluation demonstrates that it is possible to improve process discovery results by repairing event logs upfront.
Conforti et al. (2020)	Timestamp repair	The paper contributes an approach for automatically correcting same-timestamp errors in event logs.	Artificial and real-life dataset	The effectiveness and efficiency of the approach were assessed via the experiments

A.3.2.5 Quality evaluation

The studies in this category are related to some aspect of quality evaluation in event data. The *completeness* (YANG et al., 2010) of a log can be evaluated, and the *full log* (BOSE; MANS; AALST, 2013a) as well. Also, specific aspects of the log can be evaluated, such as *timestamps* (FISCHER et al., 2020), *activity labels* (SADEGHIANASL et al., 2019), *attribute and event* (ANDREWS et al., 2019). Also, some studies propose the evaluation specifically of healthcare logs (FOX et al., 2018). Our research identified 15 studies related to quality evaluation. Table A.5 presents the findings.

Table A.5 – Quality evaluation

Study	**Type**	**Description**	**Application**	**Findings**
Yang et al. (2010)	Completeness	An approach is proposed to estimate completeness of an event log	600 log files	The experiments provide some indication of the potential use of the technique

Hee, Liu and Sidorova (2011)	Completeness	A method to compute the probability that the event log is complete is proposed.	Synthetical data	The empirical studies show that the probabilistic bounds computed by the method are reliable
Yang, Wen and Wang (2012)	Completeness	An approach is proposed in the context of mining control-flow dependencies to evaluate the local completeness of an event log without knowing any information about the original process model	Generated logs	Experiment results show that the proposed approach works robustly and gives better estimation than approaches available
Bose, Mans and Aalst (2013a)	Full log	The paper identify four categories of process characteristics issues that may manifest in an event log and 27 classes of event log quality issues	Five real-life event logs	The analyzed logs illustrate the omnipresence of process and event log issues
Kherbouche, Laga and Masse (2016)	Full log	A qualitative model is proposed which aims to assess the quality of event logs	Artificial and real-life logs	The approach has been implemented in ProM and evaluated. It is noteworthy that it is not exhaustive and can be enriched by empirical research
Lu and Fahland (2017)	Full log	The paper discusses a conceptual framework to help users understand how quality issues could be presented and interrelated in event logs.	Examples	The framework is merely proposed
Fox et al. (2018)	Healthcare log	A care pathway data quality framework is proposed to identify, manage and mitigate EHR data quality in the context of process mining	Dental EHR log	The framework helped identify potential data quality issues and mark-up every data point affected.
Andrews et al. (2018)	Full log	The paper proposes a log query language that provides direct support for detecting log imperfections.	Examples	The approach identifies a set of function primitives for detecting some data quality issues
Sadeghianasl et al. (2019)	Activity labels	A automatic approach is proposed for detecting synonymous labels and polluted labels by using activity context	Real-life logs from two hospitals and an insurance company	The approach was implemented and validated and have achieved promising results in detecting frequent imperfect activity labels.
Andrews et al. (2019)	Attribute and event	A process-centric, data quality-driven approach for assessment at both attribute and event level.	A case study involving a real-life Ambulance service log	Datasets show how quality metrics and the approach can be used.
Wynn and Sadiq (2019)	None	The paper outlines foundations concepts of data quality with focus on event data	None	Key challenges and possible approaches to tackle data quality problems are elaborated on.
Kurniati et al. (2019)	Healthcare log	The paper explores data quality issues for healthcare process mining	Medical Information Mart for Intensive Care III database	It provides a case study of PM using the cited databases to illustrate the approach

Emamjome et al. (2020)	Full log	This paper introduces a framework facilitates an informed way of dealing with data quality issues in event logs through supporting both prognostic and diagnostic	Examples	It shows how the work can be applied to deal with data quality issues in logs
Fischer et al. (2020)	Timestamp	The paper defines 15 metrics related to timestamp quality in the log	Three real-life event logs	The approach has been implemented and evaluated by experts
Neumann et al. (2021)	Full log	The paper proposes and approach to assess data quality problems in production logs	Example	The results can be used to estimate the quality of the PM results and identify data quality problems.

A.3.2.6 Privacy

In this category, we included papers that are related to privacy issues in event logs. Our research identified 10 studies related to this. Table A.6 presents the findings.

Table A.6 – Privacy

Study	**Description**	**Application**	**Findings**
Pika et al. (2019)	The paper analyses data privacy and utility requirements for healthcare process data and assesses the suitability of privacy-preserving data transformation methods to anonymize healthcare data.	Example healthcare log	A framework is proposed that can support process mining analyses of healthcare processes
Burattin, Conti and Turato (2015)	An approach is proposed which allows outsourcing of PM without thwarting the confidentiality of the dataset and processes.	Real-world log	A framework which is implemented and validated
Rafiei, Waldthausen and Aalst (2018a)	An approach is proposed that allows to hiding confidential information in a controlled manner while ensuring that the desired PM results can still be obtained	Real-life event log	A framework is proposed, implemented and tested.
Mannhardt, Petersen and Oliveira (2018)	An analysis is made of the privacy challenges of using process mining on data recorded from sensorized operators in human-centered industrial environments	None	A set of guidelines is obtained
Bauer et al. (2019)	The paper introduces ELPaaS, a web application that offers state-of-the-art techniques for event log sanitization and privacy-preserving PM queries	None	The users obtain event logs and process mining results that provide privacy guarantees such as differential privacy and k-anonymity
Fahrenkrog-Petersen, Aa and Weidlich (2019)	The paper address the risk of privacy-disclosure attacks on event logs with pseudonymized employee information	Real-world data	Experiments demonstrate the use of the framework
Mannhardt et al. (2019)	The paper set out to develop a protection model for event data privacy. It also shows at which stages of privacy leakages a protection model for event logs should be used.	Two real-life events logs	The general feasibility of the approach is demonstrated

Pika et al. (2020)	The paper analyses data privacy and utility requirements for healthcare process data and assess the suitability of privacy-preserving data transformation methods to anonymize healthcare data.	Three healthcare event logs	How some of the anonymization methods affect the results is demonstrated
Voigt et al. (2020)	The papers show how to quantify the re-identification risk with measures for the individual uniqueness in event logs.	A large collection of event logs	The results suggest that potentially up to all of the cases in an event log may be re-identified, which highlights the importance of privacy-preserving techniques in PM
Rafiei, Waldthausen and Aalst (2018b)	The paper proposes an approach that allows hiding confidential information in a controlled manner while ensuring that the desired process mining results can still be obtained.	Real-life event logs	The approach was evaluated and the results were satisfactory

Next, in Section A.4, we debate how the results are discussed.

A.4 Discussion of results

Section A.2 details the review process showing numbers at each of its steps. A initial consideration has to be done regarding the main focus of this research which is to identify the challenges and studies in event log preparation for process mining. Event log preparation is a broad and wide concept that, in a generic way, could be explained as everything that has to be done with event data *before* applying the main PM techniques (discovering, enhancement, conformance checking etc). Since this is a wide concept, not every paper address the preparation step directly as "log preparation" or "log pre-processing". This can be observed by the search strings that were used (Table A.1), and the results obtained from the automatic search. Initially 926 articles were identify through automatic search, but only 107 of them were found relevant after the selection process (screening by title and abstract and application of exclusion criteria). This is a tip of how difficult is to calibrate the search strings to identify the relevant studies involved in log preparation. The main contribution of this mapping study is to identify the *types* of log preparation techniques (extraction, repair, cleaning, etc) as shown in Figure A.5. A systematic literature review could be conducted for each of this types of preparation step to obtain a full list of studies in each of them. In short, this paper can be considered as a initial overview of log preparation types of challenges and techniques to solve them.

Section A.3 shows the results that enables raising answers regarding the RQ. Firstly, the more general data (quantitative results) are shown in Section A.3.1. As shown in Figure A.2, log preparation has been an addressed topic since 2006 and it has been growing in the last decade. 2018 was the most frequent year of publication, and since then, a degrowth has been observed. This degrowth can be explained as a result of COVID-19 Pandemic

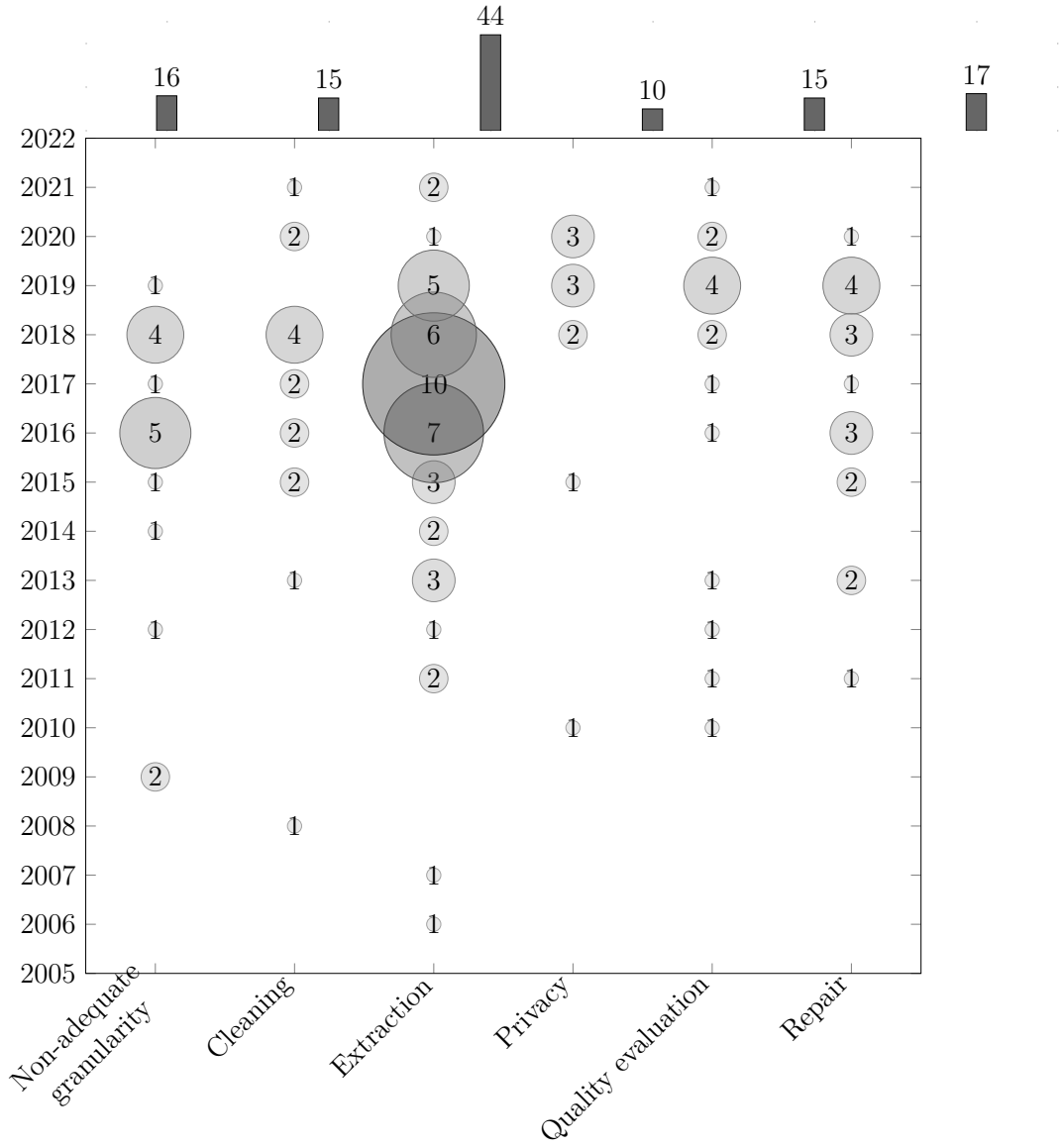


Figure A.5 – Studies classification over the years

mitigation of research projects, and also because the studies search was conducted at later 2021, when some of the paper were probably not available yet in the databases. Therefore, one can conclude that log preparation is still a relevant topic to be addressed and several research challenges are still at open.

Analysing the distribution in terms publication type, we found a significant sign of scientific maturity: 43% of studies were published in journals. The most frequent conference is "Int. Conf. in Business Process Management" and the most frequent journal is "Information Systems", which matches the most frequent venues found by related studies in process mining review studies presented in Section A.1. This means that this venues are considered the most important ones for process mining research in general and hence for log preparation specifically. When considering the distribution by countries and authors, we can observe a high concentration of studies (44%) around the two most frequent countries,

and 29% around the most frequent author indicating their relevance.

As shown in Table A.1 and Figure A.5, extraction is the most prominent task of log preparation observed in this study. The oldest articles found (2006 and 2007) are related to extraction, and also the majority (41%) of papers . Extraction still is a relevant topic, growing from 2006, peaking around 2017, and still having studies related to it. Starting on 2014, extraction papers have been published in journals in every year, indicating the maturity of the area. The most frequent keywords (besides "process mining" and "event log") are "edi", "erp system", "event log extraction", "inter-organizational business processes" and "ontology-based data access". These keywords indicate the main source from where the extraction is made in these articles. Also, it is possible to observe that, for each different source or system, research has to be made about extraction or conversion of data, since extraction is an important and inevitable step in log preparation. Non-adequate granularity articles spawn from 2009 to 2019 indicating a research topic in which a continuous exploration is made. The most frequent keywords (besides "process mining" and "event log") are "event mapping", "label refinement", and "abstraction levels" indicating the two most common solutions for granularity problems. Abstraction represents the majority (75%) of studies related to non-adequate granularity, pouting out that normally the problem of having low-level events is more common.

Cleaning related studies have also a continuous exploration since 2008. The maturity of the research can be observed by the percentage of journal articles (67%). The most frequent cleaning type is the detection and filtering of anomalous traces, but other types have been gaining attention in the last years. Repair is the second most frequent category of studies (16%). Repair of missing data and timestamp repair are the most frequent types of repair in the studies, indicating the central role that timestamp have specially in process discovery. Quality evaluation articles are in majority conference papers. This indicates a lack of maturity related to that category. Privacy article are the smaller category in this research (9%) and have been published from 2010 onward. The results indicate that is a increasing and concerning research area within PM.

A.5 Conclusion

PM is a robust tool that is increasingly being used to support managers; however, the quality of the final results depends on the reliability of the analyzed data. It is necessary to ensure that the event records used are carefully evaluated and treated to guarantee the expected quality. This is a phase of the PM that deals with preprocessing; it is a phase of event log preparation that is considered critical because it requires performing specific tasks for each log profile. This article provided a mapping review of the main approaches used in the event log preparation step for PM.

Six main categories were identified. The extraction category presents most articles and shows that event logs can be extracted from several systems and contexts. Non-adequate granularity category identifies the studies that use abstractions to handle too-fine-grained data, and refinements to handle very abstract data. The cleaning category shows studies that try to filter some aspect of the data such as anomalous traces or events. Repairing techniques can be used to insert missing data or to perform repairs in attributes, events, case ids, or timestamps. Quality evaluation articles aim to assert some aspects of the log such as completeness, activity labels, attributes, timestamps, and events. Privacy articles present approaches and techniques regarding the growing concern about the protection of sensible data. The results are shown quantitatively and qualitatively.

The purpose of this review was to serve as a guide for identifying problems related to the preparation of event logs and to provide descriptions of the methods, techniques, and frameworks applied to solve these types of problems found in PM and various related areas. Also, we believe this is a starting point for researchers to identify the studies that would help them prepare event logs for PM. As a future work, several directions can be explored. First, a full systematic literature review can be conducted for each of the six categories that were proposed by this article. Second, some new directions of PM such as the exploration of *concept drift* (SATO et al., 2021b) and *simulation* (FERRONATO et al., in press), require a (semi)-automatic way of preparing event logs. In that sense, a plausible future work would be the proposition of a framework for preparing event logs based on the problems and solutions shown in this paper.

APPENDIX B – Extraction of glossing rules

Based on the services provided to a patient in an appointment, auditors determine whether that service will be glossed, partially glossed, or not glossed. This determination is made through glossing rules. The objective of this task is characterized by the automatic extraction of glossing rules from a set of informed data. Based on this dataset, we want to extract a set of rules, each with confidence and a minimum number of occurrences required.

The dataset corresponds to a patient service event log. Given the set of events (activities) of each patient care (case) and their respective attributes, we want to create a system that suggests association glossing rules. The glossing rules are calculated by measuring the association between a set of services provided (antecedent) with the glossing performed (total, partial, or none).

An event log with 415192 events (17175 cases) provided by the partner company was used. Event log columns and data types are described in Frame B.1:

Frame B.1 – Data description for the extraction of glossing rules

Index	Name	Number of not null	type
0	ACOMODACA	415192	object
1	AUTO ID PRESTADOR	415192	int64
2	CODIGO ACOMODACA	338190	float64
3	CODIGO SERVIC	415192	int64
4	CONCLUID	415192	object
5	DATA VENCIMENT	414425	datetime64[ns]
6	DATA INCLUSA	415191	datetime64[ns]
7	DATA INTERNAMENT	338466	datetime64[ns]
8	DATA SOLICITACA	415192	datetime64[ns]
9	DESCRICAO SERVICO INF	315870	object
10	EMERGENCIA	415192	object
11	EVENTO PRINCIPAL CHAVE	415192	float64
12	FUNCAO PRESTADOR	415192	object
13	JUSTIFICATIVA MEDICA	8528	object
14	LIMITE AUDITORIA	202263	datetime64[ns]
15	MOTIV	14025	object
16	NUMERO DOCUMENT	415192	int64
17	NUMERO GUIA	415192	int64
18	PARECER	26	object
19	PARECER AUDITORIA	15099	object
20	QTDE USADA	415191	float64
21	QUANTIDADE APROVADA	415192	float64
22	QUANTIDADE AUTORIZADA	396680	float64
23	QUANTIDADE GLOSADA	134166	float64
24	QUIMI	415188	object
25	SERVIÇ	415192	object
26	SUBGRUP	315869	object
27	TIPO CONTA	415191	object
28	TIPO INTERNAMENT	415191	float64
29	TIPO SERVIC	415191	object
30	VIA ACESS	415187	object
31	VIA ACESSO AUTORIZAD	58310	object

Preprocessing operations were performed on the initial data set in order to obtain

the correct log. The variable CODIGO SERVICE was used to define the activities in the log, NUMERO GUIA to represent the case in the event log, and PARECER AUDITORIA as an attribute that defines the glossing. In addition, only the records that contained these three pieces of information were selected. The preprocessed event log contained 62426 events distributed across 2306 cases.

B.1 Apriori algorithm

Initially, the preprocessed log was converted into a table of boolean values, whose lines represent the 2306 cases, and the columns the occurrence (true or false) of a given service in that case. Other columns with relevant case attributes have also been added. Also, the last column of this table informs the result (glossed, not glossed, partially glossed) of the case reported by the auditor. The Apriori algorithm (AGRAWAL; SRIKANT et al., 1994) was used to scan this table in search of association rules.

With minimum support of 20%, 145540 association rules were found, but none with relevant structure, i.e., service that occurred that implies a gloss/non-gloss. With minimum support of 10%, the computer could not allocate the 30.8 GiB of memory needed to identify the rules.

Other tests were carried out, trying not to include all services, but the rules obtained were always in the order of thousands/millions, without rules that had an implication in gloss/non-gloss. The results are shown in Frame B.2:

Frame B.2 – Results summary for Apriori algorithm

Services used	# services	Table size	Min support	# extracted rules	# of relevant rules
100%	7362	2306 by 169434	20%	145540	0
100%	7362	2306 by 169434	10%	Memory fault	-
10%	736	1670 by 17006	20%	388 thousands	0
10%	736	1670 by 17006	15%	16,8 millions	0

Given the result obtained by the Apriori algorithm, it was decided to develop a personalized approach to extract the glossing rules. This approach is detailed in the following section.

B.2 Custom rule extraction

Each rule obtained by the Apriori algorithm has a set of antecedents that imply a set of consequences. These rules are calculated based on a specific support/number of occurrences. As the number of combinations is exponential and the number of columns is high, the applied algorithm did not obtain the desired results (described in the previous section). Thus, it was decided to create the rules in the desired format and evaluate them on the data set. Instead of analyzing the data, extracting rules, and filtering by the desired format,

it was decided to assemble the rules in a given format—focusing only on the attributes of interest to the decision—and evaluate them. The desired format depends on the type of rule one wants to extract:

1. Which occurrences of services automatically imply their glossing or partial glossing? Antecedent: occurrence of a given service. Consequence: total or partial glossing of this service.

2. Which occurrences of services automatically imply the glossing or partial glossing of another service? Antecedent: occurrence of a given service. Consequence: total or partial glossing of another service.

3. What occurrences of service and a certain quantity imply a partial or total glossing of the service? Antecedent: occurrence of a given service and a quantity. Consequence: total or partial glossing of this service.

For rules in format 1, 2317 antecedents were found in the log, with 2404 possible consequences, resulting in 2404 different rules. The confidence and support of the formed rules were measured over the log resulting in 674 rules with a minimum confidence of 80%. For rules in format 2, 1643 antecedents were found in the log, and 1730 possible consequences resulted in 2842390 different rules. The confidence and support of the formed rules were measured over the log resulting in 25 rules with a minimum confidence of 80%. For rules in format 3, 8850 antecedents were found in the log, and 1730 possible consequences resulted in 9261 different rules. The confidence and support of the formed rules were measured over the log resulting in 1313 rules with a minimum confidence of 80%.

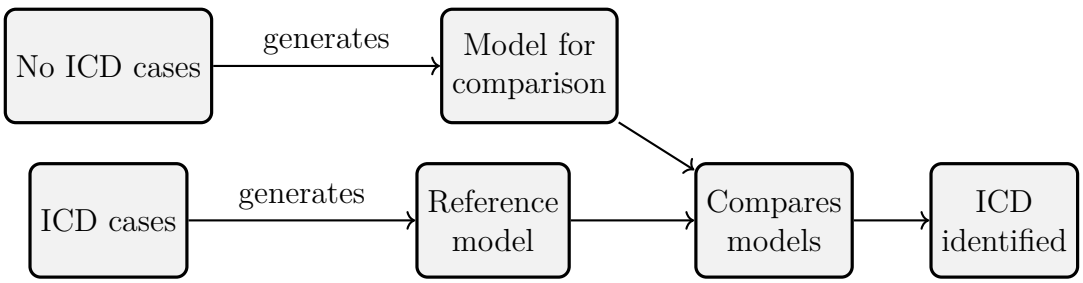
APPENDIX C – Inferring missing ICDs

Based on the services provided to a patient during an appointment, a given ICD is associated with it by a healthcare professional. However, the ICD information is often not entered due to external factors. This task consists of creating a system that automatically fills in ICDs for cases where this information is missing. For this, the history of cases in which ICD was informed is used. The provided data set has 189679 events distributed in 22100 cases. Initially, the data set was separated into "cases with ICD" (21109 cases) and "cases without ICD" (991 cases). In summary, each event record has the following information:

- Appointment: identification code for the appointment of a specific patient. It is used to identify the case.
- ICD: Some cases do not have this information. This is the information we want to predict in cases where it is missing.
- Clinic: type of clinic where the case occurred. It can be medical, orthopedic, or surgical
- Item: the activity that was performed. It can be the name of a medical procedure, a drug or material prescribed, a test, screening, discharge, or consultation
- Start Date: Timestamp defining the start of the activity
- End Date: Timestamp defining the end of the activity

From the separation of cases into cases with ICD and cases without ICD, the steps described in Figure C.1 were performed: generation of models, comparison, and identification of the probable ICD. Three different approaches were used, and the details are presented in the next three sections.

Figure C.1 – ICDs Suggestion Scheme



C.1 Classification problem

Initially, the log was converted into a table of boolean values, whose lines represent each case, and the columns the occurrence (true or false) of a given item in that case. Other columns with relevant case attributes have also been added: medical clinic, start date, and end date. Also, the last column of this table informs the case's ICD.

The task was modeled as a classification problem. Each case corresponds to an instance, the ICD is the class with which that instance is associated. The Weka platform (FRANK; HALL; WITTEN, 2016) was used to perform the classification experiments. 799 attributes (occurrence or not of item) and 969 classes (different ICDs) were used. Cross-validation with 10 partitions was used to verify the quality of the classification models.

The classifier "weka.classifiers.lazy.IBk" showed the best results. However, these results were unsatisfactory given the problem:

- Correctly Classified Instances 22.1138
- Incorrectly Classified Instances 77.8862
- Kappa statistic 0.1601
- Mean absolute error 0.0018
- Root mean squared error 0.032
- Relative absolute error 88.918
- Root relative squared error 101.3105

C.2 Fitness problem

The second approach used process discovery techniques and fitness analysis (Definition 7) between the event log and discovered process. The PM4PY library (PM group of Fraunhofer FIT, 2021) for implementing Process Mining algorithms in Python was used for this.

Initially, the log with events with ICD was divided into several logs, one for each ICD. The inductive process mining technique was applied to each of these logs, thus building a process model for each ICD.

Next, the fitness of each case without ICD was calculated and compared to each discovered ICD process model. The intention is to identify the process model and its respective ICD in which that case best fits, that is, it has a greater fitness.

Due to each case's high fitness processing time, the approach proved unfeasible. Note that this approach is the only one that considers the order of occurrence of activities in one case for inferring the ICD, while in the others (sections C.1 and C.3), the order is ignored, considering only the occurrence or not of such activities.

C.3 Similarity problem

Cases with informed ICDs were transformed into a vector that stores service occurrence information and the ICD associated with the case. The equal vectors, the cases with the same set of activity occurrences and the same ICD, were considered as a profile. Each profile created has the following information:

- Services (activities) that occurred in the case;
- ICD informed for the case; and
- Number of cases that have those services and that ICD.

Next, cases with missing ICDs were converted into a service occurrence vector, and each case was compared with the profiles created in the previous item. The comparison was made by calculating the Jaccard distance (TAN; STEINBACH; KUMAR, 2016). The ICDs of the three most similar profiles were suggested as the ICDs of that case.

The results obtained were satisfactory in the sense of comparison, that is, it was possible to predict the ICD of each case without ICD, such that the set of activities of that case without ICD is the same as the set of activities of a profile (set of cases with ICD). However, in discussion with a specialist in the area, it was observed that the ICD suggestion task does not depend solely on the activities carried out. According to her, it is impossible to state a certain ICD for a case because the profiles created do not have an adequate level of characterization to characterize each of the ICDs. The following section describes some results to elucidate these details.

C.4 Final considerations

Some identical sets of activities are associated in the dataset with different ICDs. Here, the profiles already disregard activities that always occur (discharge, screening, password withdrawal, consultation).

For example, the activity "Electrocardiogram Ecg" in "Clinic Medical" may be associated with several different types of ICDs (F, H, I, K, M, R, etc.). This means that there are many possible classes (ICDs) for the occurrence of a few attributes.

It is noticed that even combining the type of clinic and the activities carried out with the patient, it is impossible to determine the corresponding ICD precisely. Attributes are not very descriptive in this sense. The amount of attributes (799) is less than the number of classes (969). The attributes are all boolean, indicating the occurrence or not of an activity, thus being less descriptive than a numerical or categorical attribute. These characteristics are the main contributors to classification inaccuracy. Another secondary feature is the high number of paths leading to the same outcome. Some ICDs are associated with several different sets of activities. This leads also to classification inaccuracy.

Gottlieb et al. (2013) assesses the ability of a large corpus of electronic records to predict ICDs. It presents a method that exploits patient similarities along multiple dimensions to predict disease codes. It uses baseline demographic, blood, and electrocardiogram measurements and medical histories from patients hospitalized at two independent hospitals. Thus, it obtained 86% accuracy in cross-validation for the main categories of diseases, including infectious and parasitic diseases, endocrine and metabolic diseases, and diseases of the circulatory system. In this sense, some recommendations are made so that the provided data set can be updated for the proposed classification task:

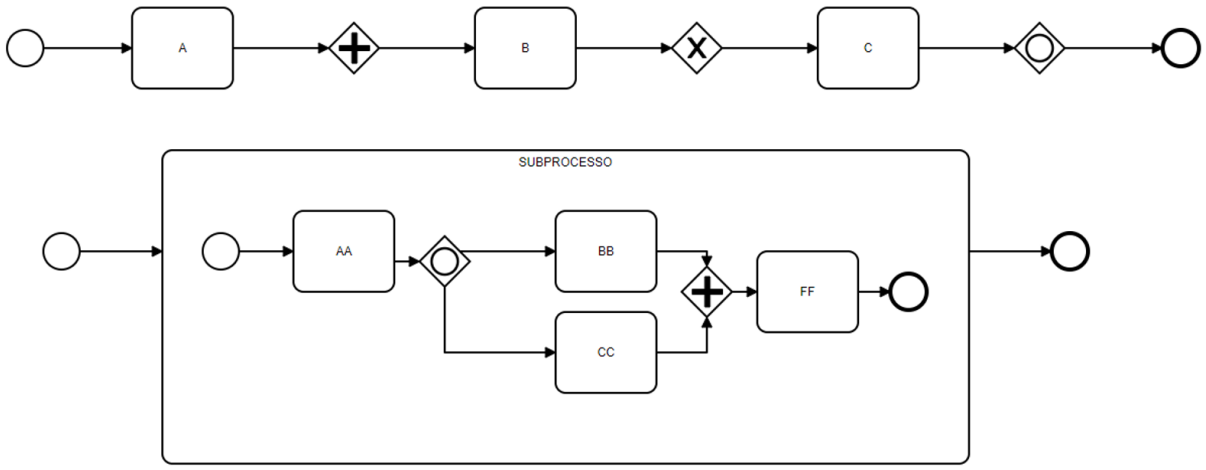
- Increase the number of attributes concerning the number of classes
- Decrease the number of classes. One solution is to detect a reduced set of classes. With that, the number of attributes available would probably be able to detect this reduced group of classes with greater precision;
- Increase the number of attributes. If it is not of interest to reduce the number of classes, it is possible to increase the number of attributes, including demographic information, laboratory test results, and other types of information that may contribute to the detection of classes. It is recommended that non-Boolean attributes be incorporated (such as age, sex, number of medications, etc.) to be used to enrich the amount of information;

APPENDIX D – Conversion of process models

This task aims to convert process models in the Upflux standard to the BPMN standard. Figure D.1 presents an example of BPMN. This example shows two processes. The first process, at the top, has the start tag, activity A, a parallel gateway, activity B, an exclusive gateway, activity C, an inclusive gateway, and the end tag.

The second process, at the bottom, has the start tag, a subprocess, and the end tag. The subprocess comprises a start tag, the AA activity, an inclusive gateway, the BB and CC activities, a parallel gateway, the FF activity, and the end tag.

Figure D.1 – Example of BPMN to be obtained



Upflux's process model is defined in a JSON object. The JSON fields used in the representation of the process model are:

- name: Model name;
- nodes: vector of one or more activities in the process model. Process start and end tags and subprocesses are also nodes. Each node has a set of attributes:
  - Name: text identifying the activity or subprocess. In the case of start and end tags, this field is "start" or "end" respectively;
  - Id: identifier of the activity or subprocess. In the case of start and end tags, this field is a negative integer;
  - IsMacroActivity: boolean that is set to true if the node is a node at the start of a process, that is, a subprocess;

  - MacroActivityId: if the node belongs to a subprocess, the subprocess identifier is provided by this attribute.

- gateways: array of zero or more gateways in the process model. Each gateway has a set of attributes:
  - Type: gateway type that can be “Parallel”, “Exclusive”, or “Inclusive”;
  - Id: gateway identifier - a negative integer;
  - MacroActivityId: if the gateway belongs to a subprocess, the subprocess identifier is provided by this attribute.

- transitions: array of zero or more transitions in the process model. Each transition has a set of attributes:
  - Actual: ID of the element from which the transition starts;
  - ActualType: type of element from which the transition starts. It can be Node or Gateway;
  - Next: ID of the element in which the transition arrives;
  - NextType: type of element on which the transition arrives. It can be Node or Gateway;
  - MacroActivityId: if the transition belongs to a subprocess, the subprocess identifier is provided by this attribute.

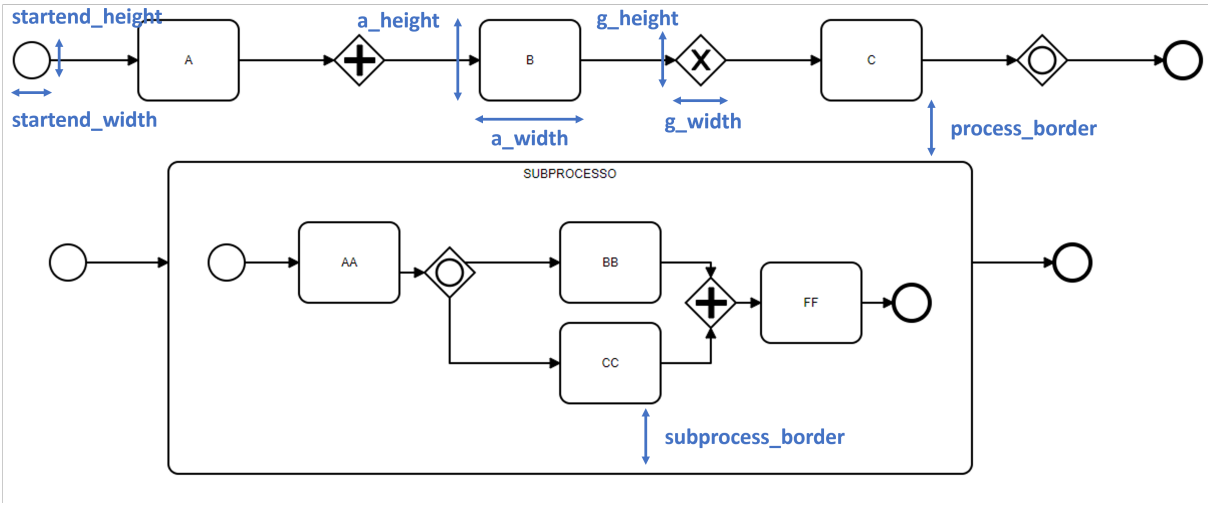
Based on this JSON file specifying the process elements, the proposed solution creates the elements in BPMN and automatically calculates the position and size of each element in BPMN. For this, some drawing parameters are specified (in pixels) and can be changed if necessary:

- a-width = 200, defines the width of each activity in BPMN;
- a-height = 150, defines the height of each activity in BPMN;
- g-width = 40, defines the width of each gateway in BPMN;
- g-height = 40, defines the height of each gateway in BPMN;
- startend-width = 40, defines the width of each start/end element in BPMN;
- startend-height = 40, defines the height of each start/end element in BPMN;
- subprocess-border = 40, defines the distance between the elements of a subprocess and the border of the element that contains them, that is, of the subprocess.

- process-border = 40, defines the distance between the different processes in case there is more than one process in the drawing.

Figure D.2 shows the configuration measurements in the previous example.

Figure D.2 – Diagram of inputs



A BMPN model is specified in an XML file based on the following definitions:

- xsi="http://www.w3.org/2001/XMLSchema-instance"
- bpmn="http://www.omg.org/spec/BPMN/20100524/MODEL"
- bpmndi = "http://www.omg.org/spec/BPMN/20100524/DI"
- dc = "http://www.omg.org/spec/DD/20100524/DC"
- di="http://www.omg.org/spec/DD/20100524/di"

Initially, the process element is created. Next, the start, end, activity, and subprocess elements are created based on the nodes specified in the input JSON. Then the gateways are created based on the input JSON gateways. Finally, sequenceFlow elements are created based on the input JSON transitions.

In addition to specifying elements, the BPMN file has specifications on how the model design should be done. Each element (start, end, activity, subprocess, and gateways) must have defined the XY coordinate of the upper left corner, the width, and the height. For transitions defined by sequenceFlow elements, it is necessary to define the XY coordinate where the transition starts and the XY coordinate where it arrives. Note that a transition must start exactly from the edge of one element and end at the edge of another element.

Each element's X and Y coordinates are calculated automatically based on the configuration parameters. Furthermore, a subprocess element has its variable size according

to its content. The elements belonging to each process/sub-process are separated, and the position of each element relative to that process/sub-process is calculated by Sugiyama's algorithm (EIGLSPERGER; SIEBENHALLER; KAUFMANN, 2004). Next, elements belonging to subprocesses are moved to be accommodated within the subprocess element. The subprocess element is then sized to accommodate all of its content. The different processes are drawn, one below the other. Finally, each transition's departure and arrival XY coordinates are calculated based on the position of the elements.

APPENDIX E – Questionnaire applied for validation

	Statement	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
1	It is interesting to include other factors, like cost, to analyze my process.	○	○	○	○	○
2	The overall results are not within the expected.	○	○	○	○	○
3	The multifactor model can model the desired factors.	○	○	○	○	○
4	The annotated log does not seem to be correctly annotated.	○	○	○	○	○
5	It is useful to color a process model based on a factor.	○	○	○	○	○
6	The color model does not seem to be correctly colored.	○	○	○	○	○
7	It is useful to perform a conformance check considering factors' values.	○	○	○	○	○
8	It is not useful to split the log into conform and non-conform logs.	○	○	○	○	○
9	The conformance component results seem to be correct.	○	○	○	○	○
10	The report results are not within the expected.	○	○	○	○	○
11	The Factor vs. numerical chart is useful.	○	○	○	○	○
12	The Factor vs. categorical column chart is not useful.	○	○	○	○	○
13	The Factor vs. categorical heatmap chart is useful.	○	○	○	○	○
14	The histogram chart is not useful.	○	○	○	○	○
15	The Statistical analysis is useful.	○	○	○	○	○
16	The Factor vs. numerical chart does not seem to be correct.	○	○	○	○	○
17	The Factor vs. categorical column chart seems to be correct.	○	○	○	○	○
18	The Factor vs. categorical heatmap chart does not seem to be correct.	○	○	○	○	○
19	The histogram chart seems to be correct.	○	○	○	○	○
20	The Statistical analysis does not seem to be correct.	○	○	○	○	○
21	The report component is useful.	○	○	○	○	○
22	The Prediction and Recommendation component does not seem to be correct.	○	○	○	○	○
23	The Prediction and Recommendation component is useful.	○	○	○	○	○
24	Cases were not correctly represented in data mining format.	○	○	○	○	○
25	The Data Mining component is useful.	○	○	○	○	○

Frame E.1 – Questionnaire answers

Question	Specialist			Process mining researchers and graduate students						
	CS1	CS2	CS3							
1	3	4		4	4	4	4	4	4	4
2	4	4								
3	4	4		4	4	0	3	4	2	3
4	4	4								
5		4		4	4	4	3	4	3	4
6		4								
7		4		4	2	2	3	0	4	4
8		4		4	4	4	3	3	4	4
9		4								
10	4	4								
11	4	4		4	4	4	4	4	3	3
12	3	4		4	4	4	0	0	4	4
13		4		4	4	3	2	3	3	4
14	4	4		4	4	3	3	0	4	4
15		4		4	4	4	2	4	4	4
16	4	4								
17	3	4								
18		4								
19	4	4								
20		4								
21	3	4		4	4	1	2	4	4	4
22		4								
23		4		4	4	4	3	4	2	4
24			3							
25			4	4	4	0	2	4	2	4