Process discovery, analysis and simulation of clinical pathways using health-care data

Martin Prodel

NNT : 2017LYSEM009

THÈSE

présentée par

Martin PRODEL

pour obtenir le grade de
Docteur de l'École Nationale Supérieure des Mines de Saint-Étienne

Spécialité : Génie Industriel

MODÉLISATION AUTOMATIQUE ET SIMULATION DE PARCOURS DE SOINS À PARTIR DE BASES DE DONNÉES DE SANTÉ

soutenue à Saint-Etienne, le 10 avril 2017

Membres du jury

Président :	Farouk TOUMANI	Professeur, LIMOS, Clermont-Ferrand
Rapporteurs :	M. Andrea MATTA	Professeur, Politecnico di Milano, Italie
	Mme Maria DI MASCOLO	Directrice de Recherche CNRS, Grenoble INP
Examinateurs :	M. Philippe LENCA	Professeur, IMT Atlantique, campus de Brest
	M. Farouk TOUMANI	Professeur, LIMOS, Clermont-Ferrand
Directeur de thèse :	M. Xiaolan XIE	Professeur, Mines Saint-Etienne
Co-directeur :	M. Vincent AUGUSTO	Maître de Recherche, Mines Saint-Etienne
Invités :	M. Ludovic LAMARSALLE	Dirigeant, Pharm.D, MSc, HEVA, Lyon
	M. Baptiste JOUANETON	MSc, HEVA, Lyon

Spécialités doctorales	Responsables :	Spécialités doctorales	Responsables
SCIENCES ET GENIE DES MATERIAUX	K. Wolski Directeur de recherche	MATHEMATIQUES APPLIQUEES	O. Roustant, Maître-assistant
MECANIQUE ET INGENIERIE	S. Drapier, professeur	INFORMATIQUE	O. Boissier, Professeur
GENIE DES PROCEDES	F. Gruy, Maître de recherche	IMAGE, VISION, SIGNAL	JC. Pinoli, Professeur
SCIENCES DE LA TERRE	B. Guy, Directeur de recherche	GENIE INDUSTRIEL	A. Dolgui, Professeur
SCIENCES ET GENIE DE L'ENVIRONNEMENT	D. Graillot, Directeur de recherche	MICROELECTRONIQUE	S. Dauzere Peres, Professeur

EMSE : Enseignants-chercheurs et chercheurs autorisés à diriger des thèses de doctorat (titulaires d'un doctorat d'État ou d'une HDR)

ABSI	Nabil	CR	Génie industriel	CMP
AVRIL	Stéphane	PR2	Mécanique et ingénierie	CIS
BALBO	Flavien	PR2	Informatique	FAYOL
BASSEREAU	Jean-François	PR	Sciences et génie des matériaux	SMS
BATTAIA-GUSCHINSKAYA	Olga	CR	Génie industriel	FAYOL
BATTON-HUBERT	Mireille	PR2	Sciences et génie de l'environnement	FAYOL
BERGER DOUCE	Sandrine	PR2	Sciences de gestion	FAYOL
BIGOT	Jean Pierre	MR(DR2)	Génie des Procédés	SPIN
BILAL	Essaid	DR	Sciences de la Terre	SPIN
BLAYAC	Sylvain	MA(MDC)	Microélectronique	CMP
BOISSIER	Olivier	PR1	Informatique	FAYOL
BONNEFOY	Olivier	MA(MDC)	Génie des Procédés	SPIN
BORBELY	Andras	MR(DR2)	Sciences et génie des matériaux	SMS
BOUCHER	Xavier	PR2	Génie Industriel	FAYOL
BRODHAG	Christian	DR	Sciences et génie de l'environnement	FAYOL
BRUCHON	Julien	MA(MDC)	Mécanique et ingénierie	SMS
BURLAT	Patrick	PR1	Génie Industriel	FAYOL
COURNIL	Michel	PR0	Génie des Procédés	DIR
DAUZERE-PERES	Stéphane	PR1	Génie Industriel	CMP
DEBAYLE	Johan	CR	Image Vision Signal	CIS
DELAFOSSE	David	PR0	Sciences et génie des matériaux	SMS
DELORME	Xavier	MA(MDC)	Génie industriel	FAYOL
DESRAYAUD	Christophe	PR1	Mécanique et ingénierie	SMS
DOLGUI	Alexandre	PR0	Génie Industriel	FAYOL
DRAPIER	Sylvain	PR1	Mécanique et ingénierie	SMS
FAVERGEON	Loïc	CR	Génie des Procédés	SPIN
FEILLET	Dominique	PR1	Génie Industriel	CMP
FRACZKIEWICZ	Anna	DR	Sciences et génie des matériaux	SMS
GARCIA	Daniel	MR(DR2)	Génie des Procédés	SPIN
GAVET	Yann	MA(MDC)	Image Vision Signal	CIS
GERINGER	Jean	MA(MDC)	Sciences et génie des matériaux	CIS
GOEURIOT	Dominique	DR	Sciences et génie des matériaux	SMS
GONDRAN	Natacha	MA(MDC)	Sciences et génie de l'environnement	FAYOL
GRAILLOT	Didier	DR	Sciences et génie de l'environnement	SPIN
GROSSEAU	Philippe	DR	Génie des Procédés	SPIN
GRUY	Frédéric	PR1	Génie des Procédés	SPIN
GUY	Bernard	DR	Sciences de la Terre	SPIN
HAN	Woo-Suck	MR	Mécanique et ingénierie	SMS
HERRI	Jean Michel	PR1	Génie des Procédés	SPIN
KERMOUCHE	Guillaume	PR2	Mécanique et Ingénierie	SMS
KLOCKER	Helmut	DR	Sciences et génie des matériaux	SMS
LAFOREST	Valérie	MR(DR2)	Sciences et génie de l'environnement	FAYOL
LERICHE	Rodolphe	CR	Mécanique et ingénierie	FAYOL
LI	Jean-Michel		Microélectronique	CMP
MALLIARAS	Georges	PR1	Microélectronique	CMP
MAURINE	Philippe	Ingénieur de recherche	Microélectronique	CMP
MOLIMARD	Jérôme	PR2	Mécanique et ingénierie	CIS
MONTHEILLET	Frank	DR	Sciences et génie des matériaux	SMS
MOUTTE	Jacques	CR	Génie des Procédés	SPIN
NEUBERT	Gilles	PR	Génie industriel	FAYOL
NIKOLOVSKI	Jean-Pierre	Ingénieur de recherche		CMP
NORTIER	Patrice	PR1		SPIN
OWENS	Rosin	MA(MDC)	Microélectronique	CMP
PICARD	Gauthier	MA(MDC)	Informatique	FAYOL
PIJOLAT	Christophe	PR0	Génie des Procédés	SPIN
PIJOLAT	Michèle	PR1	Génie des Procédés	SPIN
PINOLI	Jean Charles	PR0	Image Vision Signal	CIS
POURCHEZ	Jérémy	MR	Génie des Procédés	CIS
ROBISSON	Bruno	Ingénieur de recherche	Microélectronique	CMP
ROUSSY	Agnès	MA(MDC)	Génie industriel	CMP
ROUSTANT	Olivier	MA(MDC)	Mathématiques appliquées	FAYOL
ROUX	Christian	PR	Image Vision Signal	CIS
STOLARZ	Jacques	CR	Sciences et génie des matériaux	SMS
TRIA	Assia	Ingénieur de recherche	Microélectronique	CMP
VALDIVIESO	François	PR2	Sciences et génie des matériaux	SMS
VIRICELLE	Jean Paul	DR	Génie des Procédés	SPIN
WOLSKI	Krzystof	DR	Sciences et génie des matériaux	SMS
XIE	Xiaolan	PR1	Génie industriel	CIS
YUGMA	Gallian	CR	Génie industriel	CMP

ENISE : Enseignants-chercheurs et chercheurs autorisés à diriger des thèses de doctorat (titulaires d'un doctorat d'État ou d'une HDR)

BERGHEAU	Jean-Michel	PU	Mécanique et Ingénierie	ENISE
BERTRAND	Philippe	MCF	Génie des procédés	ENISE
DUBUJET	Philippe	PU	Mécanique et Ingénierie	ENISE
FEULVARCH	Eric	MCF	Mécanique et Ingénierie	ENISE
FORTUNIER	Roland	PR	Sciences et Génie des matériaux	ENISE
GUSSAROV	Andrey	Enseignant contractuel	Génie des procédés	ENISE
HAMDI	Hédi	MCF	Mécanique et Ingénierie	ENISE
LYONNET	Patrick	PU	Mécanique et Ingénierie	ENISE
RECH	Joël	PU	Mécanique et Ingénierie	ENISE
SMUROV	Igor	PU	Mécanique et Ingénierie	ENISE
TOSCANO	Rosario	PU	Mécanique et Ingénierie	ENISE
ZAHOUANI	Hassan	PU	Mécanique et Ingénierie	ENISE

Mise à jour : 29/06/2015

Acknowledgments

Je tiens tout d'abord à remercier les membres de mon jury, dont M. Toumani pour avoir présidé ce jury, Mme Maria Di Mascolo et M. Andrea Matta pour avoir accepté de rapporter cette thèse avec rigueur, M. Philippe Lenca et M. Farouk Toumani pour leurs regards critiques sur ces travaux, ainsi que mes chers encadrants.

Je remercie grandement mon mentor intellectuel, le Prof. Xiaolan Xie, qui a endossé le rôle de directeur de thèse. Au-delà de la grande confiance qu'il m'a accordée dans mon travail, ce sont surtout ses intuitions et sa rigueur scientifique qui m'ont montré la voie.

Je quitte le monde académique pour remercier deux personnes sans qui rien n'aurait été possible (faute de financement tout simplement) : Alexandre Vainchtock et Ludovic Lamarsalle, les deux fondateurs visionnaires de la société HEVA. Je les remercie d'avoir cru en cette aventure depuis le début et d'avoir énormément contribué, chacun bien à sa faon, à sa réussite. Je remercie également Baptiste Jouaneton pour m'avoir pris sous son aile et m'avoir tout appris sur les données PMSI (en plus du crawl à nos heures perdues). Enfin, j'ajoute une mention spéciale à mes autres collègues d'HEVA pour la qualité de l'ambiance qui règne du lundi matin au vendredi soir.

Je remercie également les personnes de l'Ecole des Mines qui ont embarqué dans le même bateau de la recherche que moi : mes compères doctorants d'I4S, notamment Sabri (en souvenirs des chants russes et de la cohabitation de bureau) et Thomas (pour le congrès ensemble à Londres); un immense merci à Thierry pour des tonnes de choses (l'accueil dans une maison ferme , les footing interminables et les soirées ping); merci également à cette formidable équipe de recherche pour les innombrables soirées débats scientifiques jeux de plateaux, avec plus récemment la participation de Raksmey. Enfin, merci à Amélie pour son extrême gentillesse et son efficacité professionnelle dans tous nos échanges.

Le paragraphe que je dédie maintenant est totalement disproportionné au regard des autres, mais il illustre très justement l'importance du rôle endossé au cours de ces 3 ans (voire 6 en tout si on remonte à notre première rencontre), par Vincent. Merci de m'avoir tout appris pendant ma formation d'ingénieur, mais aussi pour avoir réussi à me donner le goût (très dangereux) de la recherche. Vincent a su déployer un arsenal d'arguments pour me convaincre de démarrer cette thèse. Et c'est en regardant là où nous en sommes arrivés aujourd'hui que je peux affirmer, comme dirait Edith, que je ne regrette rien. Vincent a rempli son rôle d'encadrant sur les aspects scientifiques et bien plus encore : nous avons appris ensemble, et malgré nous, à être tenace dans l'adversité (Yedo et PG). Et surtout, nous avons conquis l'Amérique ensemble avec notre article WSC 2016. En résumé, je peux dire sans me tromper que nous avons gravi des montagnes ensemble!

Je n'oublie bien sûr pas ma très chère famille, à commencer par les Stéphanois, Yves et Florence, qui ont toujours été là quand j'ai eu besoin d'eux (notamment quand j'ai perdu mon toit pendant la dernière année de thèse). Je remercie éternellement mes 2 parents pour tout, et plus particulièrement pour leur envie insatiable de comprendre mon travail de thèse. J'ajoute une mention spéciale à ma sister, brillante pharmacienne en devenir, pour ne jamais avoir douté de moi. Je finis mes remerciements pour celle qui occupe toutes mes pensées, Marie. Tu as eu le courage de me suivre au bout du monde (Saint-Héand, Saint-Etienne, Lyon, Chegaga) et de m'insuffler l'énergie nécessaire aux bons moments.

We rushed on actual gold in the 19th century,
We drew out the black gold (oil) in the 20th century,
We realized the existence of a blue gold (water) at the dawn of the 21st century,
Let's rush and tap into an ever-growing, uncolored and life-saving gold called health data[1].

[1] Personal adaptation of Clive Humby's "data is the new oil" in 2006

Table of contents

List of Figures

List of Tables

Introduction

Health-care systems

This section introduces the topics of health-care expenditures, the possible benefits of industrial engineering, of clinical pathways and of health data.

Health expenditure and industrial engineering

Public health expenditures represent 5.99% of the world annual gross domestic product (World Health Organization, year 2014). Health-care is a major concern in most developed countries. It represents a tremendous part of the gross domestic product: 11.5% in France, 17.1% in the United States, 9.1% in the United Kingdom, 11.3% in Germany, 9.3% in Italy and 5.6% in China (WHO, year 2014). However, during the last two decades, hospitals have undergone major changes: faced with an increasingly severe socio-economic context, they had to comply with new management rules in order to minimize the costs while maintaining a certain quality of service. Although these two objectives are well-known antagonists, solutions exist.

Over the same past twenty years, the aim of scientific studies has been to bring substantial gains in terms of efficiency and productivity to health-care systems. It passes through the setting up of more efficient organizations, while improving the quality of care. The application of scientific methods from the field of Industrial Engineering is an excellent approach to achieve this objective. Industrial engineering techniques are widespread in many sectors, from manufacturing industry to service industries, and have shown their value in optimizing processes on many occasions. However, these approaches are challenging because, while industries and hospitals are similar in many respects, they differ in a number of crucial points: patients replace products and doctors take the role of machines. Difficulties related to the application of methods from the field of industrial engineering to a health-care environment are multiple:

- The analysis of a health-care system is closely linked to the observation and the modeling of patient flows, not products. It is difficult to predict a patient's care pathway within a hospital system because it depends on multiple factors such as biological interactions, a pathology and a care management strategy.
- Activities of care providers are very diversified, which requires a high capacity to adapt to the demand: emergency is a recurring notion which is at the origin of most organizational problems. Finally, health-care environments are highly stochastic (random processes are at work), making long-term planning more difficult.
- Health-care systems are made up of a multitude of subsystems (hospitals, general practitioners, pharmacy, etc.) that are generally compartmentalized and not well coordinated.

Tremendous efforts and works have been carried out to cross the gap in that direction. Today, in 2017, the question of whether or not the scientific community can help health-care stakeholders to provide better care is no longer on the table. The question is to know how.

Clinical pathways

Currently, in health-care organizations, a major trend for the improvement of care quality while reducing costs is the design and implementation of Clinical Pathways. A clinical pathway (CP) can be defined as a structured and multidisciplinary care plan used to detail essential steps and timing in the care of patients with a specific clinical problem (Rotter et al., 2010). CPs are used as a tool for a standardization of clinical processes. They represent an opportunity to reduce variability in the delivery of cares. CPs usually involve many stakeholders (physicians, managers, nurses, pharmacists, specialists, etc.) because they rely on good coordination and communication of the care givers. The design and the implementation of CPs in practice require a level of standardization for medical treatment processes, the training and education of young medical professionals, the implementation of health information systems and the automated analysis for the purpose of process optimization. Then, it is possible to provide a model (i.e. a guideline) of a CP.

A second way to describe a CP is to adopt the patient's point of view. A CP is the ordered sequence of medical events that happen to an individual patient. In that regard, each CP is unique and corresponds to a patient's medical history. There exist different levels of description of a CP. The highest level is to consider that a CP starts the first time that a patient is taken care of and ends when he/she passes away (several years later). An intermediary level is the description of the steps occurring between the entrance in a hospital emergency room and the discharge (several hours to days later). A short-time view of a CP would be to identify the different phases of a surgery (a minute-by-minute process). We see here that there is no limitation in space, in time or in the concept that are included in a CP.

So far, most CP models have been designed by medical experts of each field. Hence, it heavily relies on experts' opinions and on how they perceive their practices. The reality of what actually happens may differ from such references. The design of a CP is a major challenge to better understand the impact of treatments on the whole journey of the patient. Health authorities intend to propose standardization of care processes for various operational purposes: organization of care activities, assignment of human resources, reducing practice variability, minimizing delays in treatments or decreasing costs while maintaining quality. Today, there is a will to go further than experts' opinions to answer these challenges. As such, evidence-based medicine has become paramount to medical decision making and clinical judgment. The current trend is to use electronic data as the new objective source for clinical pathway description.

Health data in the 21^{st} century

During the last two decades, the amount of data collected in hospital information systems has increased exponentially. As other domains before it, health-care has been struck by computerization. Still, new technologies are generally slow to spread in health-care systems, which explains why computerization remains an ongoing process, especially in community medicine (outside hospitals). Hospitals are the most advanced organizations regarding the collection and the storage of health data. Over several countries, many hospitals are reporting databases containing the individual data of millions of patients. However, hospitals are now facing the need for strong analytic skills to take advantage of these massive data.

For a long time, health data have been considered too sensitive to be extensively shared and analyzed. Indeed, health databases contain critical information about each ill person and are considered as personal

data (they belong to the patient). The most advanced "anonymization" algorithms have been deployed to hide personal information from health database, so that they could be used for research purposes. The 2016 French law about the "modernization of the French health system" is a national initiative which illustrates the ongoing trend toward an opening of health data. A responsible use of health data represents a big opportunity to improve health-care systems. The large amount of data collected in hospital information systems is valuable because it may reveal important patterns of clinical pathways, allowing the creation of realistic models.

Scientific objectives

The main objective of this thesis is to develop a complete methodology, based on mathematical models, to automatically create clinical pathway models from large health databases. The resulting models can be used as new references of what actually happened. They provide the ground foundations for a better knowledge of health processes and allow for the identification of promising improvements. The strength of such clinical pathway models comes from the use of databases containing a large number of patient data. They can successfully be used to answer domain-specific questions. More specifically, our main objective can be split up in 4 sub-objectives:

1. **Developing an optimal process discovery algorithm capable of dealing with variable and heterogeneous data**: the discovered clinical pathway models must balance two opposing criteria, being as small as possible (low complexity) and as much representative as possible (high quality). The proposed approach will show the benefits of combining combinatorial optimization and process mining techniques.

2. **Proposing a health-care analytic toolbox to address 3 specific problems related to clinical pathway modeling**: decision point analysis, definition of a similarity score for two events and a 2-sequence global alignment method. A clinical pathway is made of choices which depend on each patient's condition and medical history. Classic probabilistic models are not sufficient to discover such complex patterns and interactions. We combine a sequence alignment method with a data mining algorithm to perform that task.

3. **Solving domain-related questions with an actionable model of clinical pathway**. We propose a new class of state chart to convert a static model of clinical pathways into a simulation model. It enables the evaluation of new care management scenarios.

4. **Proposing a methodological framework capable of performing the above mentioned points automatically** (apart from the initial data preparation, no hand-made interventions are needed). This guarantees the re-use of the approach on any disease case study and with new data sources.

Thesis outline

This thesis is made up of 6 chapters.

Chapter 1 provides a broad review of the literature on the modeling and simulation of health-care systems over the past decade. A focus is made on data-driven approaches, especially process mining. This state of the art is crucial as it allows us to identify the unsolved challenges related to health-care environments and data, which we intend to answer through this thesis.

Chapter 2 presents the methodological flow-chart of this thesis. It explains how we automatically turn raw data from a database of events into a simulation model in a step-by-step approach. For each step, we describe the required inputs, the scientific challenges, our proposal to address these challenges and the generated outputs. Finally, the originality and the scientific contributions of the present thesis are introduced.

Chapter 3 addresses the problem of process discovery from large and complex event logs. It includes a mathematical programming model based on a novel hierarchical structuration of the event logs. Desired properties of an optimal process model are described. A combination of Monte-Carlo optimization and tabu search is proposed to overcome the complexity related to the huge size of event logs and the combinatorial solution space. Numerical results show that our approach performs better than state-of-the-art approaches.

Chapter 4 addresses the problem of enriching a process model which represents a clinical pathway. We specifically focus on the study of two perspectives: the decision point analysis and the time perspective. The decision point problem aims at finding relations between data attributes and the routing choices in the process. We present the challenge that we face when using a noisy and heterogeneous log, such as health data, and we develop a solution.

Chapter 5 presents the final methodological step to automatically create simulation models of clinical pathways. We introduce an automatic procedure to convert a process model, discovered with process mining, into a simulation model. We propose a new subclass of state charts, called “Clinical Pathway State Chart”, with the required properties to simulate a cohort of patients while taking into account the pathways discovered using process mining techniques presented in Chapter 3 and the features found using the health analytics toolbox presented in Chapter 4. The clinical pathway simulation model is used to perform sensitivity analyses and what-if scenario evaluations.

Chapter 6 presents a comprehensive case study to illustrate the practical use of the approaches introduced in the previous chapters. The French national database of the hospital records from 2006 to 2015 is used as an event log. It contains the hospital records of several millions patients. The case study focuses on the clinical pathway of patients with cardiovascular diseases. This illustrates the benefit of the method for medical decision aid.

Chapter 1

Literature review

Contents

Abstract

In this chapter, we present a broad literature review on the topic of health-care data analysis. After a discussion on existing data-driven approaches, such as statistical analyses and data mining techniques, we describe existing works on the topic of modeling and simulation in health-care. Then, we specifically focus on the case of clinical pathway modeling. We compare existing definitions and scopes of clinical pathways, and we present existing modeling techniques. Among them, process mining stands out as a dedicated field for process discovery and analysis. We discuss the limitations of existing works and the remaining challenges to address.

1.1 Introduction

Nowadays, the scientific research applied to the health-care sector is an ever-growing field. Researchers from a variety of domains (Operational Research, Industrial Engineering, Business Process Management, Data Analytics, Artificial Intelligence, Computer Science, etc.) have found a tremendous interest in applying their approaches to improve health-care systems. Confronted with a difficult socio-economic context, many hospitals around the world must comply to new regulations and new management rules to balance their financial situation. In each scientific discipline, new applied case studies arise in the field of health-care. The inherent nature of health-care is to be more diverse and heterogeneous than other sectors where processes are carefully mastered and controlled. This forced researchers to develop new methods which are more flexible to incorporate human-related behaviors. Health-care is human centered at every level: the care process is dedicated to a patient who is diagnosed by a doctor, taken care by a nurse, supplied by a pharmacist and operated by a surgeon. Each patient is unique (compared to manufactured goods), which makes generic models more difficult to build. For those reasons, health-care has become a dedicated field of research.

The application of theoretical models on real-life cases requires to know how the actual system works. Models need data to be applied and tested. Yet, in their work, researchers often suffer from the lack of real data to fuel their models. Samples are often too small or the data quality is too poor. It is a large consensus of the research community that the access to reliable data has been the number one challenge for the practical application of their models, no matter how efficient they may be. This is changing. While the interest of health-care practitioners for scientific methods, capable of lastingly improving health-care systems, increased, digital technologies have grown. The perpetual modernization of facilities, in particular through computerization and implementation of information systems, generates large amounts of data on all care activities. Health-care systems, and hospitals in front line, have invested significant resources (human and material) to be able to collect, store and re-use data related to their activity. This now makes it possible to collect amounts of data that exceed the analytic capacity of the care providers. They no longer have the skills nor the means to take full advantage of this mass of information. This makes the contribution of researchers all the more important, and this has given rise to a paradigm shift: data are no longer used solely as a tool for validating pre-existing models, they become themselves the creative source of new models. Such methods are called *data-driven approaches*. The idea is to investigate existing data sources to create new added value. In the following, we present existing works related to such data-driven approaches in health-care and the ongoing research on modeling and simulation applied to health-care systems. Then, we specifically focus on the problem of clinical pathway modeling, including the contribution of process mining.

1.2 Data-driven approaches in health-care

A data-driven approach is a general work methodology where the starting point is available data. Then, a set of methods and techniques is implemented to use this data in order to answer a problem. These data-driven approaches all assume the same hypothesis: data indeed contain the answer (or elements of the answer) to this problem. Then, the challenge is to develop the means to find it. In health-care, data-driven approaches cover a broad spectrum of possibilities, depending on the type of available data and the question to solve.

The recent literature review of (Vuokko et al., 2017) presents the impact of structuring electronic health record for secondary use of patient data. The primary use of data is to provide physicians and nurses

with real-time information about the patient who is being taken care of. Secondary use of patient data is dedicated to an a posteriori analysis of the data for various purposes (statistics, decision support, resource management and reimbursement). Through the review of 85 articles, (Vuokko et al., 2017) presents the challenges of recording data in a structured manner and how it drastically improves the quality of secondary use studies. This work illustrates the ongoing interest of health practitioners and researchers to think of the final utilization when building health databases.

1.2.1 Levels and types of health data

Health data can be categorized according to the level of description they provide. (Herland et al., 2014) proposes 4 levels of detail: molecular level, tissue level, patient level and population level. Depending on the level, the nature of data, the analytic techniques and the pursued objectives are not the same. Genomics, proteomics and bioinformatics are the field dedicated to the analysis of genes, molecules and DNA (Table 1.1). In this thesis, we consider patient level data.

Table 1.1: The 4 levels of health data (Herland et al., 2014)

Sections	Data level(s) used	Subsections	Question level(s) answered
Using Micro Level Data – Molecules	Molecular	Using Gene Expression Data to Make Clinical Predictions	Clinical
Using Tissue Level Data	Tissue	Creating a Connectivity Map of the Brain Using Brain Images	Human-Scale Biology
	Patient	Using MRI Data for Clinical Prediction	Clinical
Using Patient Level Data	Patient	Prediction of ICU Readmission and Mortality Rate	Clinical
		Real-Time Predictions Using Data Streams	Clinical
Using Population Level Data – Social Media	Population	Using Message Board Data to Help Patients Obtain Medical Information	Clinical
		Tracking Epidemics Using Search Query Data	Epidemic-Scale
		Tracking Epidemics Using Twitter Post Data	Epidemic-Scale

At the patient level data, we identified 4 types of health data that are commonly used for data-driven analyses.

- Data directly related to a patient (diagnoses, administrative information, characteristics)
- Data related the care activity (medication, surgeries, medical imaging, biology tests, etc.)
- Data related to a care event (date, duration, severity, cost, outcome, etc.)
- Data related to the organization (appointments, human and material resources, number of beds, work schedule, etc.)

These data come from various sources and can take different forms: **structured database** (electronic patient files, claim systems) or unstructured data. Examples of unstructured health data are **texts** (medical report), **images** (radiography) and **signals** (electrocardiogram, times series). Table 1.2 categorizes existing literature of data-driven approaches depending on the nature of the data at stake. Here, we especially focus on structured data, which is the most widespread source of data. Examples of works on text data, images and signals are given but are not exhaustive.

Table 1.2: Overview of the different natures of health-care data

Nature of data	References in health-care
Structured data	(Lee et al., 2011; Hess et al., 2012; Vuokko et al., 2017; Lin et al., 2001; Cote and Stein, 2007; Adeyemi et al., 2013; Huang et al., 2012; Zolfaghar et al., 2013; Arslan et al., 2016; Huang et al., 2012; Shahin et al., 2014; Adeyemi et al., 2009)
Text	(Raja et al., 2008; Corley et al., 2010; Culotta, 2010)
Images	(Celebi et al., 2005; Xie et al., 2006; Rajendran and Madheswaran, 2010)
Signals	(Padoy et al., 2008; Bouarfa et al., 2011; Kusiak et al., 2005)

The most predominant sources of health data are **Electronic Medical Records** (EMR). They represent a broad type of data that are collected and stored during health-care activities. For a given patient, the EMR is an unique source of information about his/her medical history (diagnoses, imaging, medication, etc.). It provides nurses and doctors with all the relevant pieces of information required to take care of the patient. EMR essentially contains patient data and care activity data. Such data can be used for a variety of purposes. It can be used for the development of diagnosis aid tools. (Hess et al., 2012) developed a data mining technique to take advantage of the electronic medical records of 91 patients to help in the diagnosis of ovarian cancer. Historical data enable the creation of a decision tree which emphasizes the optimal strategies for an accurate diagnosis. In (Zolfaghar et al., 2013), the hospital encounters of 6,739 patients during 1 year are used to assess the re-hospitalization risk within a month. Reducing re-hospitalizations is a relevant strategy to reduce cost and improve the quality of care. A k-mean clustering method was developed to determine the patient features that most impact such a risk.

Data related to the organization of health-care systems are of major interest for Operation Research approaches (Rais and Viana, 2011). It can be used for optimal patient scheduling, logistic problem, forecasting, decision aid support, resource allocation and capacity planning. In that context, data are used to validate, test and improve the quality of the proposed models.

Data collected from sensors can also be of interest (Padoy et al., 2008). In (Bouarfa et al., 2011), a framework is introduced to recognize surgical tasks from data collected on surgical tool sensors. A hidden Markov model is used to represent the different steps of the surgery and the possible transitions. Preliminary results on a data set of ten surgical procedures show that it is possible to recognize surgical tasks with high detection accuracy. Ultimately, such a model could be used to detect deviations from guidelines or to determine the optimal location of resources in the operating room depending on the surgery duration.

1.2.2 Data mining in health-care

The most dedicated field for data-driven analysis is **Data Mining**. Data mining is a general analytic approach whose objective is to discover patterns in large data sets. It uses methods from the fields of artificial

intelligence, computer science and statistics. The objective is to discover new information from data, whereas in traditional statistics data are used to validate pre-conceived hypotheses about possible relations. In health-care, clinical research heavily relies on clinical trials, which are experiments in a controlled and limited environment. It often results in small samples of patients (few hundreds), but with very detailed clinical pictures. The objective is then to find statistical differences among different groups of patients. The idea of using data mining is different. The focus is on large-scale databases, which is at least a single hospital, and sometimes an entire country health system. Data mining can be used in two manners: supervised learning (historical data show the value of the variable to explain) and unsupervised learning (no examples of the target variable exist). The first category refers to predictive models (classification, regression) whereas the second refers to descriptive models (clustering, association).

Literature reviews

Several literature reviews on the topic of data mining applied to health-care can be found (Yoo et al., 2012; Niaksu et al., 2014; Herland et al., 2014; Das et al., 2015). They provide a complete landscape of this topic's different facets. (Das et al., 2015) specifically focuses its review on real-life applications related to the Indian health system. The lecture of (Yoo et al., 2012) is highly recommended to get an extensive understanding of the contributions, the challenges and the techniques of data mining in health-care. The following topics are discussed.

- General definition of data mining and differences with statistics
- How data mining in health-care differs from other domains
- Theoretical aspects of existing data mining algorithms (for classification, clustering and association)
- Practical guidelines for real-life application of data mining algorithms in health-care
- Description and categorization of existing works by types of application

(Jothi et al., 2015) reports the literature review of 50 articles related to the "application of data mining techniques in health-care". They highlight the massive interest of existing works for classification tasks. The most representative health-care application of classification is to determine the diagnosis of a patient based on his/her symptoms. In that context, data mining models are decision aid tools.

Perspective of data mining in health-care

Several conclusions can be drawn from these works: first, data mining in health-care is not new. It has been widely applied since the emergence of health databases and sufficient computational power. Data mining is capable of answering key questions through new descriptive and predictive models. Examples of successfully applied data mining are numerous, such as the prediction of undesirable medical events (prediction of patient falls with artificial neural network (Lee et al., 2011), prediction of ischemic strokes with 3 data mining algorithms (Arslan et al., 2016)) or the classification of patients (grouping patients with random forest in one of 4 risk levels of starting a disease (Shahin et al., 2014)). Specific applications also intend to find correlations in the events of medical history. In Huang et al. (2012), the medical records of 9,862 patients were extracted from a hospital information system (HIS) to find correlations between the presence of certain diseases and hypertension. All the input variables, as the target to predict, were defined as binaries (presence/absence). Naive Bayesian and J48 classifiers were implemented to create the predictive model. The results show the difficulties of predicting unbalanced classes (rare events, there are much fewer patients with hypertension compared to patients without).

Still, several challenges of using data mining in health-care remain: the configuration of data mining algorithms requires dedicated skills and cannot be performed by end users such as doctors. This is especially true as most data mining algorithms are parameter sensitive. Second, the predictive model accuracy might not be high enough to be used in a clinical environment. It is sometimes due to the quality of the collected data (missing values), or to the originally intended purposes of the data collection (financial purposes, but not clinical studies). In addition, all the predictive factors of a disease might not even be known. A last challenge of health data is the presence of extremely imbalanced classes (the studied group is much smaller than the reference group) (Khalilia et al., 2011). For instance, there are much fewer patients who develop a nosocomial infection compared to the population of hospitalized patients. Standard algorithms struggle to build outstanding models, especially in comparison to a dummy "most-frequent" classifier which gets high accuracy. Imbalanced classes shows the need for several performance criterion and for suitable mining algorithms.

It makes data mining applied to health-care an interesting ongoing research area where the current focus is much more application oriented (getting appropriate data, automated analysis, auto-tuning algorithm) than purely theoretical (from the algorithmic point of view). The search for performance improvement of predictive models is not in new algorithms that would bring a 0.01% improvement in accuracy. Instead, it is the combination of methods from different fields (Operation Research, Data Mining, Computer Science, Bio-Informatics) which appears the most promising (Corne et al., 2012; Gomes et al., 2012; Carrizosa and Romero Morales, 2013).

1.3 Modeling and simulation in health-care

Discrete Event Simulation (DES) has been widely used in the literature for modeling health-care systems for various purposes: performance evaluation, optimization, demand forecast, etc. Several literature reviews (Jun et al., 1999; Fone et al., 2003; Augusto and Xie, 2006; Günal and Pidd, 2010; Khudyakov et al., 2014) are strongly indicated to get a complete landscape of the scientific contributions using such methods. Taking a look at the DES related literature from the past twenty years, most of the modeling effort has been at a micro level and related to specific aspects of the hospital, such as emergency departments, operating theaters, outpatient departments, inpatient wards, and intensive care units. Indeed, most simulation studies may be classified depending on the case study, which is often a hospital service.

1.3.1 Modeling of hospital services

Emergency department seems the most popular area for simulation modeling in health-care: such system contains highly stochastic yet easily observable processes (Centeno et al., 2001; Miller et al., 2003; Glaa et al., 2006). Although a lot of articles describe specific yet complex models to achieve realistic results (Takakuwa and Shiozaki, 2004; Duguay and Chetouane, 2007), some studies propose generic models supposed to be transferred to other hospitals (Sinreich and Marmor, 2004). Simulation has also been used to propose control strategies for emergency services taking into account costs (Prodel et al., 2014) or patient satisfaction (Pehlivan et al., 2013a). Inpatient facilities have also been extensively studied using DES; most observed objectives within this topic are staffed beds capacity sizing (Wiinamaki and Dronzek, 2003; El-Darzi et al., 1998), length of stay minimization (Vasilakis and Marshall, 2005) and patient flow modeling (Augusto and Xie, 2009b). Outpatient facilities and services are a subject of growing interest since hospital managers push for the increase of ambulatory surgery, which is cost effective. Outpatient clinics case stud-

ies are frequent (Wijewickrama and Takakuwa, 2005; Takakuwa and Katagiri, 2007), providing guidelines for patient flow management. Ambulatory surgery is also a hot topic, where DES provides optimal sizing of ambulatory services connected to the operating theater (Ramis et al., 2001; Ferrin et al., 2004).

Other hospital services such as operating theater (M'Hallah and Al-Roomi, 2014), pharmacy department (Augusto and Xie, 2009a), and geriatric services organization (Franck et al., 2015) have also been studied using simulation. Finally DES has also been used for multi-services clinics and whole hospital monitoring and performance evaluation (Moreno et al., 1998). Modeling and simulation frameworks have also been proposed (Augusto and Xie, 2014). DES has also been used for performance evaluation related to the usage of health-care information systems in cancer patient pathway (Augusto et al., 2015), including a micro costing model for cost analysis. High scale formal models are also proposed to represent the global pathway of patients including health-care structures outside of the hospital (Hamana et al., 2015). To the best of our knowledge, DES has been extensively used for hospital services and patient pathway on the short term, but was never used for patient pathway global modeling. Isolated initiatives propose mixed agent-based-discrete-event simulation models for specific diseases such as Chronic Obstructive Pulmonary Disease (Charfeddine and Montreuil, 2010). Recent initiatives have also been proposed for the perinatal application (Pehlivan, 2014).

1.3.2 Real-time simulation

The issue of simulation models being single use because they cannot constantly adapt to the ever changing nature of the actual system was addressed in a way by the definition of new paradigm, "Dynamic Data-driven Application Simulation". The idea is to enable the incorporation of new data into an existing simulation model continuously, and thus to allow the model to dynamically steer the measurement process. It offers the promise of improving modeling methods, and improving the analysis capabilities of simulation (Darema, 2004; Douglas and Efendiev, 2006). This approach was applied in diverse forms and in various areas (Douglas and Efendiev, 2006), in the very same way that simulation is applicable to a very wide range of domains. The challenge of real-time data feeding of a simulation model more often relies on technical challenges than on conceptual ones. Difficulties may arise when one needs to process structured and unstructured data from several sources in a relatively small amount of time. The real-time model must accommodate both the simulation objectives and the timing constraints. Moreover, this approach assumes that an initial simulation model was already built and is ready to receive new data. Here, we are focusing on the automatic construction of such a model, so that the construction process can be applied again when needed (for instance if new data are available).

1.4 Clinical pathway modeling

1.4.1 Definition of a clinical pathway

In this thesis, we are interested in developing a data-driven approach to model and simulate health-care processes, a.k.a **clinical pathways** (CP). A process is a collection of related activities that serve a common goal. A clinical pathway is a care process made of tasks whose ultimate objective is to make the patient healthy. It describes a set of treatment or administrative activities, such as consultation, appointments, imaging examination or surgery, with the common goal of treating a patient (Rebuge and Ferreira, 2012). In the literature, clinical pathway is also referred as "care pathway", "critical pathways", "integrated care pathways", "care maps" or "patient trajectory". There is no absolute definition of a clinical pathway. CP

can refer at the same time to a very detailed view of a care process (e.g. the minute-by-minute sequence of surgical acts in an operating room) or to the macroscopic description of a patient's medical events (hospitalization in January, general practitioner consultation in March and surgery in July). The time window of CP can be short (e.g. from the entrance in the emergency room to the discharge 4 hours later) or very long term (e.g. from a heart failure diagnosis until the death 10 years later).

1.4.2 Clinical pathway modeling approaches

Existing literature on the study of clinical pathways is vast, due to a large diversity of **modeling approaches**, of **analysis purposes** and of **CP description levels**. The CP modeling methodology of many articles is based on experts' opinion. Doctors are interviewed by a modeler to know how the care process is (supposedly) happening. Then, a model is built based on the collected information. Here, we do not extend further on that part of CP literature because they do not consider any data source and are subjective.

Most recent studies focus on the use of existing data to model and discover clinical pathways (Lin et al., 2001; Huang et al., 2013; Bouarfa and Dankelman, 2012; Cote and Stein, 2007; Adeyemi et al., 2013). We gathered these works in 4 groups, depending on the field of the modeling technique: *statistical techniques*, *data mining methods*, *business process modeling* and *process mining algorithms*. These approaches receive an increasing attention in the field of Medical Informatics.

Statistical and mathematical techniques

Statistical techniques are dedicated to the use of mathematical methods to find significant relations among two or more variables. Statistical tests (and p-value) can be used to assess the relation between patient characteristics and their medical history. In (Adeyemi et al., 2009) and (Adeyemi et al., 2013) a **logistic model** is implemented to determine the risk of being readmitted at hospital within 36 days in chronic obstructive pulmonary diseases. The model shows a significant difference of risk for patients of different region and gender. This conclusion was achieved thanks to a database of all the hospital events over a large territory. Such method can be implemented as a decision aid tool for clinicians. Statistical method can also be used to discover CP from data. The work of(Bouarfa and Dankelman, 2012) derives a work-flow consensus from multiple clinical activity logs to automatically detect work-flow outliers (without prior knowledge from experts). Work-flow mining was used to derive a consensus work-flow (i.e. the average surgery) from 26 surgical logs using **multiple sequences alignment**. The large computational requirements make the method non-scalable for large data.

In (Huang et al., 2013), a **probabilistic** (Latent Dirichlet Allocation) is used to automatically discover treatment patterns in unstable angina (2,934 patients) and several cancers. For each of the discovered patterns, each patient is assigned a probability of following this pattern. Results show how the different treatments are distributed during a stay, depending on the ongoing length of stay (e.g. medical imaging is made within the first 48 hours and hemoglobin test after 7 days). This approach proposes an innovative way to describe CP, through statistical distributions, compared to control-flow models.

Markov chains are a mathematical formalism from probability theory which is used to model the possible states of a system and the transitions among these states. A CP can be seen as a Markov chain where the studied system is a patient and the states are the care steps (Cote and Stein, 2007; Marwick et al., 2013; Lin et al., 2005; Elghazel et al., 2007). In (Cote and Stein, 2007), the authors propose a stochastic Markov chain to model the care process of doctor consultation. Their model includes 5 states (wait, nurse care, examination, imaging, check-out). Historical data were used to derive the transition

probabilities among the states. The scope of the model is limited to the doctor consultation, it does not consider the entire pathway of a patient. Similarly, in (Lin et al., 2005), each time that a patient visits a doctor or a hospital, he/she is considered as a new patient. The model does not provide the big picture of the care process. Markov models can model different aspects of CP. In (Yen and Chen, 2013), the states of the Markov chain represent the development of chronic diseases for a patient (e.g. hypertension, diabetes, obesity). The objective is to propose a stochastic model capable of modeling transition rates and temporal sequences of a patient's condition for any number of co-morbidities. 3-state and 4-state Markov models are analytically solved and a generalization to a n-state model is proposed. The model is proven to better estimate the proportion of susceptibility to co-morbidity based on the current situation of a patient. In (Marwick et al., 2013), a 6-state Markov chain is used to model the pathway of patients with mitral regurgitation. States represent patients' condition (asymptomatic, heart failure, replacement, repair, stroke, dead). The model is used to identify eligible patients for an early surgery in order to prevent the occurrence of a stroke. An advantage of Markov chains is to allow for nested models that can describe a process at different levels. In (Zhang et al., 2015), 4 levels of aggregation are used to model a clinical pathway. A real-life application of the CP in chronic kidney disease is described (data of 1,576 patients). In the most detailed view, each state represents a care activity such as doctor consultation, medication (e.g. diuretics) and medical diagnoses (CKD stage 3, hypertension), whereas top-level clusters contain tens of such activities. The nested construction allows for a simplification of the clinical pathway, so that major trends stand out. In all cases, the main weakness of Markov models is the limited number of states that they can handle.

Data mining for clinical pathways

In the context of clinical pathways, data mining approaches are mostly used for two purposes: (1) discovering patterns in the sequence of medical events and (2) predicting the outcome of the next steps in the pathway.

The idea behind **pattern discovery** is that there exist some general schemes in the order and in the time line of patient pathway. This is due to the standardization of care processes. First, a patient is diagnosed, then he is treated and finally he is cured. However, each patient is unique and has a personalized pathway in response to his condition. Pattern discovery aims at unraveling patient pathway from such noisy and large medical data. Different aspects of clinical pathway can be specifically discovered, such as the time dependencies (Lin et al., 2001; Huang et al., 2013; Dagliati et al., 2014), the routing choices (Rozinat and van der Aalst, 2006b; de Leoni et al., 2016) and the control-flow (the most frequent paths and the deviations) (Bouarfa and Dankelman, 2012; Iwata et al., 2013).

In (Lin et al., 2001), the authors reports a mining strategy to discover time-dependency patterns in the clinical pathways of managing brain stroke. The medical record of 113 patients are used . The dependency between two care activities exists if they directly follow one another. Hence, they derive an oriented graph of the possible pathways from the data. A predictive model, based on association rules, was also implemented to determine the most probable path that a new patient may follow. The limitations of their work is to be extremely sensitive to noise (a noisy activity may interfere between two regular activities), to work only for small data volumes and to only consider direct following dependencies (whereas long-term dependency may exist). The approach of (Huang et al., 2013) also focuses on the identification of care events that occur within a certain time window. They develop a method which segments each patient pathway into relevant continuous and overlapping time intervals. The optimal time intervals are found when frequent medical behavior patterns are discovered. The objective is to gather close events in clusters

to create a compact CP model from large and heterogeneous data sets. They successfully apply the method on 4 medical case studies. Records include 48 patients and 3,405 medical events (225 distinct) for the smallest sample, 445 patients and 23,106 events (513 distinct) for the largest. This approach is particularly adapted to the case of patients who repeatedly receive several medical cares in a short period of time (e.g. a day), but not for completely sequential ordered events.

The **prediction of the next step** in a clinical pathway can be defined as a classic data mining problem. Given a set of input characteristics, the model tries to predict which patients are more at risk of having a given medical event. The challenge of predicting the re-admissions is a dedicated example of such approaches (Adeyemi et al., 2009; Zolfaghar et al., 2013). The same prediction problem can be formulated at any step of a clinical pathway and is called a *decision mining problem* (Rozinat and van der Aalst, 2006b). We address this problem in detail in Chapter 5.

Business Process Modeling

Business Process Modeling (BPM) is the activity of representing processes of an enterprise (in a broad sense). The idea is to be able to represent a system, so that it can be analyzed and improved. The modeling step often relies on a domain-specific ontology. An ontology is a formal naming and definition of the types, properties and relationships of the entities that exist in a particular domain. The work of (Yao and Kumar, 2013) proposes a flexible modeling framework, compared to traditional business process models, because CP frequently involves deviations and atypical behaviors. In (Braun et al., 2015), a BPMN model is used to represent CPs. In addition, several extensions are incorporated to the model for multi-perspective modeling (medical resources, documentation, financing, etc.), in addition to the control-flow. The work of (Shitkova et al., 2015) proposes to unify existing modeling formalism (ULM, EPC, BMPN, etc.) in a common CP framework. The proposed methodology takes into account the characteristics and the possible usage scenarios of CP through semantic annotations, thus facilitating the choice of an appropriate modeling technique.

Process Mining algorithms

Process mining is a field entirely dedicated to the analysis of processes (in all domains) through the utilization of event logs (i.e the history of occurred events). Process mining is used to provide an impartial view of a process based on facts that really happened, as seen in the data. Thus, process mining combines the advantage of being data-driven and to focus on processes, such as clinical pathways. The following section presents existing works on process mining applied to health-care.

1.5 Process Mining in health-care

The general goal of process mining is to extract new information about processes from an event log (van der Aalst, 2011). Process mining is an ever-growing field which is a bit more than a decade old. There is a rich literature on the subject. It can be applied in various systems and domains (industry, administration, finance, health-care, etc.). Process mining is interested in discovering process models from raw data. It finds hidden patterns related to the order of the activities in a sequence. Process mining can be used to automatically discover a process model from data, to validate an existing model or to enrich a work-flow model with other perspectives (van der Aalst, 2011). Figure 1.1 shows an example of a process model discovered from a health-care event log. The model formalism is a Petri Net.

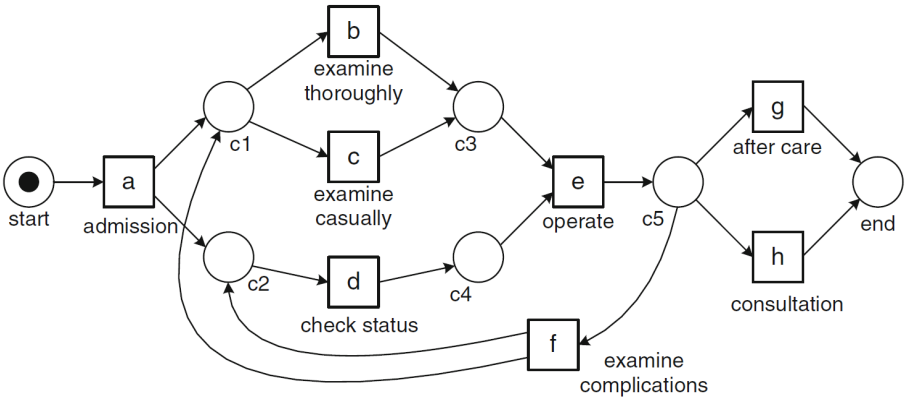


Figure 1.1: Illustration of a process model discovered from a health-care event log (Mans et al., 2015)

1.5.1 From the emergence to a widespread topic

Emergence. Regarding health-care, it appeared very soon as a potential field of application for process mining techniques, especially for process discovery; (Lang et al., 2008) in a medical imaging service (Mans et al., 2008) in stroke care, (Mans et al., 2009) in gynecological oncology, (Gunther et al., 2008) in the operation of X-ray machines, (Blum et al., 2008) in surgical work-flow and (Mans et al., 2012) in dentistry. Conformance checking of existing models was also addressed (Zhou, 2009; Kirchner et al., 2013). However, several studies pointed out the difficulties of existing mining algorithms to perform well on heath data (Lang et al., 2008; Gunther, 2009; Mans et al., 2015; Rebuge and Ferreira, 2012). Models tend to be over complex ("spaghetti-like" because of too many nodes and arcs) because the methods are not flexible enough for variable processes. The variability is due to the uniqueness of each patient and to the large number of possible medical actions to take care of a patient. In that context, creating understandable models that can be used for improvement of anomaly detection is a challenge. Further works proposed new strategy to perform process mining on health data. In (Rebuge and Ferreira, 2012), they propose to cluster patients in homogeneous groups before trying to discover a process model, each patient being seen as an ordered sequence of medical events. The clustering method is based on a first-order Markov chain and was proven to be robust to noise in the data.

Several works intend to identify the commonly faced difficulties related to health data for the application of process mining (Mans et al., 2013, 2015). The main differences compared to other domains are the high variability and the presence of unstructured processes. Data quality is also a matter of issue. In (Mans et al., 2015), they point out that existing works (35 publications) only apply basic process mining approaches on limited hospital data, hence not exploiting the full potential of process mining. The authors gather scattered works to identify and group the specific features of health-care processes. In (Mans et al., 2015), they present how the four types of process mining questions apply to health-care ("what happened", "why did it happen", "what will happen" and "what is the best that can happen"), illustrated by 6 case studies which answer domain-specific questions. Finally, they propose a "health-care reference model" which outlines all the different classes (e.g. patients, diagnoses, staff and appointments classes) of data that are potentially available for process mining in a HIS. It provides the big picture of the medical concepts that can be mined and linked together through the data. This general mapping work is essential to fully understand the nature of the data and the underlying concept they represent. A good knowledge of what the data represent is mandatory to mine relevant models. The limitation of (Mans et al., 2015) is to only consider hospital data, thus ignoring community medicine.

Figure 1.2 shows the general concept of process mining application in health-care. HIS is seen as the standard source of data related to hospital activities (nursing, surgeries, diagnoses, registering, etc.). Event logs can be derived from the database by selecting at least 3 fields (a date, an activity name and a patient ID). Process mining approaches can then be applied to analyze this event log and to produce a process model.

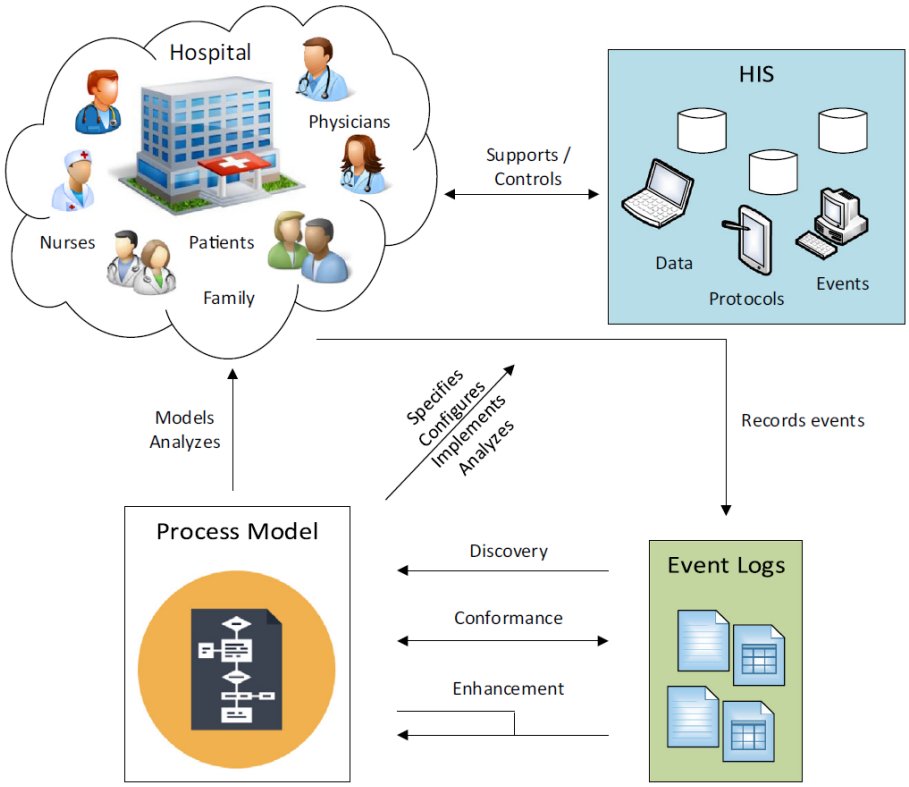


Figure 1.2: Overview of process mining in health-care (Rojas et al., 2016)

Literature reviews. Process mining applied to health-care continued to be addressed in the following years (Tsumoto et al., 2014; Dagliati et al., 2014), focusing on improving the existing discovery algorithms (Verbeek and van der Aalst, 2013), or on comparing process models (Montani et al., 2014). All these works have resulted in the enrichment of a substantial corpus of articles on the subject. Very recently, literature reviews on the exact topic of "process mining in health-care" arose (Yang and Su, 2014)(37 studies from 2004 to 2013),(Kurniati et al., 2016)(37 papers from 2008 to 2016) and (Rojas et al., 2016)(74 papers from 2007 to 2016). The literature review of (Kurniati et al., 2016) specifically focuses on process mining in oncology. The most frequently studied application is gynecological cancer. Conclusions of this review are encompassed in (Yang and Su, 2014) and (Rojas et al., 2016).

In (Yang and Su, 2014), the authors address 3 research aspects of clinical pathway modeling: (i) process discovery for clinical pathways design (19 papers), (ii) variants analysis and control (13 papers) and (iii) continuous evaluation and improvement (5 papers). This classification is different from (Rojas et al., 2016) where papers were categorized according to the following 11 criteria:

- **Process type** (Medical treatment processes, organizational processes, elective care)
- **Data type** (vital signs, personal data, drugs, administrative data, data from medical devices)

- **Frequently posed question** (understand what happened, find patient profiles at risk of deviating)
- **Process mining perspective** (control flow, conformance checking and organizational perspective)
- **Tool** (ProM, DISCO, RapidMiner)
- **Algorithm** (Heuristic miner, fuzzy miner, trace clustering)
- **Methodology** (single step, extensive, partial, clearly stated)
- **Implementation strategy** (direct implementation, semi-automated, integrate suit)
- **Analysis strategy** (single shot analysis with existing tool, personalized approach)
- **Medical field** (Oncology, surgery, cardiology, radiotherapy)

The review of (Rojas et al., 2016) shows that the most frequently used process mining techniques are process discovery techniques, namely the heuristic miner (26%) (Weijters et al., 2006) and the fuzzy miner (20%) (Gunther and van der Aalst, 2007). These two specific algorithms are used for process discovery but do not provide any guarantee that the discovered model is optimal regarding any quality measure. It illustrates the need for new and more flexible methods dedicated to health mining.

The conclusions that can be derived from existing works are the following. Process mining has a great potential in health-care process management. It provides objective ways to study clinical pathways through a meaningful usage of stored data. Data provide real knowledge about the execution of care processes and facilitate the identification of improvement opportunities. Also, medical processes appear more complex than business processes of other domains. Many efforts have been made to accommodate existing techniques to better answer the specific challenges of health data. Still, several challenges remain for a broader and practical use of process mining in health-care and shall be addressed in future researches.

1.5.2 Limitations and perspective in process mining applied to health-care

Based on existing works and literature reviews, the following limitations can be listed about process mining in health-care:

1. Data access and data quality remain obstacles for an efficient use of process mining in health-care.
2. Only data from Hospital Information Systems are considered, which is a partial view of the health system.
3. Using health data requires to understand them, which is not straightforward due to the complexity of medicine (e.g. even doctors may disagree on a diagnosis, interpreting data can be just as difficult).
4. Data sources are heterogeneous and hard to use jointly (patient file, imaging, vital signs, medical history)
5. Health-care processes (clinical pathways) are inherently variable and unstructured, due to the diversity of patients and situations. It makes event logs noisy, most traditional algorithms inefficient to answer relevant questions.
6. Most of the process mining methods only pay attention to the starting time and the name of an action (e.g. medication), but not to the action results (e.g. Is the patient cured?).
7. Existing works were developed and tested for specific medical centers or case studies. No generic model which could be automatically reused has been proposed.

8. There is a lack of suitable visualization strategies for less-structured processes.
9. No benchmark study of existing tools on different health data has been proposed.
10. Existing studies were mostly applied to small to middle size events logs (few hundreds of patients).

In this thesis, we intend to address several of these challenges, that-is-to-say using large volume of data from many hospitals, the capacity to deal with heterogeneous data, a generic process discovery methodology and a suitable visualization tool. In Chapter 3, we propose a new process mining approach which automatically builds patient pathways for a given pathology. The scientific challenge lies in the complexity of the health-care data that we use as an input for our model. To address this problem, we propose two approaches, one based on integer-linear programming and exact optimization and the other on a new formulation of the optimal discovery problem. They take into account health-care related parameters such as patient pathology and diagnosis. A real case study related to cardiac defibrillators is also addressed in Chapter 6.

1.6 Summary

In the previous sections, we have seen the current situation of existing works on the topics of process discovery, analysis and simulation of hospital health data. A focus was made on the modeling of clinical pathways. The advantages and the challenges of promising process mining approaches were discussed.

Clinical pathway modeling is a popular topic which has been addressed in various ways, depending on the techniques and the description level. Nowadays, hospital information systems can store huge amounts of data about the care processes as they happen. These data can be used to study the unfolding of such processes. This way, we can analyze the operational processes within a hospital based on facts. In the case of a region with multiple hospitals, each gathering its own data, we can also study the entire pathway of a patient on the long term. The pathway model is no longer confined within the walls of a single hospital, it describes the complete sequence of care events that happen, even if the patient is taken care of in different places.

Data mining, statistics, operation research and process mining are the most represented research areas in the literature. Each field provides a set of techniques that are capable of mining data to answer new questions. The same observation was made about health data: new techniques need to be specifically developed to fit the inherent variability of health-care processes, and thus of health data. Otherwise, even the most sophisticated models only bring already known information by experts of the domain. The added value of researchers for practitioners can only be brought by flexible models. In addition, thanks to new technologies and computerization of medical files, the current trend is to store massive amount of data. Analytic methods and models must adapt to deal with such data. Our literature review shows that existing real-life applications do not exceed few tens of thousands patients. This is mainly due to access restrictions to health data because they are sensitive, and also because of quality issues.

Among other techniques, process mining stands out as a promising way to automatically derive and analyze clinical pathways from raw event logs. The research area of process mining in health-care is a growing field which has been prolific in the last 5 years. Still, many scientific challenges remain for a broader diffusion use of such techniques on health data. Among these challenges, the most striking is the difficulty of existing algorithms to handle less-unstructured processes. Several techniques have been developed on a case-by-case basis to overcome this issue. The resulting models are specific to a hospital

information system or to a given disease model. It leads to the current situation where no generic methodology can be re-applied. In this thesis, we propose a complete methodology to automatically transform heterogeneous event logs into actionable simulation models. For that, we combine approaches from operation research, data mining and process mining. Our approach is proven suitable to deal with large volumes of data (any number of patients).

Our modeling methodology is applied on a several millions-hospital event database, the French national database of all hospital activities (11 million patients per year). To the best of our knowledge, few studies describe the use of process mining to discover clinical pathways at a national level, by considering hospital stays as activities. Most works only consider what happens during a single stay. One study can be mentioned to that regard (Jensen et al., 2014). The work was done on 15 years and 6.2 million-patients database in Denmark (Jensen et al., 2014). A statistical experiment was designed to discover time dependencies among several key diagnoses. They finally suggested that trajectory analyses may be useful for predicting and preventing future diseases of individual patients. Our method is different as we define the problem with a process mining approach and we use optimization to find the best process model rather than assessing the statistical validity of discovered correlations.

The literature review presented in this Chapter was intentionally general on the subjects of process discovery, modeling and simulation of clinical pathways using health data. In the remaining of this thesis, each technical Chapter is provided with its own substantial literature review.

Chapter 2

General methodology and contributions of an automatic conversion of complex event logs into simulation models

Contents

Abstract

This chapter presents the methodological flow-chart of this thesis. Our general goal is to extract, represent and study a process (e.g. a care management process). Here, we explain how we automatically turn raw data from a database of events into a simulation model in a step-by-step approach. For each step, we describe the required inputs, the scientific challenges, our proposal to address these challenges and the generated outputs. Finally, the originality and the scientific contributions of the present thesis are introduced.

2.1 Introduction

During the last two decades, the amount of data collected in Information Systems has drastically increased. Data related to Business Process are being extensively recorded on various mediums (databases, text reports, image banks, transaction logs or audit reports). This large amount of available data has become highly valuable as it may reveal important patterns of the underlying processes. It can lead to a better understanding of processes and of their potential improvements. This reality is currently even more prominent in health-care where the computerization has been slower compared to other domains. Moreover, as health data are extremely sensitive, they are not easily released and shared to researchers or analysts.

Computer simulation is a tool allowing to reproduce the behavior of a system. It is based on an abstracted model of the actual studied system. Any type of system can be simulated, e.g. biological, human, natural, political, health-care or economic systems. Simulation is a helpful way to study the impact of key features on an entire system by taking into account complex interactions and without having to design a real-life experiment. It is a cost-saving way to test many alternative conditions or decisions of a situation. The main advantage of simulation is to enable this evaluation of "what-if" scenarios and to quantitatively predict their outcomes. The current chapter is dedicated to the description of a complete and innovative methodology to automatically convert raw data from an event log into a simulation model and scenarios evaluations.

Process mining is an innovative method that can be used to discover business process models from event logs. The main contribution of this research lies in the proposal of a comprehensive and automatic approach to generate a simulation model from raw data instead of using a handmade model. Handmade models are built based on available documentation, observations of the modeler and on interviews of experts. This is a time-consuming approach and a partial view of the processes. The perception of the actual process is influenced by the experience of the human studying it and it introduces a bias which may have a potential impact on key performance indicators and on overall results of the simulation study. Also, such approach is hardly reproducible as the model is built on a case-by-case basis. An automated conversion procedure to transform raw data into an actionable simulation model combines the advantages of using objective information contained in the data, and of being a fast and reproducible method on any new data set.

2.2 Literature review

The search for an automatic conversion, i.e. with the minimum handmade interventions, has been addressed in several ways by the research community. It is a consensus that building a simulation model by hand is time consuming, not impartial and size limited. A simulation model may also become out-of-date as soon as the actual system changes and is hardly updated by the initiators of the models. Hand-made models are not durable, which is a drawback for long-term analysis of a system.

The general problem of converting a preexisting model into a simulation model has been addressed in the literature (Augusto and Xie, 2014; Zhu and Kong, 2012; Ndiaye et al., 2016; Mueller et al., 2007; Popovics et al., 2012; Akhavian and Behzadan, 2013). In (Augusto and Xie, 2014), the authors defined a framework for the automatic conversion of Unified Modeling Language (UML) models into simulation models. They used a special class of Colored Petri Net (CPN), namely a Health-Care Petri Net. As a result, the proposed methodology leads to a fast-prototyping tool for easy and rigorous modeling and simulation of health care systems. A case study on the pharmacy delivery process is presented to show the benefits of the methodology. In (Zhu and Kong, 2012), the authors proposed an approach with an intermediate layer to

also automatically convert UML to CPN. They used it in the context of software performance evaluation. In both (Augusto and Xie, 2014) and (Zhu and Kong, 2012), the same limitation applies: the initial UML model is currently hand-made by a modeler. Only the conversion to CPN is automated. In (Ndiaye et al., 2016), the authors present a conversion procedure to transform an informal description of an Industrial Control Systems architecture into the formal framework of Colored Petri Nets. The ultimate goal is to assess the performance of this architecture and to allow industrial companies to use it, especially when they are familiar with Industrial Control Systems but not experts of CPN. However, the methodology is not suitable for large size architectures. In (Mueller et al., 2007), the authors proposed a novel approach to the simulation models generations. They introduced a simulation framework based on a specific type of Petri Net structure. This framework is an extension of classical Petri Nets for time and priorities in transitions firing. Thus, they used CPN in a slightly different way than usual to adapt to the analysis of semiconductors manufacturing. We can see from the literature that Petri Nets, and more specifically Colored Petri Nets, are commonly used for simulation models. The Petri Net formalism is extremely powerful and relevant for theoretical analysis and for the demonstration of remarkable properties. However, Colored Petri Nets expose very few analytical properties, and thus analyzing the behavior of a system using colored nets cannot be done, except by simulation (Proth and Xie, 1996).

In (Popovics et al., 2012), the authors present a methodology to reduce the time needed to build a discrete-event simulation model. They introduced a way to automatically convert data obtained from Programmable Logic Controller (PLC), a widely used and standardized digital computer in manufacturing processes. PLCs produce standard data structure with logical conditions that can be systematically transformed into input variables for a simulation model. In (Akhavian and Behzadan, 2013), the authors propose a data-driven approach for the automated generation of simulation models in the specific field of construction engineering and management. They converted the data recorded in several data sources into the matching concept used in discrete-event simulation model (e.g. entities, resources, times, localization).

The process mining approach is different from previously described conversion methods. Process mining techniques are dedicated to the automatic transformation of raw data, namely event logs, into conceptual process models. Such process models provide a static view of what happened as recorded in the data. The field of process mining goes far beyond process discovery when exploring data. The *model extension* phase intends to integrate other perspectives into the model, such as resource consumption, duration of activities and costs. It provides new insights about the process and it gives ideas for process improvement (van der Aalst, 2011). Moreover, such an extended model can be used to create a simulation model covering all perspectives. To the best of our knowledge, (Rozinat et al., 2009) is the closest paper to the methodology presented in this thesis. In (Rozinat et al., 2009), they showed how process discovery and model extension can be used to generate a simulation model in *CPN Tools*. CPN Tools is a powerful simulation environment based on Colored Petri Nets[1]. The process discovery was made using the Alpha-algorithm, the first invented process discovery algorithm, thus producing a Petri Net. Then, the authors explained how to use a classification model to study the decision points of the models, i.e. finding the correlations between the data and the routing choices of the process model. The classification problem to solve is the prediction of the next activity based on the traces' features (a trace is an instance of the process). They used a decision tree algorithm. The study lacks an evaluation of the classification models' quality and of their impact on the system. Indeed, the model obtained for each decision point should be tested on new data (not the same as for the learning phase) to avoid over-fitting and to ensure the quality of the learned rules. Moreover, Decision Trees are not guaranteed to be the best classification algorithm. A comparison of several algorithms

[1] see www.cpntools.org

could help getting better results. The learned routing rules will have a major impact on the simulation of new process instances. Hence, the search of the best possible classifiers is of major interest to ensure that the resulting simulation models closely follow reality. Finally, a CPN model is used to integrate these rules with other perspectives into a simulation model. The work of (Rozinat et al., 2009) is the first to propose a methodology to initially convert raw data into a process model, and then to convert this process model into a simulation model with several perspectives (resources, decision points, delays).

This work of (Rozinat et al., 2009) was focused on proposing a comprehensive methodology that can be applied to transform event logs into a simulation models. Our work intends to reach the next step in the automatic conversion of raw data into actionable models. It includes the definition of an enhanced methodology, with additional steps and improvement of the already existing, with the goal to deal with logs of any size and complexity. First, we formally define the optimal process discovery problem, which is a formulation that ensures that a discovered model is optimally representative of the data. We propose a new discovery algorithm to solve this optimal discovery problem. Our approach was built to be suitable for real logs, with any number of traces and of different activities (from tens to hundreds of thousands). This is an important contribution to allow the practical use of the method. After a model is discovered, we introduce several methodological steps to create classification models that fully take advantage of all the available data when learning decision points rules. Finally, we present a framework for the automatic conversion of a process model into a simulation model. The proposed simulation model is an extension of the classical notion of state chart (or finite state machine). Finally, the model is run using a Monte-Carlo simulation based on a state chart formalism and that reproduces the random behaviors of clinical pathways. It works by repeating random sampling to obtain strong numerical results. The contribution related to this specific phase is to provide an automatic sensitivity analysis of all the parameters also extracted from the raw data on the simulated system (patients' features, decision trees, size of the model, alignment scores). This work can be seen as the foundation to build even more complex models. More specifically, Discrete Event Simulation models and Multi-Agent models could be further improvement of the present work by capturing patient's behaviors and preferences, as well as resource sharing and sizing on a national scale.

2.3 The 8 proposed steps for an automatic study of processes

Our complete methodology to automatically convert raw data from an event log into a simulation model is made up of 8 distinct steps. Figure 2.1 shows how these 8 steps are interlinked. Most steps are sequential, meaning that the output result of a step is used as the input of the following step. 4 of the 8 steps require at least two distinct inputs, which creates a more complex work-flow than a mere ordered sequence of tasks. The following provides a general explanation of each step.

2.3.1 The starting point: data

The starting point of our methodology is a database. The available database shall contain the following information. First, it must contain records related to the execution of a process, i.e. of the ordered sequence of events that were realized together to achieve a common goal (e.g. the succession of 5 weekly chemotherapy sessions aims to cure a patient). Each event that is recorded is the database can have many attributes. For instance, if each event represents a hospital stay, the attributes can be its starting date, its duration, the medical diagnosis, the age of the patient, the amount of delivered drugs, the location of the hospital, etc. In this example, we see that attributes may refer to the event itself (e.g. its duration), or to the underlying

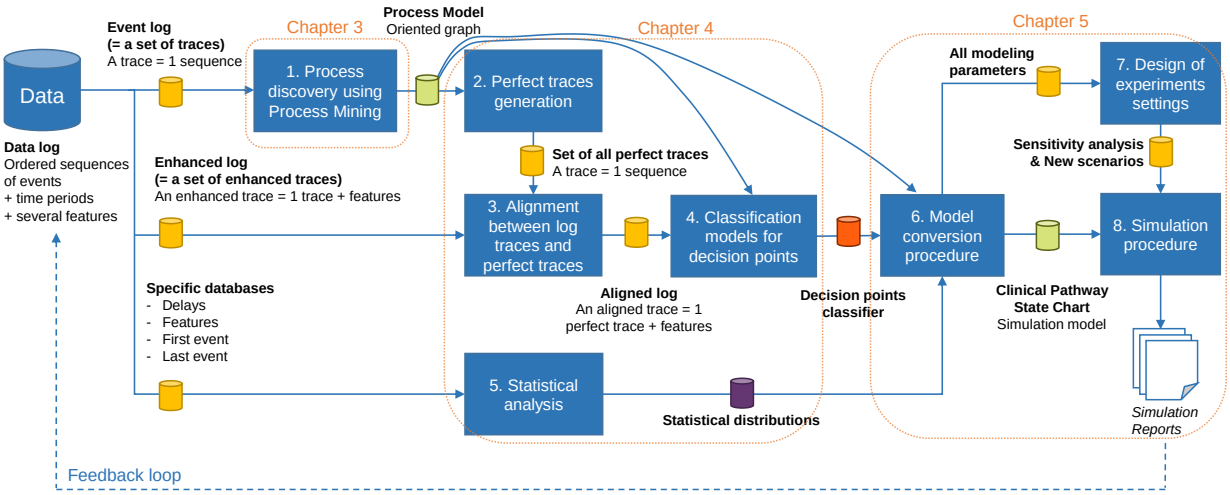


Figure 2.1: The flow-chart methodology of this thesis: from data to performance indicators

instance (e.g the age of the patient), or to the event provider (e.g. the hospital location). All these aspects will be used at different steps of the methodology. Still, the most important attribute of an event is its label. A label is used to globally describe the event, i.e. to summarize what happened during the event. The label is also used to gather several events in a class and to compare such classes (e.g. a hospitalization for a chemotherapy session is different from a hospitalization for an appendicitis). A sample of such a database is shown in Figure 2.2.

Database of hospital events

Case ID (Patient)	Event label (or class)	Starting date	Duration (days)	Location	Diagnosis	Gender	Age	Diabetes
1	Chemotherapy session	01/01/2017	< 1	Lyon hospital	Lung cancer	Female	50	Yes
1	Chemotherapy session	08/01/2017	< 1	Lyon hospital	Lung cancer	Female	50	Yes
1	Chemotherapy session	15/01/2017	2	Lyon hospital	Lung cancer	Female	50	Yes
1	Chemotherapy session	22/01/2017	< 1	Lyon hospital	Lung cancer	Female	50	Yes
2	Stomach surgery	02/02/2017	8	Paris hospital	Appendicitis	Male	65	No
2	Surgery Follow-up	28/02/2017	2	Paris hospital	Post appendicitis	Male	65	No
3	Heart failure	19/01/2017	3	Lille hospital	Heart failure	Male	70	No
3	Pacemaker implantation	27/01/2017	12	Lille hospital	Heart failure	Male	70	No

Figure 2.2: Illustration of a suitable database of events

In this work, we present a new methodology to specifically be able to analyze complex databases. The complexity of a database is multi-sources. A database can be complex if it contains a large number of recorded events (e.g. hospital stays). Similarly, a database's complexity can be the number of concerned instances (e.g. patients). For both the number of events and the number of instances, the main challenge for analytic purpose is to have sufficient computational power. However, there is another aspect of complexity that heavily impacts the way to analyze the data. It is the number of different classes of events (or label of events) in the database. The literature on process mining shows that this criterion is the most complicated to deal with when developing a process discovery algorithm. It is even truer if some of the event classes are noise for the studied process. The first discovery algorithms were unable to deal with a large number

of classes and with noise in the data (van der Aalst, 2004). Further algorithms were more resilient to noise (Weijters et al., 2006) and capable of clustering classes to reduce their number (Gunther and van der Aalst, 2007). Still, existing methods suffer on databases with more than few hundreds classes and a million of events, which is rather common in many databases nowadays. Beyond process discovery, we also face complexity when it comes to the conversion into a simulation model. Not all supervised learning methods are suitable to handle large data sets, and it is not conceivable to manually create a simulation model with hundreds of variables.

The complexity of a database also depends on the actual process that it represents. When the actual system is already considered as complex (as it is often the case in medicine or in disease management), a technical expertise is needed for the interpretation of the data. Even in a single therapeutic area, which means that patients suffer from the same disease, each patient is unique. A generic algorithm for process discovery or modeling could not perform well. The approach shall be flexible and capable of modeling complicated decision rules that occur during a process (medical decisions, surgical acts, natural evolution of diseases, patient's preferences). Here, we intend to analyze any database that concentrates one or several of the above aspects of complexity.

2.3.2 Step 1: optimal process model discovery using process mining

Step 1 is dedicated to the discovery a process model from the data using a process mining technique. Several process mining algorithms have been proposed in the literature to address this process discovery issue (van der Aalst, 2011). Yet, most of them are not suitable for very large and complex data sets which are more and more frequent in real-life applications. Also, to the best of our knowledge, no formal evaluation of the quality of the model generation has already been proposed in the literature. We propose in this thesis a formal definition of replayability, used as a quantitative evaluation metric.

Event log

The raw material of process mining is a specific type of data set, namely an event log. An event log is a set of traces, where each trace is a sequence of timely ordered events. The only requirement is to have an event log such that (i) each event refers to an activity (i.e., a well-defined step in the process, such as a chemotherapy session or a surgery), (ii) each event refers to a trace (e.g., a patient, a client), and (iii) events have a time stamp and are totally ordered. Other information about the involved resources, costs or date may also provide a deeper understanding of the process.

Process model

A Process Model is an abstracted and simplified way to represent a real process, i.e., an event log. It is useful if the model is representative of the data of the log (van der Aalst, 2011). Here, a Process Model is defined as an oriented graph, i.e. as a set of nodes and a set of oriented edges. An advantage of this notation is to be simple to represent and straightforward to interpret. Nodes represent tasks in the process. Arcs, connecting the nodes, represent ordering relations upon the tasks. No theoretical knowledge is required to read a model, unlike Petri Nets and BPMN. Here, all the incoming joins and the outgoing splits of the nodes are exclusive disjunction (XOR): exactly one path is chosen in the flow. There is no need to define complex structures to deal with combinations of XOR/XAND splits or joins. Loops are allowed in the model, but there is no duplicate activities.

Process discovery

Process discovery is one of the most challenging task of process mining. The objective of process discovery is to build a process model that is representative of the behavior seen in the log. The quality of a process model will be evaluated with a quantitative metric. Any metric refers to one or several of the three following dimensions: the model must be highly representative, it must be as detailed as possible, and it must have a low complexity. In Chapter 3 (Optimal Process Discovery), we extensively define the properties that a relevant evaluation metric should have, we propose the first rigorous mathematical formulation of the optimal process mining problem, and we investigate 8 possible score functions called replayability scores. The optimal process mining problem is stated as: "finding the smallest possible process model that maximizes the replayability score." These two objectives, minimizing the size of the model and maximizing the replayability, are antagonistic and no balance can be found between them. Hence, we reformulate the discovery problem as "finding a process model that maximizes the replayability score under a given size constraint". In the second part of Chapter 3, we present two solutions to solve this process discovery problem. The first approach is an Integer Linear Programming model solved with IBM Cplex solver. It performs extremely well for small and middle size problems. The second approach is a combination of Monte-Carlo optimization and tabu search to overcome the complexity related to the huge size of the event logs. Finally, the result is the best possible process model in regards to the replayability function.

Step 1 in short

Input: An event log (i.e. a set of traces)
Action: Process discovery with a process mining technique
Output: A Process model highly representative of the data
Scientific challenges: Mathematical formulation of the optimal process mining problem, definition of a score function and proposition of a solving method for complex and large logs.

2.3.3 Steps 2, 3 and 4: decision point analysis

Step 2 and step 3 are two preparatory steps for the decision point analysis carried out at step 4. The goal of the decision point analysis is to find the features that impact the choice of a given path in the process model. This approach performs better than a simple probabilistic distribution based on historical data. For that purpose, we assume that several features about each trace of the log are available in the database. For instance, if a trace is a patient, its sequence represents all the medical events that the patient had during a given time window, and his features can be age, gender, height, weight and his medical history. To find the rules explaining the different possible paths, we want to learn from the data. This learning step requires that the traces are perfectly re-playable in the model. Otherwise, there exist mismatches between the possible paths of the model and the actual paths observed in the traces. This is due to the fact that the model is an abstracted representation of the data that cannot represent exhaustively all the behaviors seen in the data. Such mismatch creates a modeling bias, especially when dealing with complex and large event logs: only few traces of the log can be used to learn routing rules. Thus, a whole part of the data is ignored and small samples are not sufficient to ensure a significant statistical learning. To close this gap and be able to efficiently analyze decision points, we generate a set of perfect traces (step 2) and we align the actual traces of log with these traces (step 3).

Step 2: perfect traces generation using a process model

Step 2 is the generation of all the possible perfect traces derived from the process model obtained at step 1. The objective of this step is to know all the possible sequences that could be 100% replayed in the process model, unlike most of the actual trace from the event log. We design an automatic procedure that generates such perfect sequences. This procedure is detailed in Chapter 4. As a model can have loops, i.e. there exists a path from a given node to another node of the model which was previously visited in the sequence, it may possibly generate infinite-sized traces. For that reason, a *maximal size* value is imposed to perfect traces.

The generation procedure is the following: starting from any node of the process model, we create as many traces as there are different paths to follow in the graph until we reach the *maximal size*. Any sub-trace of such traces is also added to the set of perfect traces. Hence, for each generated trace, we obtain the set of all traces whose size is between 1 and *maximal size*, and which is perfectly re-playable in the model. The number of perfect traces grows exponentially with the number of nodes and edges in the model.

Step 2 in short
Input: A process model
Action: Generation of all the possible perfectly replayed traces
Output: A set of perfect traces
Scientific challenge: The definition of a size constraint to handle the exponential number of possible perfect traces in practice.

Step 3: Alignment of log traces with perfect traces

Step 3 is the alignment of the original traces of the log with the perfect traces generated at step 2. We present an innovative way of combining a sequence alignment algorithm from the field of bioinformatics to our process mining approach (See Chapter 4). For any given trace of the log, this algorithm allows us to find the closest perfect trace in terms of sequence. Hence, instead of using the original traces of the log for the decision point analysis, we now can use its best counterpart from the perfect traces, and so avoid the mismatch bias.

Enhanced event log

In addition to the previously used event log for process discovery, we now use trace's features. A set of features is defined for all the traces. A trace is defined as a mere sequence of events. We use the concept of enhanced trace to describe a trace with feature values.

Similarity matrix

To compare two traces, i.e. two sequences of events, and evaluate their closeness, we need to be able to compare two events. There are 3 possible different situations when comparing one event of a sequence to one of another sequence: they are the same (=match), they are different (=mismatch) or one event aligns with a gap in the other sequence (=gap insertion). For instance, when comparing the sequences A-B-C and A-B-D, we see that 2 events match out of 3. The next question is to know if A-B-C is closer to A-B-D, or to A-B-E or to A-*(gap)*-C. For that, we use a *similarity matrix*. It is a square and positive matrix whose size is equal to the number of different events in the log. For any pair of events (e_1, e_2) the matrix gives a similarity score between e_1 and e_2; the higher the score is, the more similar e_1 and e_2 are.

Sequence alignment with the Needleman-Wunsh algorithm

For each enhanced trace of the log, we use the Needleman-Wunsch algorithm (Needleman and Wunsch, 1970) to measure its closeness with each of the perfect trace. The perfect trace with the highest score is chosen as its closest perfect match and will be used instead of the original trace for the decision point analysis, using the features of this original enhanced trace. The NeedlemanWunsch algorithm is an algorithm used in bioinformatics to align amino acids or nucleotide sequences (e.g. proteins or DNA / RNA strains). The goal is to identify the regions of similarity among them. It is used to infer structural, functional and evolutionary relationships between the sequences. The NeedlemanWunsch algorithm was one of the first applications of dynamic programming to compare biological sequences. It was developed by Saul B. Needleman and Christian D. Wunsch and published in 1970 (Needleman and Wunsch, 1970). It quantifies the alignment between two sequences by assigning scores (scores for matches, mismatches and gaps). The alignment is scored globally, meaning that it is carried out from beginning till the end of the sequence to find out the best possible alignment.

Step 3 in short
Input: A set of perfect traces + A set of enhanced traces
Action: Sequence alignment with the Needleman-Wunsh algorithm
Output: A set of aligned perfect traces with features, one for each original trace of the log
Scientific challenge: Proposition of an innovative way to avoid the mismatch bias at decision points when using complex and large event logs

Step 4: Creation of classification models for decision points

At step 1, we have discovered a process model reflecting the causal relations between the activities of the event log. Now, we want to gain a better understanding of another perspective of that process. More specifically, here in step 4 we are looking for data features that influence the choice of a path in the model. This type of approach was already conducted by members of the process mining community (Grigori et al., 2004; Ly et al., 2006; Rozinat and van der Aalst, 2006c; Rozinat et al., 2009). Our approach here remains similar to existing methods. The differences are the adaptation of works done on Petri Nets to the causal net models, and the proposition of a full methodology to deal with large event logs (step 2, 3, 4). To analyze the routing choices, we first need to identify the parts of the model where there is a decision point, i.e. where several paths are possible after a node. Then, we need to select as many relevant features from the log as we can to explain this choice. Features can be events' features, but it would only provide a local vision of what happened. It would assume that the decision on the next event is only determined by the current event. Instead of that, we also consider traces' features to analyze the decision points. It allows to include information about what happened previously to the trace and about inherent features of the trace. Finally, we select a classification algorithm that is capable of learning these decision rules from the historical data. As we showed at step 3, using the original traces of the log would lead to a major bias during the learning phase. For that reason, we used an alignment algorithm that provides us a set of perfect traces enhanced with their original trace counterpart's features. This contribution is an innovative way to reinforce the quality of the learned decisions.

Identification of the decision points

A process model is an oriented graph. In terms of oriented graph, a decision point corresponds to a node with multiple outgoing arcs. Since only one path can be chosen by a process instance, each time that several

outgoing arcs can possibly be chosen after a node, a rule must be defined to explicit this choice. There is no way to know the number of decision points in a process model based on the number of nodes and edges. It depends on the structure of the process model, especially in the presence of loops or of terminal nodes with no outgoing edges.

Definition of the classification problem

The classification problem can be stated as follows:"for each node of a process model where there is a decision point, find the probabilities of following each possible path based on the instance's features". A classifier will be created independently for each decision point. For a given node x with a decision point, the learning phase is done by selecting all the aligned traces, among the aligned traces obtained at step 3, that had an event x in their sequence. Each selected trace becomes a learning observation where the target variable is "the next event after x in its sequence".

Selection of a machine learning algorithm

To solve the classification problem, there exist many algorithms (Caruana and Niculescu-Mizil, 2006), namely Support Vector Machines, Decision Trees, Stochastic Gradient Descent, Naive Bayes, Generalized Linear Models, Nearest Neighbors, Ensemble methods, Neural network. Choosing an algorithm to solve a classification problem is a non-trivial task and will strongly impact the quality of the results. Each algorithm performs better than others under specific condition. It heavily depends on the data and the available computing power. A more in depth discussion of the algorithm selection is carried out in Section 4.5.3 of Chapter 4.

Validation of the classification models

For any data mining approach, some of the observations are used to learn the classifier whereas others are used to test and validate it. The validation results will determine if the classifier's predictions are good enough on unseen data. If they are not, then the parameters of the algorithm will be tuned until it reaches satisfactory results. This creates an optimization loop where the objective is to maximize the quality of the classifier and the variables are the algorithm's parameters and the choice of features in the data. This validation process is part of a more general scheme, the CRISP-DM reference model (Cross Industry Standard Process for Data Mining)(Chapman et al., 2000). CRISP-DM is a comprehensive data mining methodology that provides anyone, from novices to data mining experts, with a complete blueprint for conducting a data mining project.

Step 4 in short
Input: A set of aligned traces (= perfect traces with features) + a process model
Action: A machine learning approach to solve a classification problem at each decision point of the model
Output: A classifier model for each decision point of the process model
Scientific challenge: Integration of domain related data into a model than explains the path choices much better than mere routing probabilities.

2.3.4 Step 5: Statistical analysis

Step 5 is dedicated to a statistical analysis of the data to enhance the process model with information about activity duration and waiting times between activities. We also add information about the generation of new traces from the model, such as features distributions and the probability for any node to be the

first or the last event of a trace. The model enhancement is done by extracting information from the log. Execution and waiting times are observed in the data thanks to the time-stamp of events. For both traces' features and execution and waiting times, we fit the historical data to find the closest theoretical random distribution (Bowman and Azzalini, 1997, Chapter 1). Fitting is done by maximizing the likelihood or minimizing the square error. A process model can be enhanced with many other perspectives, depending on what is available in the original data. A common extension is the organizational perspective. It deals with organizational units, roles and resources needed to perform each activity of the model.

Step 5 in short
Input: Historical observations derived from the event log
Action: Data fitting with random distributions
Output: A set of the best-fitting theoretical random distributions for each measure
Scientific challenge: Modeling of behaviors observed in data without preconceived hypothesis such as normally distributed data

2.3.5 Step 6: Model conversion procedure

In the previous steps (1,2,3,4,5), we have seen how to discover a process model from an event log, how to learn complex rules for the decision points routing and how to enrich this model with other perspectives such as process times. Step 6 is the integration of all these notions into a single concept: a simulation model. In Section 5.3 of Chapter 5, we present a formal framework for such an automatic conversion.

Definition of a new type of state chart

To transform the static model into a dynamic simulation model, we use the general concept of state chart. A state chart (also called a finite-state machine) is a mathematical model that describes the behavior of a system. It is an excellent way to model the process steps of an entity, the activity of a resource or the coordination of several entities. It includes the definition of states, transitions, probabilities of activating a transition and a state duration. We enrich this generic state chart definition by proposing a new type of state chart, namely a *clinical pathway state chart*. We use this new concept to be able to specifically study care processes. A care process is often characterized by a long follow-up duration (from months to years), by complex medical decisions and natural evolution of diseases. Our *clinical pathway state chart* intends to capture these aspects and to evaluate their impact on the patient's outcome.

Automatic conversion

The conversion procedure is made of two phases. First, we convert the model's structure. Each node and each edge of the process model is converted into a dedicated type of state. Then, we convert the decision. The formulation of the conversion procedure allows us to automatically reuse it on a completely new data set.

Step 6 in short
Input: A process model + decision points classifiers + random distributions
Action: Conversion into a simulation model
Output: A specific state chart simulation model called a *Clinical Pathway State Chart*
Scientific challenge: Definition of a generic mathematical framework for the conversion procedure which allows for an automatic reuse

2.3.6 Step 7: Design of experiments settings

Step 7 is dedicated to the setup of a design of experiments for the discovered simulation model. The simulation model can be used for different types of analysis. In most cases, simulation is used to test new scenarios. Indeed, after a simulation model has been validated, it is assumed to be a good model to represent the reality. It is capable of capturing the complex interactions of many variables (time, decisions, sequence of events, resources, cost ...). Then, it becomes possible to feed the model with new inputs (for instance with a new cohort of patients who are older than those observed in the actual data) and to evaluate the impact of this change on predefined Key Performances Indicators. The proposed design of experiments is twofold: an automatic sensitivity analysis and a personalized set of scenarios.

Sensitivity analysis

A sensitivity analysis is proposed to evaluate the impact of a wide range of parameters on key performance indicators. Hence, we have at our disposal a tool which is capable of detecting the most impacting factors on the modeled system (patient's feature, the size of the model, the classification algorithm, the similarity score).

Evaluation of personalized scenarios

After determining the crucial parameters through the sensitivity analysis, the practitioners may design specific **scenarios** depending on their own goals. Defining a new scenario requires a deep knowledge of the modeled process. The same user will hardly be able to think about the levers that would improve a process from any domain. For instance, finding a scenario could be either a search for an improvement of the care performance in the pathway management of patients diagnosed with metastatic lung cancer, or the need for an optimal way to reduce the total span time between the submission and the acceptation of a mortgage request. In addition to the domain knowledge, there is also a need for the capability to convert an idea into actionable rules that can be added into the simulation model. For instance, if we want to test the impact of an aging population in a model of the care management of influenza, we just need to shift the random distribution representing the age variable toward slightly higher values. More complex modifications would require more modeling skills.

A simulation model does not necessarily need a new scenario to be run. In fact, running a simulation model with its default parameters, i.e. the parameters automatically derived from the process model and the data, imparts extremely useful knowledge about the process as it currently happens (or at least as it happened in the data).

Step 7 in short
Input: Knowledge about the modeled process
Action: To conceive a new "what-if" scenario to test into the simulation model
Output: A set of specific rules explaining how to modify the simulation model to incorporate the new scenario **Scientific challenge:** It requires in-depth knowledge of the modeled process and the capacity to convert an idea into a set of rules that can be integrated in the model

2.3.7 Step 8: Simulation procedure

Simulation procedure implementation

The discovered simulation model can now be used to simulate new traces, e.g. to simulate the clinical

pathway of new patients. The procedure follows 7 steps.

1. First, a new entity is created.
2. A set of feature values (e.g. age, gender, weight, diabetes) is assigned to the entity based on the random distributions found at step 5.
3. Then, its sequence starts at one of the state node, chosen randomly according to the probability distribution that was also found during the performance analysis. This state becomes the entity's current state.
4. Whatever the state, duration is drawn according to the state specific distribution. It is added to the total time span of the entity.
5. The classifier of the entity's current state is then used to determine the next step of the entity. The entity's sequence might stop there or continue toward another state of the model. The choice is based on the entity's features.
6. A waiting time between the current state and the next state is drawn according to a specific distribution.
7. The next state becomes the entity's current state and the procedure goes back to 4. The procedure ends if the entity's current state has no outgoing transition or if the sequence is stopped prematurely by the classifier at step 5.

Validation of the simulation model

Before evaluating new scenarios with the simulation model, we have to validate it. For that, we run the model with its default configuration, meaning with the discovered parameters and distributions. Then, we use a large set of indicators to evaluate if the model's behavior is close to the original data (number of events, number of entities going through each state and each transition, features' values, mean total duration, etc.). In the case where the simulation results are too far from the original data, we can adjust one or several of the parameters used in step 1,2,3,4,5,6 (e.g. size of the discovered model, the learning ratio of the classifiers, distribution fitting method). In other word, we use a feedback loop to optimize the quality of the simulation model. Once the simulation model has been validated, it is possible to test new scenarios and to extend the results to various case studies.

Test of new scenarios

When the simulation model has been validated by proving that it is close enough to the original data, we can start testing the new scenarios defined at step 7. The power of using a simulation model is to test as many scenarios as we can imagine and to compare them on several objective functions (e.g. total cost of care management, death rate and relapse rate). The results of such a comparison is of major value for decision makers. The simulation is repeated a large number of times to simulate the behavior of a cohort of entities. The exact number of required entities to ensure significant results depends on the desired level of confidence for performance measures. Relevant performance indicators are gathered when entities exit the model (e.g. total duration spent in the model, total cost of care, number of deaths in states, etc.). To ensure a given confidence interval for such indicators (e.g. 95%), the minimal number of replications is given by the formula of Law and Kelton (Law and Kelton, 2000).

Step 8 in short
Input: A simulation model + A set of scenarios to evaluate (a new conceived scenario or the current situation)
Action: Launch multiple simulation runs
Output: A simulation report (Performance measures and confidence intervals) + benchmark of different scenarios
Scientific challenge: Definition of relevant performance measures + enough computing power for multiple runs.

2.4 Note to practitioners

In this thesis, we propose a methodology for the automatic conversion of raw data into performance indicators. It allows for an in-depth analysis of a process. All the work described here was coded and integrated in a computer program, so that it can be re-used on new data. The automatization of the technical steps is the key for a successful transfer of the tool from researchers to practitioners, without specific programming or modeling skills. As the tool can answer various questions about the studied process, it is intended to serve different users in the health-care domain.

The first concerned entities by the use of health data are the hospitals themselves. Hospitals are responsible for recording and storing all the data related to their activities. The hospital staff and managers can then use the program to better understand the care management of any disease in their hospital. They should be able to understand, adjust and improve an abstracted model of their system. As a result, they could find improving drivers and test new ways of managing specific patients.

Our methodology and program are also of interest for any *analyst*, whether he/she belongs to a private consulting firm, a hospital direction board, a public institute for epidemiological studies or the ministry of health. We propose a way to drastically simplify the modeling process of complex systems, while guaranteeing the scientific rigor of the results. It can benefit to any user who has access to a database, for instance in the context of a one-time audit or of an internal reorganization, and who want to take benefit of it. Finally, our work is also intended to promote the use of health data by researchers. We propose very formal and generic frameworks at each step of our work, so that comparisons with future works and re-usability on new data can be made. Our methodology can also save time to researchers who want to dig into new data to study processes.

2.5 Summary and contributions of the thesis

The originality and the scientific contributions of the present thesis are multifold:

- **A comprehensive methodology to automatically convert event logs into a simulation model:** based on existing methodologies in the literature, we added new steps that fully exploit all the data recorded in large event logs. It includes a perfect trace generator from a process model and a sequence alignment algorithm to extract information from imperfectly replayed traces. We also propose new mathematical frameworks for already existing steps, especially with the use of causal nets for process discovery, of classifiers for decision point analysis and state charts for simulation.
- **The first rigorous mathematical formulation of the optimal process mining problem:** we define formally the problem of discovering a process model from an event log in the form of a causal net

while maximizing its representativeness under a size constraint.

- **A hierarchical representation of event relations:** we designed a framework where events can be gathered in high level clusters so that we can capture much information even in small size models.
- **An event log-size independent Monte-Carlo simulation approach for performance evaluation of process models at any desirable precision:** To address the complexity issue of evaluating millions of traces during the optimization procedure of process discovery, we use a sampling strategy that guarantees that the method is scalable for large event logs.
- **Properties of the optimal process models:** based on the mathematical framework defined for the optimal discovery problem, we were able to demonstrate several properties of optimal solutions.
- **A tabu search algorithm for process model optimization:** which is proved by extensive numerical experiments to be superior to state-of-the-art process mining techniques.
- **An innovative combination of a bioinformatics algorithm for sequence alignment and supervised learning classifiers for decision points analysis:** Existing approach struggled to deal with large event logs and to produce automatic and reliable model enhancement with big data. We propose an innovative way to overcome this issue.
- **The definition of a new class of state chart to create a clinical pathway simulation model**: based on the generic concept of state chart, we propose a new version, a so-called Clinical Pathway State Chart. It is a generic concept for the modeling and simulation of clinical pathways, i.e. the follow-up of patients and their conditions both on short term (hospital stay) and long-term (disease evolution over years).
- **A complete validation strategy of the created simulation model:** the simulation model is validated thanks to the comparison of tens of variables measured through multiple runs of the simulation, on the one hand, and from the actual database, on the other hand.
- **A systematic sensitivity analysis of the factors impacting the modeled system:** the simulation model is automatically run over several replications to determine the contributions of any of the modeling parameters and of the data on the modeled system.
- **The application of all the previous points on health-care data**: we applied the presented methodology on real-world data to model and simulate clinical pathways. We show the results of the approach on two main case studies: heart failure and lung cancer.
- **A design of experiments was proposed to evaluate new scenarios:** in the context of the heart failure case study, we demonstrate the power of the simulation model to evaluate what-if scenarios.

Perspectives of the proposed approach are numerous and will be detailed in the corresponding technical chapters in the following.

Chapter 3

Optimal Process Mining for Large and Complex Event Logs

Contents

Abstract

This chapter addresses the problem of process discovery from large and complex event logs. We depart from existing literature and formulate the problem of optimal process discovery. A formal mathematical programming model is given based on a novel hierarchical structure of the event logs. Desired properties of event trace score functions are described and properties of optimal process models proved. A combination of Monte-Carlo optimization and tabu search is proposed to overcome the complexity related to the huge size of the event logs and the combinatorial solution space. Numerical results show that our approach is suitable for large event logs and that it performs better than state-of-the-art approaches. We also demonstrate the applicability of our method on a real case study in health-care. This study illustrates the benefits of combining techniques from the Operational Research and the Process Mining fields.

3.1 Introduction

A process is a collection of related activities that serve a common goal. This definition applies widely. In automotive industry, assembly lines are designed to ensure the right order of production steps. In the banking sector, loan applications follow strict procedures with several intermediate validation steps. In health-care, the patient care process is the so-called clinical pathway. It describes a set of treatment activities, such as consultation, imaging examination or surgery, with the common goal of treating a patient (Rebuge and Ferreira, 2012). In all domains, the design of a process tends to standardize work practices. It can be decomposed into operational purposes: planning activities, assigning human resources, reducing practice variability, minimizing delays or decreasing costs while maintaining quality.

Process discovery from data has been studied in the literature. It is mainly done by using Business Process Management and Analysis (BPM and BPA) (van der Aalst, 2012), Data Mining or Process Mining techniques. The focus of data mining approaches is slightly different from the two others. Based on historical observations contained in a data set, data mining is used to find hidden patterns among different features of these observations (Tan et al., 2005). Even if data mining includes tools such as sequential pattern mining, it remains data-centered and does not consider the concept of end-to-end processes. On the other hand, BPM and process mining consider such processes and are good complement to data mining approaches. BPM has little interest in event data and is mostly model-driven (van der Aalst and Weijters, 2004; van der Aalst, 2012). Models are often hand-built by experts. The goal of process mining is to bridge the gap between BPM and data mining approaches, using data to study business processes. Process mining is not used to find data patterns, but rather to find process relationships in the data which provide an overview of activities in the process, of deviations and of process performance such as throughput, bottlenecks and discrepancies. Several process mining algorithms have been proposed in the literature to address this process discovery issue (van der Aalst, 2011). Yet, most of them are not suitable for very large and complex data sets which are more and more frequent in real-life applications.

This chapter is motivated by health-care applications in which each patient is naturally unique. There are potentially as many different processes as the number of patients in the database, which may be millions or more. In this chapter, we present a new approach for Process Discovery in complex logs by combining process mining and optimization techniques. Our scientific contribution is multifold: (1) the first rigorous mathematical formulation of the optimal process mining problem; (2) a hierarchical representation of event relations; (3) an event-log-size independent Monte-Carlo simulation approach for performance evaluation of process models at any desirable precision; (4) properties of the optimal process models, (5) a tabu

search algorithm for process model optimization which is proved by extensive numerical experiments to be superior to state-of-the-art process mining techniques. We hope that this work opens a new avenue of applying advanced combinatorial optimization techniques for complex process mining problems.

The remaining of the chapter is organized as follows: a literature review of process mining is given in Section 3.2. Basics of process mining are described in Section 3.3. The optimal process mining problem is set in Section 3.4. A preliminary resolution method, based on integer linear programming and dedicated to small-size event logs is proposed in Section 3.5. A more advanced optimization method is described in Section 3.6 and Section 3.7 to handle logs of any size. A numerical experiment is designed in Section 3.8 to compare the new method with existing ones. Finally, conclusion and perspectives are discussed in Section 3.9, while a real-life case study is presented in Chapter 6.

3.2 Literature review

The goal of process mining is to extract new information about processes from event logs (van der Aalst, 2011). The field of process mining emerged in the early 2000s and the bases were formalized by (van der Aalst, 2004) in 2004. process mining aims at providing an impartial view of a process based on what really happened, and not on a supposed organization. The use of process mining is motivated by two observations (van der Aalst et al., 2003): more and more information is stored in information systems and techniques from Business Process Management have reached their limits. BPM techniques only study theoretical processes, unlike process mining which studies the actual behavior as it happened (van der Aalst, 2011). It can be applied in various systems and domains (industry, administration, finance, health-care, etc.) (van der Aalst, 2011). The raw material of process mining is a specific type of data set, namely an event log. A log is a set of traces, each trace being a sequence of ordered events. The only requirement is to have an event log such that (i) each event refers to an activity (i.e., a well-defined step in the process), (ii) each event refers to a trace (e.g., a patient, a client), and (iii) events have a time stamp and are ordered.

There are three types of process mining approaches (van der Aalst, 2011). The first is *Process Discovery*. The goal is to analyze the control-flow perspective. Process Discovery is concerned with the process behavior, namely the activities in the process and their order of execution. It results in the creation of a process model, which is unknown beforehand and which reproduces the behavior of the recorded events. Examples of Process Discovery techniques include the Alpha-Algorithm (van der Aalst, 2004), the Fuzzy Miner (Gunther and van der Aalst, 2007), the Genetic Miner (van der Aalst et al., 2005) or the Heuristic Miner (Weijters et al., 2006). The second type of approaches is the validation of a preexisting model and is called *Conformance checking*. It quantifies the difference between a model and an event log by using a conformance metric (Rozinat and van der Aalst, 2008). This step is required before starting the third approach: the *extension* of a model. Starting from a preexisting model, the aim is to enrich it with observed data (e.g., add organizational and time perspectives). Many of these process mining techniques are available in the ProM framework (van Dongen et al., 2005), an open-source Java application developed and expanded by the process mining academic community.

Here, we will focus on process discovery and model evaluation. Different process discovery algorithms have been proposed and are suitable for different contexts. The Alpha-Algorithm (van der Aalst, 2004) performs poorly on noisy logs with infrequent behaviors. The Heuristic miner (Weijters et al., 2006) is capable of dealing with infrequent activities, but it tends to create the infamous spaghetti models when there are too many activities in the log. The Genetic Miner (van der Aalst et al., 2005) becomes extremely time consuming to create a process model when there are many activities in the log, and hence is unusable

in practice. A challenge of using process mining in health-care is to deal with the high variability of cases. There are almost as many pathways as there are patients, due to the uniqueness of each person's health. The use of classical process mining techniques produces the so-called spaghetti-like models that are too complex to be comprehended (van der Aalst, 2011; Rebuge and Ferreira, 2012; Gunther et al., 2010). The Fuzzy Miner (Gunther and van der Aalst, 2007) used the zooming cartography metaphor to reduce the model's level of detail. Models are more aggregated and simpler to read. It works by applying several successive graph reductions based on thresholds. However, the fuzzy miner does not consider any semantic significance related to the domain, therefore there is a risk of aggregating irrelevant activities together to a cluster. Regarding this limitation, (Bose et al., 2012) proposed hierarchical discovery approaches to deal with detailed event log and less structures process models. Different from the fuzzy miner, the hierarchies are obtained through the automated discovery of pattern abstractions (Bose and van der Aalst, 2009). It is proved that the discovered patterns always have its specific domain semantics.

Data pre-processing techniques also addressed this issue (Ekanayake et al., 2013). The logs are split in smaller homogeneous logs to reduce the size of the discovered models, which is related to comprehensibility. Still, it may produce too complicated models or it may require advanced settings which compromises an automatic and repeatable discovery. Here, we address the complexity issue by using a threshold on the maximum size of the models. The main limitation of most process mining algorithms lies in the size of the event logs they can handle. An algorithm will have a widespread utility if it can deal with "big" event logs. A log can be big in two different ways: it may contain a huge number of traces or it may contain many different activities. Most existing process mining algorithms scale very badly in the number of activities (Verbeek and van der Aalst, 2015).

Lack of a common framework to compare existing algorithms on a common basis is also highlighted in (on Process Mining , 80 authors). In 2011, a standard benchmark of existing discovery algorithms was to be made (on Process Mining , 80 authors). Representative benchmarks and reference logs are badly needed to test process mining algorithms on some common basis. To the best of our knowledge, very few works attempted to take up this challenge. The first effort toward a common evaluation framework for process mining algorithms was done by (Rozinat et al., 2008). The authors presented the necessary components of an evaluation framework to enable process mining researchers to compare the performance of their algorithms. These components are process model quality metrics, common data sets for benchmarks and model formalism verification (e.g., check if duplicate tasks or self-loops are allowed in the mined model). They also performed an extensive benchmark of 6 algorithms by comparing their performances on simple logs. The concepts from (Rozinat et al., 2008) were extended by (Wang et al., 2013) and (Weber et al., 2013). In (Wang et al., 2013), they defined several new comparison metrics. Instead of comparing models on a fitness value over a log, they focused on comparing models based on their structure (number of blocks, loops, etc.). They used this approach to efficiently select the most suitable process mining algorithm for a given enterprise. In (Weber et al., 2013), they introduce probabilistic automaton as a unifying representation of several representation languages. In (Rozinat et al., 2008; Wang et al., 2013; Weber et al., 2013), they all compared models obtained as Petri Nets. This formalism is strong and allows for reliable comparisons. However, no comparison includes the Fuzzy Miner algorithm which is acquiring a growing popularity in practical use, especially in its extended version available in the DISCO software developed by Fluxicon.

Here, we aim at contributing to the effort of easier comparisons of discovery algorithms, and especially for models using more intuitive notation for non-experts than Petri Nets. We propose to compare models based on a quality metric. For that, we introduce a new intuitive way to measure the conformance of a model. Our metric can be seen as an extended version of the parsing measure defined in (Weijters et al.,

2006). Our metric can better differentiate and deal with a wide range of flexible behaviors.

The work of (van der Aalst et al., 2005) is the closest work to the present approach. A genetic algorithm (GA) was proposed for computing the "most" appropriate Petri net model of an event log with incomplete events and event noises. Petri net models are transformed into so-called Causal Matrices on which genetic operators are defined. A fitness score based on parsed activities and completed traces in a token game is used. Note that there is no one-to-one relation between Petri net models and causal matrices and the problem under consideration was not formally stated. Further it was observed that the genetic algorithm performs well on small logs but struggles with large logs with many traces and activities. Optimization tools are also used in (van der Werf et al., 2008). Integer linear programming (ILP) was used to solve theory-of-region-based linear systems such that the language of the resulting Petri net is exactly equal to a given event log language. Note that this ILP approach does not attempt to find the "most" appropriate model.

To summarize, there has been no attempt to rigorously formulate process mining optimization problems. The optimization criteria and the feasible process model space have not been clearly defined. Our work is, to the best of our knowledge, the first attempt to fill this gap. In Section 3.5, a preliminary version is proposed. The ILP process mining model used to optimize a simplified objective function does not fully capture flexible behaviors in traces. A rigorous mathematical programming model will be proposed for process mining optimization. Event log complexity will be addressed by (i) a hierarchical structuration of event relations and (ii) a scalable Monte-Carlo simulation approach for evaluation of large event logs. Properties of optimal process models will be addressed. A tabu search approach will be proposed to solve the optimization problem.

3.3 Basics of Process Mining

This section introduces basic concepts of process mining used in this chapter. It includes the concepts arising from the data, the notion of process model and quality metrics. The readers are referred to (van der Aalst, 2011) for more details of process mining.

3.3.1 Event logs

Process mining is a data-driven approach. The goal is to extract useful information from existing data sources, so-called event logs. The followings are formal definitions of relevant concepts including events, traces, and logs from (van der Aalst, 2011; Gunther, 2009).

Definition 1. (Event) Let $A_{Event} = \{a_1, ..., a_p\}$ be a finite set of attributes (time-stamp, activity type, case ID, duration, ...), $p \in \mathbb{N}$. An event defined on A_{Event} is a set of p values, one for each of the attributes. Each event is uniquely determined by the combination of all its attribute values.

Definition 2. (Trace) Let T be a set of events, a trace σ is an ordered sequence of T: $\sigma = \langle c_1, ..., c_n \rangle$, where $\forall i \in [\![1, n]\!]$, $c_i \in T$. $n \in \mathbb{N}$ is the trace's length. The set of all the traces over T is denoted T^*.

Definition 3. (Log) Let T be a set of events, a log L over T is a non-empty set of traces over T: $L = \{\sigma_1, ..., \sigma_m\}$, $m \in \mathbb{N}$ and $\forall i \in [\![1, m]\!], \sigma_i \in T^*$. The events of a given log are defined on the same set of attributes (with different values).

Example. The goal of our case study is to address the process mining of a well-structured and exhaustive hospitalization database. It contains the record of each hospital stay for any patient in France from 2006 to

2015 for about 15 million patients and 280 million stays. In the generic process mining lexicon, a patient is a trace, a stay is an event and the entire database is a log. The attributes are patient features and medical diagnosis.

Definition 4. (Event Class) Let A_{Event} be a set of attributes. An Event Class is a subset of the attribute vector space defined on A_{Event}. Let T be a set of events and C be the set of all the Event Classes. Alternatively, the function "*Class*" maps each event of T to an Event Class, $Class \in T \rightarrow C$. The set of event classes of T is $C(T) = \{Class(e) \mid e \in T\}$.

Example. In the health-care process mining of (Prodel et al., 2015), event classes were defined by an attribute describing the medical reason of the stay: the diagnosis. This data field is filled using the 10^{th} International Classification of Diseases. Any event was assumed unique as two stays could not happen at the same time, for the same patient and the same medical reason. However, two stays were said similar if they have the same class. For example, an appendicitis operation is the medical reason of the stay, but two stays may be different if they last 2 or 5 days.

Events are **unique**. Two events related to the execution of the same activity are not identical events. The concept of *Event Class* is introduced to describe the relations among events. Events of the same class are considered **similar**. This notion of class is extremely important in the search of important trace patterns. Due to the uniqueness of the events, each trace is also unique at the event level. The only way to model precisely the underlying process would be to represent each of the trace entirely, which is impracticable for systems with a huge number of traces. The concept of class will allow us to identify commonality of traces.

In the remaining, events are assumed to have at least 3 attributes: a time-stamp, a trace ID and a class. After mapping each event to its class, the order of events in a trace still holds as the time-stamps remain unchanged. Hence, for a given log, a class is said to be directly followed by another if there exists at least one trace in the log for which the two classes are following each other. It can be formalized as follows:

Definition 5. (Direct following relation) Let T be a set of events, L a log over T and $C(L)$ the set of event classes of L. The direct following relation among classes of L is defined as follows: let C_1, $C_2 \in C(L)$, $(C_2 \Rightarrow C_1) \iff (\exists\ \sigma \in L, k \in [\![1,(n_\sigma - 1)]\!] \mid \sigma(k) = C_1 \wedge\ \sigma(k+1) = C_2)$. Then (C_1, C_2) is called a *transition* over L.

Definition 6. (Transition set) Let T be a set of events, L a log over T and $C(L)$ the set of event classes of L. The set of transitions of L is $E_{max} = \{(C_1, C_2) \in C(L) \times C(L) \mid\ C_2 \Rightarrow C_1\}$.

The direct following relation between event classes is the starting point of process discovery for most process mining algorithms (van der Aalst, 2011). For instance, the Alpha miner algorithm (van der Aalst, 2011) builds a Petri Net with all existing direct relations whereas the Heuristic miner algorithm (Weijters et al., 2006) only considers the most frequent transitions. In this chapter, we use the direct following relation to define evaluation metrics of process models.

The previous definitions lay formal foundations of the data concept. It allows us to introduce the abstract concept of *process model*.

3.3.2 Process model

A process model (PsM) is an abstracted and simplified way to represent a real process, i.e., an event log. It is useful if the model is representative of the data of the log (van der Aalst, 2011). A model is always

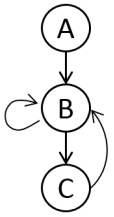


Figure 3.1: Example of a process model with 3 nodes and 4 arcs

created by using a notation formalism. Several notations are available (Petri Nets, BPMN, Markov chain, Flowchart, PERT, ...). Petri nets are often used in the context of process mining. They are used for process discovery by the Alpha-algorithm and the region-based techniques (van der Werf et al., 2008), and also for conformance checking (Rozinat and van der Aalst, 2008). In our approach, we choose the Causal Net of (Gunther, 2009).

Definition 7. (Process Model) A process model PsM is composed of a set N of nodes (event classes), and a set E of arcs (transitions). Let T be a set of events, L a log over T, $PsM = (N,E) = (\{n_1,...,n_x\},\{e_1,...,e_y\})$ where $\forall i \in [\![1,x]\!]$, $n_i \in C(L)$, and $\forall j \in [\![1,y]\!]$, $e_j \in E_{max}$.

Example. Let $T = \{A,B,C,D,E\}$ be a set of events, $L = \{ABCD, ABB, ABCB\}$ be a log over T containing 3 traces. Then, $PsM(L) = (\{A,B,C\},\ \{(A,B),(B,C),(B,B),(C,B)\})$ is a process model of L. Figure 3.1 gives a graphical representation of $PsM(L)$.

An advantage of this notation is to be simple to represent and straightforward to interpret. Nodes represent tasks in the process. Arcs, connecting the nodes, represent ordering relations upon the tasks. No theoretical knowledge is required to read a model, unlike Petri Nets and BPMN. Here, all the incoming joins and the outgoing splits of the nodes are exclusive disjunction (XOR): exactly one path is chosen in the flow. There is no need to define complex structures to deal with combinations of XOR/XAND splits or joins. These three global criteria need to be defined by a quantifiable metric. They are discussed in the following.

3.3.3 Quality metrics

After defining a process model, we now present a way to evaluate its quality. Our objective is to build a process model metric referring to one or several of the three following dimensions:

- The model must be **highly representative**.
- The model must be as **detailed** as possible.
- The model must have a **low complexity**.

First, we discuss the choice of a metric to assess how representative a model is. It refers to how the behavior in a log is correctly captured by the model. For a process model, the representativeness of a process model regarding a log is measured by the **replayability**. The idea is to take each trace of the log and to try to replay its sequence through the nodes and arcs of the model. The replayability is also found by the names of fitness or fidelity. The idea of replayability metrics is not new in the scope of process mining (Huang and Kumar, 2012; van der Aalst, 2011; Rozinat et al., 2008). Currently, no metric stands out as a standard for any type of model. When the process model is written as a Petri Net, the token

game is often chosen as the reference quality metric (Rozinat et al., 2008; Rozinat and van der Aalst, 2008; van der Aalst et al., 2005). It counts the number of missing and remaining tokens in the Petri Net when replaying the traces. In (Weijters et al., 2006), the fitness measure is defined as the number of correctly parsed events divided by the number of events in the event log. This metric is similar to the Petri Net token game but with a causal matrix representation. It performs poorly on flexible logs, when models lack completeness. In (Rozinat et al., 2008), the authors provided a survey of existing metrics proposed before 2008 for the quality evaluation in different types of models (completeness, soundness, parsing measure, fitness, appropriateness, precision, footprint, etc.). Many other metrics were also used since then. In (Huang and Kumar, 2012), the authors defined several quality metrics based on the structure of the model (number of self-loops and activity blocks). Their goal was to use a metric balancing both a model's *Fidelity* and *Specificity*.

In Causal Nets notation, other metrics were used (Weijters et al., 2006). They measured several straightforward measures (e.g., frequency of activities or number of direct following relations) that are reliable when dealing with noise-free logs and simple traces. However, these metrics perform poorly on flexible logs (e.g., with diverse and complex trace behaviors). In (Gunther, 2009), a metric is defined to overcome this issue and match the Causal Net notation.

The **complexity** of a process model is usually defined as the number of its components. Nodes and arcs are considered separately. Let $PsM = (N,E)$ be a process model, then

$$Node\ Complexity(PsM) = |N| \quad \text{and} \quad Arcs\ Complexity(PsM) = |E| \tag{3.1}$$

These measures are directly linked to the readability of the model (Mendling et al., 2007). Models with higher complexity necessarily have higher replayability as they allow for more traces. Hence, maximizing replayability and minimizing the complexity are contradictory objectives.

In this chapter, we propose two complementary approaches to solve the optimal process discovery problem. The first approach is a preliminary work where the discovery problem is defined as an integer linear programming model (Section 3.5). This approach was found to perform well on rather small logs, but it lacks efficiency on larger logs and the replayability function to maximize only considers global behaviors of the traces. In the second approach, we provide a new replayability metric which better captures highly flexible behaviors (Sections 3.6 and 3.7). To do so, our metric needs to use a wide range of values to express both general replayability of traces and the small variations in their sequences. It is important to well discriminate the differences of the sequences. The notion of **detail** of a model is also captured in the new replayability function we define in section 3.6. In both cases (preliminary and advanced replayability), in order to balance replayability and complexity, our approach is to set a complexity threshold (maximum number of nodes or arcs) and to maximize the replayability under this constraint. Similarly to (Gunther, 2009) who introduced a framework to show different levels of abstraction of a process model, we allow the choice of different complexity thresholds. It provides different levels of insight in the model. The main difference between our work and (Gunther, 2009) is that we set the size of the model first before starting the optimization step. Hence, for each size, we provide an optimal model. In (Gunther, 2009), one model of maximal size is created (i.e. all classes are included in the model), which is then aggregated in smaller models without optimality guarantee.

3.4 Problem description and mathematical formulation

This section is dedicated to the formal definition of process model optimization problem for a given event log. For this purpose, we first provide an overview of the mathematical formulation. We then provide precise definitions of different components of the optimization model, including a hierarchical structure of the event classes and the replayability of traces in the event log.

3.4.1 Mathematical formulation overview

The purpose of our work is to determine the most representative process model subject to model complexity constraint. More specifically, we consider a given log L defined on a set C of event classes. The problem consists in determining a process model $PsM = (N,E)$ in order to maximize some replayability subject to some process model complexity constraint. Each node n is to be selected from a given set $S \subseteq 2^n$ of event class subsets to be defined in Subsection B. More specifically, our process model optimization, formulated as a deterministic optimization problem (DetOpt), is defined as follows:

$$\text{(DetOpt)} \max_{PsM=(N,E)} R(PsM,L) \tag{3.2}$$

with

$$R(PsM,L) = \frac{1}{||L||}\sum_{\sigma\in L} R(PsM,\sigma)$$

subject to

$$C(n) \in S, \forall n \in N \tag{3.3}$$
$$E \subseteq N \times N \tag{3.4}$$
$$C(n) \cap C(n') = \varnothing, \forall n,n' \in N \tag{3.5}$$
$$||N|| + ||E|| \leq U \tag{3.6}$$

where $R(PsM,\sigma) \in [0,1]$ and $R(PsM,L) \in [0,1]$ are respectively the replayability scores of trace σ and event log L to be defined in Subsection 3.6, $U \geq 2$ is the process model complexity bound, $||L||$ is the number of traces in L. Constraint (3.3) links each node to an event class subset in S, constraint (3.4) defines the arcs, constraint (3.5) ensures that each event appears in at most one node, constraint (3.6) is the process model complexity constraint.

Remark 1. One important feature of our model is to allow assignment of more than one event class to the same node. This is necessary to build a process model of limited complexity. The meaningfulness of the process model strongly depends on the closeness of event classes assigned to the same node.

Remark 2. Relation (3.2) defines the replayability of a process model as the average replayability score with respect to different traces. It gives equal weights to different traces. Results of this chapter easily extend to the case of weighted replayability score, i.e., $R(PsM,L) = \sum_{\sigma\in L} w_\sigma R(PsM,\sigma)$.

Remark 3. Constraint (3.6) bounds equally node complexity and arc complexity. Again, the results of this chapter can be adapted to the case with separate node complexity limit and arc complexity limit. The complexity limit allows the construction of representative process models with different degree of event class granularity (i.e., details).

Remark 4. Constraint (3.5) ensures no duplication of any event class in the process model. Its relaxation is an interesting research avenue leading to better model representativeness but needs to be treated carefully due to event location ambiguity in the process model.

The deterministic optimization formulation (3.2) can be transformed into an equivalent stochastic optimization formulation as follows:

$$\text{(StochOpt)} \max_{PsM=(N,E)} R(PsM, L) \tag{3.7}$$

with

$$R(PsM, L) = E_{\sigma}[R(PsM, \sigma)]$$

subject to (3.3)-(3.6) where the random trace σ has equal probability of being any trace in L, i.e., with probability $\frac{1}{||L||}$. The stochastic optimization model can then be approximated by the following Monte-Carlo optimization problem:

$$\text{(MCOpt)} \max_{PsM=(N,E)} \hat{R}_K(PsM, L) \tag{3.8}$$

with

$$\hat{R}_K(PsM, L) = \frac{1}{K} \sum_{k=1}^{K} R(PsM, \sigma_k)$$

subject to (3.3)-(3.6) where σ_k are i.i.d. uniformed sampled traces from L.

Remark 5. The remarkable feature of the Monte-Carlo optimization model is that $R(PsM, \sigma_k)$ are i.i.d. random variables, upper bounded by 1 here as the replayability scores will be scaled to be a number in $[0, 1]$, implying that $Var[R(PsM, \sigma)] \leq 1$. As a result, even though the event log L can be of huge size with millions of traces, **a small finite number of sampled traces is enough to ensure an unbiased Monte-Carlo replayability estimation with any desired precision**, i.e., any given confidence interval length. This salient feature will be exploited to speed-up our optimization algorithm.

Remark 6. Even though ILP has been used in (van der Werf et al., 2008) to determine a Petri net model having the same state graph as the event log and genetic algorithms in (van der Aalst et al., 2005) to determine a Petri net model, there is no rigorous formulation of the process mining optimization problem. It is often unclear what criteria to optimize and under what constraints.

3.4.2 Hierarchical structure of the event classes

Our key idea for compact representation of potential assignments of event classes to the same node is the following powerful yet flexible hierarchical structure of the event classes.

Definition 8. (Hierarchical event structure) The set $S \subseteq 2^n$ of event class subsets defined on a set C of event classes has the following hierarchical structure: (i) $n \in S, \forall n \in C$, and (ii) $\forall n, n' \in S$, either $C(n) \cap C(n') = \varnothing$ or $\big(C(n) \subset C(n')$ or $C(n') \subset C(n)\big)$.

Event classes with such hierarchical structure can be represented by a single **hierarchical tree** with all basic event classes as leaves and the whole event class set C as the root if $C \in S$, or by a collection of disjoint hierarchical trees if $C \notin S$. Nodes of the hierarchical trees correspond to different meaningful event

class aggregations. Figure 3.2 shows an example of a hierarchical tree obtained with health-care data (a sample of diagnosis for French hospital stays). Each class represents a medical diagnosis. The higher in the tree, the more aggregated the information is. The highest level of aggregation is the general group of the "diseases of the circulatory system". It can be split in 3 sub-classes "chronic ischemic heart disease", "cerebral infarction" and "secondary hypertension". Similarly, this classes can be split again in even more precise diagnoses.

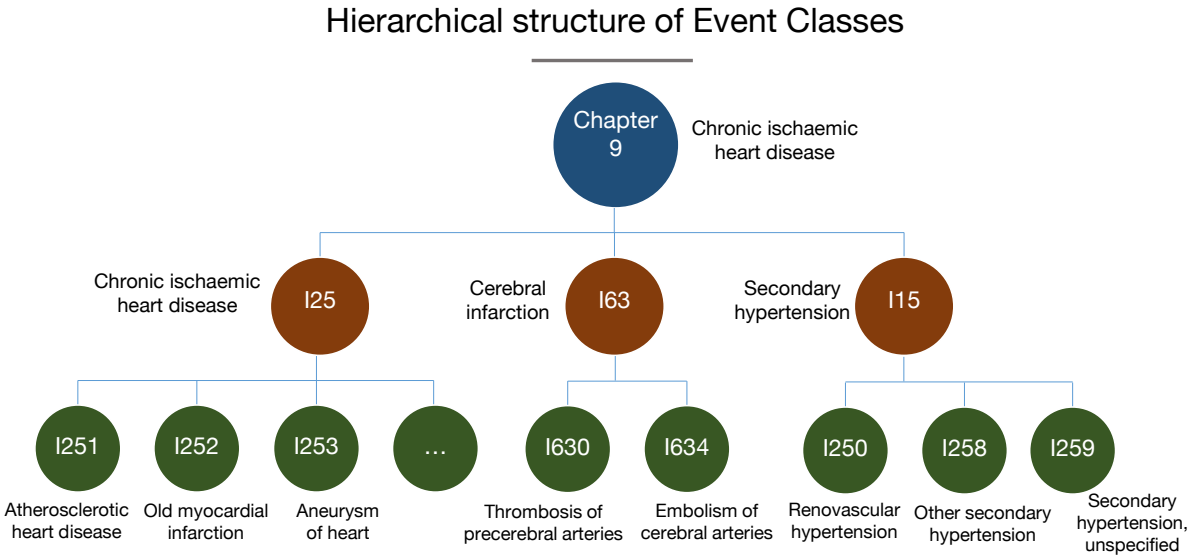


Figure 3.2: Hierarchical tree of event classes on health data, a node is a medical diagnosis. Medical specialties are split in 22 different chapters of the International Classification of Diseases.

With this new event class structure, assigning similar event classes to the same node in a process model is equivalent to directly assign upper level event class subset in the hierarchy. The drawback of using high-level event class subsets is the potential loss of precision. A penalty value associated to high level event class subsets is integrated in the replayability function (Section 3.6). Hence, the event class aggregation will be used only when the gain in number of elements compensate the loss of details.

From now on, any event class subset is called a **cluster** as it contains several basic event classes. A basic event class is sometimes called an event cluster, i.e., a cluster with only one element. In the remaining of this chapter, the term of "event class" (or just "class") is used when describing a trace's sequence, whereas "cluster" is used when describing nodes of a process model.

Remark 7. The idea of grouping similar event classes into one single node was also proposed by (Gunther and van der Aalst, 2007). In their *Fuzzy Miner Algorithm*, the authors created an **aggregation mechanism** to limit the number of nodes to display. They used the analogy of a road map to describe a process model: the number of elements to display depends on the expected level of information asked by the user. In cartography, at the highest level of abstraction of a country map, only highways and major cities are shown. When looking specifically to a neighborhood map, particular houses and small roads become useful to display. This *Fuzzy Miner* has become the predominant process Mining algorithm in practical use with real flexible data. The aggregation is made by first scoring each class seen in the log with a significance metric (e.g., its frequency in the log). Classes with a significance lower than a user-defined threshold are candidates for aggregation. Finally, if two or more classes are found correlated enough (having close names or same duration, occurring both in a short time window, etc.), they are aggregated in one "super-node".

The advantage of the approach of (Gunther and van der Aalst, 2007) is to let the user choose a desired level of abstraction on practical cases by choosing an aggregation threshold. Nevertheless, their aggregation mechanism does not allow grouping low significance classes with high significance classes, even if they are strongly correlated. In addition, even if the *Fuzzy Miner* only uses a reasonable computation time to build process models, the resulting models are not optimal with regard to any quality measures. It may even reveal impracticable to just find outstanding models when trying to set the 5 thresholds of the *Fuzzy Miner*.

Remark 8. Instead of merging only specific classes based on their significance value as in (Gunther and van der Aalst, 2007), our approach allows for any classes to be aggregated. A correlation metric is measured for any tuple of event classes found in the log. Only tuples having a correlation above a threshold are considered feasible and usable in a model. The correlation metric can be defined in two ways: (i) as some generic functions of classes attributes (duration, name, co-occurrence, etc.) (Gunther and van der Aalst, 2007); (ii) by expert opinion or domain knowledge on similarity of event classes (e.g., treating a broken shin or a broken forearm require similar material and care, and thus can be considered as similar activities).

Remark 9. If $C \in S$, the problem (DetOpt) has at least one feasible process model with a single node with $C(n) = C$ and with a self-loop arc. It will serve as a benchmark solution in our numerical experiments.

3.5 A preliminary approach for optimal process discovery

In this section, we propose a first mathematical model to solve the optimal process discovery problem. We define the replayability score, i.e. the objective function to maximize, as the weighted sum of several global measures between the log and the discovered model. Then, an integer linear programming (ILP) model is proposed to find the optimal solution.

3.5.1 Optimization objectives

A model is representative of the log if it highlights the paths with high probabilities and high number of involved traces. It does not contain unnecessary elements and it emphasizes the most important ones. We use the notion of significance to compare the elements of the future model (nodes and edges) (Gunther, 2009). Adapting some metrics of (Gunther and van der Aalst, 2007), we use weighting factors δ and ε so that $\delta + \varepsilon = 1$ ($\delta, \varepsilon \in [0,1]$). Then, the significance of a cluster C_i in a model is:

$$S(C_i) = \delta.Frequency(C_i) + \varepsilon.Routing(C_i)$$

where

$$Frequency(C_i) = \frac{NB\ occurrence(C_i)}{\max\limits_{C_j|(C_i \cap C_j = \varnothing \vee C_j = C_i)} (NB\ occurrence(C_j))}$$

and

$$Routing(C_i) = \frac{|NB\ input(C_i) - NB\ output(C_i)|}{NB\ input(C_i) + NB\ output(C_i)}$$

NB stands for "Number of" (integer value). The significance is a value in $[0,1]$ and it is composed of both the relative frequency of the class and a specific routing feature. A cluster has a high significance if it often occurs in the log, and if it has a large difference in the number of incoming and outgoing transitions

in the log. The significance of a transition is only based on its relative frequency:

$$S(E_i) = \frac{NB\ occurrence(E_i)}{\max\limits_{E \in E_{max}} (NB\ occurrence(E))}\ , E_i \in E_{max}$$

Then, we define two quality criteria of a process model $PsM = (N, E)$ discovered from a log L. First, the level of details evaluates the proportion of significance in the model compared to the overall significance of the log. The *Detail* function represents both the generalization and the precision aspects of the model. Let γ, $\theta \in [0,1]$ be weighting factors, and $\gamma + \theta = 1$, then:

$$Detail(PsM) = \gamma \frac{\sum_{C_i \in N} S(C_i)}{\sum_{C_i \in C(L)} S(C_i)} + \theta \frac{\sum_{E_j \in E} S(E_j)}{\sum_{E_j \in E_{max}} S(E_j)}$$

Secondly, the conformance represents the proportion of included transitions in the model compared to the original log. It is a simplified conformance metric that reinforces the importance of the transitions of the model. As the objective is to build the main pathways (successions of several activities), it fits our idea that the model should highlight the possible paths of the log:

$$Conformance(PsM) = \frac{|E|}{|E_{max}|}$$

The *Detail* and the *Conformance* have values in $[0,1]$. This is a multi-objective optimization problem. We choose to use a linear combination to reduce the problem to only one objective function. Let α, $\beta \in [0,1]$ be weighting factors, and $\alpha + \beta = 1$:

$$Quality(PsM) = \alpha.Conformance(PsM) + \beta.Detail(PsM) \tag{3.9}$$

We intend to find a process model that maximizes such quality value. Other approaches could be investigated to solve such a multi-objective problem (no-preference method, a priori and a posteriori methods, or interactive methods).

3.5.2 Modeling hypotheses

In order to build our ILP model for process discovery, we make the following hypotheses.

1. A given event class cannot be both represented in a cluster and as itself in a model's node. Thus, if a class is a node in a model, any cluster including this class cannot be in another node of the model. Such constraint compacts the model and emphasizes the central nodes. We assume that a precise labeling of the event classes is done ahead so that it smooths the impact of this strong hypothesis.
2. We choose to locally constraint the maximal number of incoming and outgoing edges of each node. The threshold for both was set to 5 for this case study. This choice was made to prevent the "flower model effect" (chap. 5 of (van der Aalst, 2011)). A significantly higher or a smaller value of the threshold make the model meaningless. Furthermore, self-loops edges on the nodes are forbidden for the moment as they do not provide much additional information on the process.
3. In the discovered process model, we virtually add a starting and an ending nodes where all traces start and end. It helps for the model description and interpretation.

4. The significance of a cluster is equal to the mean significance of the classes that it contains. It is motivated by the fact that the goal of a cluster is to compact the model and to factorize incoming and outgoing arcs of the included classes.

5. The model is built with several weighting factors and thresholds. The choice of values will shape the model. The study of the individual impact of each possible combination is out the scope of this work, so some parameter value are set empirically here.

3.5.3 The integer linear programming model

So far, we have defined the optimal process discovery problem, an objective function to maximize, a size constraint and a set of modeling hypothesis. Now we provide an *integer linear programming* (ILP) model that can find an optimal solution to this problem.

Let $i \in [\![1,N]\!]$ be the index on the event classes. Let k, $l \in [\![1,K]\!]$ be indexes on the clusters of classes. The event log related parameters are the following: let **T** be a set of events, **L** be a log over T, **C(L)** be the set of event classes of L, **N** be the number of event classes of L, N $= |C(L)|$, $\mathbf{E_{max}}$ be the set of transitions of L, and **PsM(L)** $= (N,E)$ be a process model from L.

Modeling parameters:

- Let $MaxComplexity \in \mathbb{N}$ be the complexity threshold that the model cannot exceed (user-specific).
- Let $p \in [\![1,N]\!]$ be the maximal number of classes allowed in a cluster (user-specific).
- Let $G_p(C(L))$ be the set of the K clusters obtained in the hierarchical structure. The k^{th} element of $G_p(C(L))$ is denoted G_k.
- Let $M_{k,i} = 1(0)$ if the class C_i is in the cluster G_k, $\forall\, i \in [\![1,N]\!]$, $\forall\, k \in [\![1,K]\!]$ (M: matrix of affiliations).
- Let $T_{k,l} = 1(0)$ if $\exists\ C_1 \in G_k$ and $C_2 \in G_l \mid (C_1,C_2) \in E_{max}$, $\forall\ k,l \in [\![1,K]\!]$ (T: matrix of cluster precedence).
- Let $S(i) \in [0,1]$ be the significance of C_i, $\forall\ C_i \in C(L)$. Let $S(k,l) \in [0,1]$ be the significance of the transition from G_k to G_l, $\forall\ k,l \in [\![1,K]\!]$.
- Let $c_{k,l}$ be the number of occurrences of the transition from G_k to G_l, $\forall\ k,l \in [\![1,K]\!]$.
- Let α, β, γ, θ, η, μ, δ and ε be weighting factors so that $\alpha+\beta = 1$, $\gamma+\theta = 1$, $\eta+\mu = 1$, $\delta+\varepsilon = 1$.

Decision variables:

- $w_k = 1(0)$ if cluster G_k is kept (removed) in $PsM(L)$, $\forall\, k \in [\![1,K]\!]$
- $y_{k,l} = 1(0)$ if the edge between G_k and G_l is kept (removed) in $PsM(L)$, $\forall\, k,l \in [\![1,K]\!]$.

ILP formulation:

$$\max_{PsM(L)} \alpha.Conformance(PsM(L)) + \beta.Detail(PsM(L))$$

s.t.

$$w_k \text{ and } y_{k,l} \text{ binary } \forall\, k,l \in [\![1,K]\!] \tag{3.10}$$

$$y_{k,l} \leqslant w_k \;\forall\, k,l \in [\![1,K]\!] \tag{3.11}$$

$$y_{k,l} \leqslant w_l \;\forall\, k,l \in [\![1,K]\!] \tag{3.12}$$

$$y_{k,l} \leqslant T_{k,l} \;\forall\, k,l \in [\![1,K]\!] \tag{3.13}$$

$$\sum_{k=1}^{K} M_{k,i}.w_k \leqslant 1 \;\forall\, i \in [\![1,N]\!] \tag{3.14}$$

$$\sum_{\substack{l=1 \\ l\neq k}}^{K} (y_{k,l} + y_{l,k}) \geqslant w_k \;\forall\, k \in [\![1,K]\!] \tag{3.15}$$

$$(\eta \sum_{k=1}^{K} w_k + \mu \sum_{k=1}^{K}\sum_{l=1}^{K} y_{k,l}) \leqslant MaxComplexity \tag{3.16}$$

$$y_{l,l} \leqslant 0 \;\forall\, k \in [\![1,K]\!] \tag{3.17}$$

$$\sum_{l=1}^{K} y_{k,l} \leqslant 5 \;\forall\, k \in [\![1,K]\!] \tag{3.18}$$

$$\sum_{l=1}^{K} y_{l,k} \leqslant 5 \;\forall\, k \in [\![1,K]\!] \tag{3.19}$$

where

$$Detail(PsM(L)) = \gamma\Big(\frac{\sum_{k=1}^{K} S'(k)w_k}{\sum_{i=1}^{N} S(i)}\Big) + \theta\Big(\frac{\sum_{k=1}^{K}\sum_{l=1}^{K} S(k,l)y_{k,l}}{\sum_{k=1}^{K}\sum_{l=1}^{K} S(k,l)}\Big)$$

and

$$S'(k) = \frac{\sum_{i=1}^{N} M_{k,i}S(i)}{\sum_{i=1}^{N} M_{k,i}} \quad \text{(cluster significance)}$$

The constraint (3.10) ensures that decision variables are binary. (3.11) and (3.12) ensure that an edge is kept only if its two end clusters are kept. (3.13) forbids edges with no occurrence in the log. (3.14) ensures the uniqueness of a class in the model. (3.15) ensures that each cluster that is kept is linked with at least one edge to another node. (3.16) is making certain that the complexity of the model remains under a given threshold. (3.17) forbids self-loop from a node to itself. (3.18) and (3.19) limit the number of incoming and outgoing edges for each node.

3.5.4 Numerical results

Our ILP model was implemented in C++ and solved with IBM Cplex 12.6 (Linux cluster, 8 GB of RAM). The ILP model has 5 input parameters (α, γ, δ, η and *Complexity threshold*). We studied the impact of the complexity threshold and of two weighting factors (α and γ) on the model quality. Conclusions are similar for the other weighting factors. The impact is evaluated on two output measures: the Objective Function (OF) of the ILP and a criteria from the literature that was measured a posteriori (after the optimal model was found). This measure is the trace replayability (RP), also called fitness. RP is a commonly used measure in process mining (van der Aalst, 2011; Rozinat et al., 2008; Gunther and van der Aalst,

2007). It indicates how much of the observed behavior is captured by the process model. RP is the average percentage of each trace's sequence that fits the model. It cannot be incorporated as a part of the objective function because of its non-linearity. In the next section, we propose a new formalism to overcome this limitation. We characterize the properties of a relevant replayability score and we propose several scores that can be used as the objective function to maximize in the optimal process discovery problem.

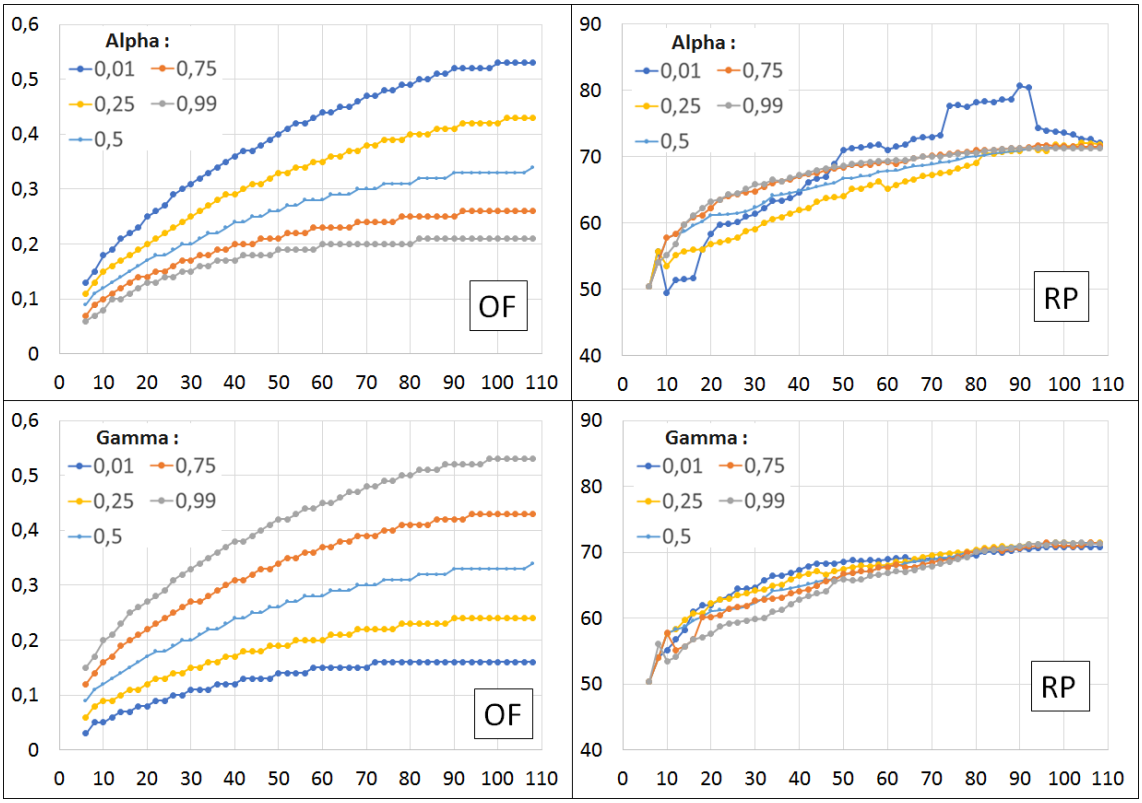


Figure 3.3: RP and OF of the optimal model depending on the model size, for different values of α and γ

Our ILP model was applied to an event log from a real case application in health-care. An extensive description of the real system and of the domain related elements is proposed in Chapter 6 (Case study). Here, we focus on the performance of the model. The log contained 134 possible clusters derived from 59 event classes. The four plots of Figure 3.3 show the increase in OF and RP of the optimal discovered model depending on the maximum size of the model (i.e. the complexity threshold). The first result validates the optimization behavior: the more elements are allowed in the model, the more OF and RP increase (the four plots). For high values of the threshold, both OF and RP finally converge toward maximum values (convergence not reached yet for RP when $\alpha = 0.05$), which means that adding more elements in the model does not provide much improvement after a certain level. The two top plots of Figure 3.3 show OF and RP for 5 values of α between 0.01 and 0.99. For a given complexity threshold, when α increases, OF decreases significantly whereas the RP is more stable. Higher value of α means a higher importance for the $Conformance$ over the $Detail$. The two bottom plots show that OF increases when γ does, whereas RP remains stable. Such experiment demonstrates that the replayability of the model is not highly sensitive to the weighting factors, which guarantees an easier reuse on other data sets.

We did not extend the numerical study of this model because of the limitations discussed below. Instead, we focused our attention on creating a flexible framework to deal with large event logs.

3.5.5 Limitations of the ILP model

The first limitation of the practical use of our ILP program is related to memory usage. The number of possible clusters must remain under 200-300. Furthermore, the execution time has a bell shape on the complexity threshold. When few or a lot of elements are allowed in the model, the decision is faster to make (under 1 second for 15 elements or more than 110, among 134). However, it is more challenging when the number of elements is in the middle (up to 485 seconds for 35 elements). This computation time increases exponentially with the number of clusters.

The second limitation of the approach is methodological and much more concerning. The interpretation of the objective function is fuzzy. It is only partially representative of the log. It only measures global behaviors of the log (e.g. the number of direct transitions and the frequency of included clusters in the model), but it does not consider each trace independently. A lot of information from the log is not taken into account in the model evaluation. Even if the first numerical results tend to show that a classical trace replayability, that was measured a posteriori, increases when the size of the model increases (which was expected beforehand), it does not prove that the model is a good fit of the sequencing of events. Only two by two transitions are well captured in the model, independently from other sequence components. Hence, few traces with hundreds of events will much more contribute to the objective function than smaller traces. Moreover, such global indicators do not consider the complex interactions that may occur within traces' sequence (such as indirect causality among two events, noise, incomplete sequences). An objective function which can integrate direct and indirect transitions, but also capable to skip noisy events in a trace's sequence, is not linearly related to our decision variables (choice of the nodes and arcs of the model). This is why we proposed an alternative and linear substitution.

In conclusion, we can say that the problem of finding an optimal model based on the replayability function is highly combinatorial. In this preliminary work, the objective function is greatly simplified, an Integer Linear Programming model is proposed and standard ILP solvers are used to solve small instances. The complexity makes the ILP approach inapplicable even for problem instances of small size with about 100 different event classes. A tabu search method is now proposed to address problems of larger size and a new replayability function is defined.

3.6 New process model replayability scores

In this section, we first characterize the general properties that a replayability score function should have and we then provide eight relevant replayability score functions to be investigated here. The following notation will be used in this subsection:

- $R(PsM, \sigma)$: replayability score of a trace σ in a process model PsM,
- $C(\sigma)$, $C(PsM)$: the set of event classes of σ and PsM,
- $c \in PsM$: a short-hand notation of $c \in C(PsM)$, i.e., event class c is represented by PsM,
- $\langle c, c' \rangle \in PsM$: a short-hand notation of c and c' being represented by nodes n and n' of PsM and arc $(n, n') \in E(PsM)$, i.e., following relation $\langle c, c' \rangle$ is represented by PsM,
- $C(PsM, c)$: the set of event classes of a node n such that $c \in C(n)$
- $\sigma \in PsM$: a shorthand notation indicating the full replayability of σ in PsM, i.e., $(\sigma = \langle c_1, c_2, ..., c_n \rangle, c_i \in PsM, \langle c_i, c_{i+1} \rangle \in PsM, \forall i)$.

3.6.1 Properties of the replayability score function

The following defines some regularity properties of the replayability score function, i.e., better replayability for richer process models.

Definition 9. (Regularity) A replayability score function is said regular if assumptions $A1$-$A5$ hold. Otherwise it is said irregular.

A1 - Perfect replayability:

$$(\sigma = \langle c_1, c_2, ..., c_n \rangle, \sigma \in PsM, ||C(PsM, c_i)|| = 1) \Rightarrow R(PsM, \sigma) = 1$$

A2 - Null replayability:

$$(C(\sigma) \cap C(PsM) = \varnothing) \Rightarrow R(PsM, \sigma) = 0$$

A3 - Preference of better event representation:

$$(\sigma = \langle \sigma_1, c, \sigma_2 \rangle, \sigma' = \langle \sigma_1, c', \sigma_2 \rangle,\ c \in PsM,\ c' \notin PsM) \Rightarrow R(PsM, \sigma) \geq R(PsM, \sigma')$$

A4 - Preference of better (direct or indirect) following relation representation:

$$\left(\begin{array}{c} \sigma = \langle c_1, c_2, ..., c_n \rangle \\ PsM = (N, E), PsM' = (N, E') \\ \langle c_i, c_j \rangle \in E' \Rightarrow \langle c_i, c_j \rangle \in E, \forall j > i \end{array} \right) \Rightarrow R(PsM, \sigma) \geq R(PsM', \sigma)$$

A5 - Preference for detailed process model

$$\left(\begin{array}{c} \sigma = \langle c_1, c_2, ..., c_n \rangle \\ c_i \in PsM' \Leftrightarrow c_i \in PsM \\ \langle c_i, c_j \rangle \in PsM' \Leftrightarrow \langle c_i, c_j \rangle \in PsM, \forall j > i \\ ||C(PsM, c_i)|| \leq ||C(PsM', c_i)|| \end{array} \right) \Rightarrow R(PsM, \sigma) \geq R(PsM', \sigma)$$

3.6.2 New replayability score functions

In the following, we introduce 8 replayability score functions that capture properties of a desirable replayability score function at different degrees (see Table 3.1).

$$R^1(PsM, \sigma) = 1(\langle c_1, c_2, ..., c_n \rangle \in PsM)$$

$$R^2(PsM, \sigma) = \frac{1}{n} max\{i : \langle c_1, c_2, ..., c_i \rangle \in PsM\}$$

$$R^3(PsM, \sigma) = \frac{1}{n} max\{j : \exists k, \forall i \leq k, c_i \notin PsM, \langle c_{k+1}, c_{k+2}, ..., c_{k+j} \rangle \in PsM\}$$

where $1(x)$ is a binary variable equal to 1 if $x = TRUE$. $R^1(PsM, \sigma)$ is a binary measure of whether the trace σ is fully represented by PsM. $R^2(PsM, \sigma)$ measures the percentage of events in the sub-trace starting from the beginning represented by PsM. $R^3(PsM, \sigma)$ measures the percentage of events in the first sub-trace represented by PsM.

Table 3.1: 8 replayability functions with different properties

Function index	Feature description
R^1	Strict replayability
R^2	Head partial replayability
R^3	Partial replayability
R^4	Γ game replayability
R^5	Γ game replayability with details
R^{4a}	Sub-trace replayability
R^{5a}	Sub-trace replayability with details
R^6	Forced transition replayability

We then introduce a **replayability game** Γ and define two other score functions. The replayability game Γ starts from the first event $c_{[m]}$, called **m-th event played**, with $m = 1$ of trace σ represented by PsM. We then seek the next event c_i of trace σ represented by PsM. If the transition $\langle c_{[m]}, c_i \rangle$, called **transition attempted**, exists in PsM, then $c_{[m+1]} = c_i, m = m+1$ and repeat above from $c_{[m]}$. Otherwise, we repeat above from c_{i+1}. $L(PsM,\sigma)$ denotes **the number of events played**, i.e., $L(PsM,\sigma) = m$, $\eta(PsM,\sigma)$ **the number of abandoned attempted transitions**, and $\delta(PsM,\sigma) = 1([m] \neq m)$, called **event skipping indicator**, is a binary variable equal to 0 if no event is skipped before the last event $c_{[m]}$.

$$R^4(PsM,\sigma) = \left(\frac{L(PsM,\sigma)}{n} - \alpha\delta(PsM,\sigma) \right)^+$$

$$R^5(PsM,\sigma) = \left(\frac{1}{n} \sum_{i=1}^{n} \sum_{n \in N} \frac{1}{||C(n)||} x_{i,n} - \alpha\delta(PsM,\sigma) - \beta \frac{\eta(PsM,\sigma)}{n} \right)^+$$

where $x_{i,n}$ is binary and equal to 1 if event c_i is played in PsM at node n. Convention $0/0 = 0$ is used.

$R^4(PsM,\sigma)$ measures the percentage of events played modified by an event skipping penalty. $R^5(PsM,\sigma)$ combines a detail-level dependent event replayability measure and penalties of event skipping and abandoned attempted transitions of the Γ game. Score functions R^4 and R^5 are also extended to all playable sub-traces.

$$R^{4a}(PsM,\sigma) = \max_{s \in PsM(\sigma)} \left(\frac{||s||}{n} - \alpha\delta(\sigma,s) \right)$$

$$R^{5a}(PsM,\sigma) = \max_{s = \langle c_{[1]},\ldots,c_{[m]} \rangle \in PsM(\sigma)} \left(\frac{1}{n} \sum_{i=1}^{m} \frac{1}{||C([i])||} - \alpha\delta(\sigma,s) - \beta \frac{\eta(PsM,\sigma,s)}{n} \right)$$

where $PsM(\sigma)$ is the set of sub-traces $s = \langle c_{[1]}, c_{[2]}, \ldots, c_{[m]} \rangle$ of trace σ such that $s \in PsM$, including the empty sub-trace ε; $\delta(\sigma,s) = 1([m] \neq m)$; $\eta(PsM,\sigma,s) = \sum_{i=[1]}^{[m]} 1(c_i \in PsM) - m$.

We now introduce a new **replayability game** and a last score function. It plays exactly the sub-trace $s^* = \langle c_{[1]}, c_{[2]}, ..., c_{[m]} \rangle$ of all events represented by the process model PsM, i.e., s^* is such that $[1] = min(i : c_i \in PsM), [j+1] = min(i > [j] : c_i \in PsM)$. A transition $\langle c_{[i]}, c_{[i+1]} \rangle$ is said a **forced transition** if $\langle c_{[i]}, c_{[i+1]} \rangle \notin PsM$.

$$R^6(PsM, \sigma) = \left(\frac{1}{n} \sum_{i=1}^{n} \sum_{n \in N} \frac{1}{||C(n)||} z_{i,n} - \alpha \delta(\sigma, s^*) - \beta \frac{\phi(PsM, \sigma, s^*)}{n} \right)^+$$

where $z_{i,n}$ is a binary variable equal to 1 if event class c_i is represented by node n, $\phi(PsM, \sigma, s^*)$ is the number of forced transitions.

These replayability scores have the following relations.

Property 1. $R^i(PsM, \sigma)$ with $i = 1, 2, 3, 4a$ and $5a$ are regular replayability score functions, whereas $R^i(PsM, \sigma)$ with $i = 4, 5$ and 6 are irregular.

Proof. First, assumptions A1, A2 and A5 hold for all replayability score functions. A3 and A4 trivially hold for $R^i(PsM, \sigma)$ with $i = 1, 2, 3$. A4 clearly holds for R^{4a} and R^{5a} as $PsM'(\sigma) \in PsM(\sigma)$. A3 holds for R^{4a} as $PsM'(\sigma) \in PsM(\sigma)$ and $\delta(\sigma, s) = \delta(\sigma', s)$. A3 holds for R^{5a} as $PsM'(\sigma) \in PsM(\sigma)$ and $\eta(PsM, \sigma, s) \in \eta(PsM', \sigma, s), \forall s \in PsM(\sigma')$. It can also be checked that A4 holds for R^6.

The irregularity of $R^i(PsM, \sigma)$ with $i = 4, 5, 6$ is proven by counter-examples with $PsM = (N, E)$ for which $R^5(PsM, \sigma) = R^4(PsM, \sigma)$ if $\beta = 0$. Counter-example of $A3$ for R^4: $\sigma = ABCDEF$, $\sigma' = ABGDEF$, $N = \{A, B, C, D, E, F\}$, $E = \{(A,B), (B,C), (B,D), (D,E), (E,F)\}$, $\alpha = 0.1$, $R^4(PsM, \sigma) = 0.5$, $R4(PsM, \sigma') = 5/6 - \alpha$. Counter-example of $A4$ for R^4: $E' = E - (B,C)$, $\alpha = 0.1$, $R^4(PsM, \sigma) = 0.5$, $R^4(PsM', \sigma) = 5/6 - \alpha$. Counter-example of $A3$ for R^6: $R^6(PsM, \sigma) = (1 - 1/6\beta)^+ < R^6(PsM, \sigma') = 5/6 - \alpha$ if $\beta = 2$ and $\alpha = 0$. □

Property 2.
(a) $R^1(PsM, \sigma) = 1 \Rightarrow R^i(PsM, \sigma) = 1 \; \forall i = 2, 3, 4$;
(b) $R^2(PsM, \sigma) = R^3(PsM, \sigma)$ or $R^2(PsM, \sigma) = 0$;
(c) $n^{-1} L(PsM, \sigma) \geq R^3(PsM, \sigma)$, $R^4(PsM, \sigma) \geq R^3(PsM, \sigma) - \alpha$;
(d) $R^4(PsM, \sigma) \geq R^2(PsM, \sigma)$ if $\alpha \leq n^{-1}$;
(e) $R^{4a}(PsM, \sigma) \geq R^4(PsM, \sigma)$, $R^{5a}(PsM, \sigma) \geq R^5(PsM, \sigma)$

Proof. Trivial. □

Property 3. $R^6(PsM, \sigma)$ is regular if $(\alpha n + 2\beta) \leq \chi$ where $\chi \stackrel{\Delta}{=} \min_{g \in S} ||g||^{-1}$.

Proof. From the proof of Property 1, only A3 needs to be checked. As $\sigma = \langle \sigma_1, c, \sigma_2 \rangle, \sigma' = \langle \sigma_1, c', \sigma_2 \rangle, c \in C(PsM), c' \notin C(PsM)$, then $R^6(PsM, \sigma) - R^6(PsM, \sigma') \geq \frac{1}{n}\chi - \alpha - \frac{1}{n} 2\beta \geq 0$. □

In the remaining of this chapter, we assume $\alpha = 0.5n^{-1}$ and $\beta = n^{-1}$. Table 3.2 summarizes the replayability scores of seven traces with respect to the process model of Figure 3.4. For the strict replayability, traces 1 and 2 score 1 and others score 0. Head partial replayability allows finer ranking of the replayability but depends only on the head sub-trace played. Partial replayability improves it by allowing the first event to be not played. Γ game replayability enriches the previous scores by allowing event skips. The mixed replayability and Γ game replayability with details further enrich the scores by taking into account detail-level of the process model.

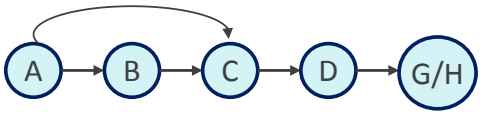


Figure 3.4: Example of a process model with 5 nodes and 5 arcs

Table 3.2: Example of replayability values of the process model of Figure 3.4 on 7 traces

Trace	R^1	R^2	R^3	R^4	R^5	R^{4a}	R^{5a}	R^6
1. A-B-C-D-G	1	1	1	1	0.90	1	0.90	0.90
2. B-C-D	1	1	1	1	1	1	1	1
3. B-C-D-E-F	0	0.60	0.60	0.60	0.60	0.60	0.60	0.60
4. A-B-C-D-E	0	0.80	0.80	0.80	0.80	0.80	0.80	0.80
5. E-A-B-F-C	0	0	0.40	0.50	0.50	0.50	0.50	0.50
6. A-B-D	0	0.67	0.67	0.67	0.56	0.67	0.56	0.89
7. A-B-G	0	0.67	0.67	0.67	0.56	0.67	0.56	0.72
Mean value	***0.29***	***0.68***	***0.73***	***0.75***	***0.70***	***0.75***	***0.70***	***0.79***

3.6.3 Properties of optimal solutions

This subsection establishes the properties of optimal process models depending on replayability score functions.

Theorem 1. *Assume that $\sigma \in PsM$ implies $R(PsM,\sigma) = 1$. If $C \in S$, then the process model PsM^* with a single cluster C and a self-loop arc is optimum and $R(PsM^*,L) = 1$.*

Proof. Trivial as all traces $\sigma \in PsM^*$. □

Remark 10. Replayability score functions R^1, R^2, R^3, R^4 and R^{4a} meet the conditions of Theorem 1 and hence have trivial optimal process model with only the most aggregated clusters. As a result, **taking into account process model details is crucial in process mining optimization and only details-dependent score functions (R^5, R^{5a}, R^6) will be considered in our numerical experiments**.

Theorem 2. *Assume that $PsM'(\sigma) \subseteq PsM(\sigma)$ implies $R(PsM,\sigma) \geq R(PsM',\sigma)$. If $C \notin S$, then there exists an optimal process model PsM^* that contains only root clusters of event class hierarchical trees.*

Proof. For any process model PsM, merge into the same node all clusters of PsM belonging to the same hierarchical tree, and assign to the merged node the root cluster lead to a process model PsM^* such that $PsM(\sigma) \subseteq PsM^*(\sigma), \forall\sigma$. As a result, $R(PsM^*,\sigma) \geq R(PsM,\sigma)$ which concludes the proof. □

Theorem 3. *Under A4, $R(PsM',\sigma) \geq R(PsM,\sigma)$ where $PsM = (N,E)$, $PsM' = (N,E')$ and $E \subseteq E'$.*

Proof. Trivial from A4. □

Remark 11. The above theorem implies that the complexity bound is always reached in optimal process models.

3.7 Optimization of process discovery

This section proposes a tabu search algorithm to solve the process mining optimization problem presented in Section 3.4. We first give an overview of the algorithm and then the details of its components.

3.7.1 Overview of the tabu search

A tabu search is a local search method that avoids being stuck in local optimum by allowing non-improving solutions (Glover, 1986). As a local search, it takes a feasible solution of the problem and checks for other similar solutions (the neighbors) to find an improving solution. The idea for escaping local optima is to record information about already visited solutions and to introduce two mechanisms to prevent from looping to them. First, at each step of the neighbors search, a deteriorating solution can be chosen if no improving solution is found. Second, the method forbids to go back to previously visited solutions ("tabu solutions") to avoid loops, even if it would improve the current solution. It is done by recording features of visited solutions in a tabu list. Our tabu search algorithm can be summarized as follows (Algorithm 1).

Algorithm 1 Tabu search algorithm for process mining

Step 1 – Initialization
1.1 Select an initial solution: s_0
1.2 Update the current solution: $s \leftarrow s_0$
1.3 Update the best known solution: $s^{best} \leftarrow s_0$
1.4 Initialize the tabu list as empty: $TL \leftarrow \{\}$
Step 2 – Explore the neighborhood of the current solution
2.1 Generate a set of non tabu neighbor solutions;
2.2 Evaluate all solutions and determine the best: $s^*_{neighbor}$
2.3 Update the current solution: $s \leftarrow s^*_{neighbor}$
2.4 If $(s^*_{neighbor} > s^{best})$,
Update the best known solution: $s^{best} \leftarrow s^*_{neighbor}$
2.5 Update tabu list: $TL \leftarrow TL + \{s^*_{neighbor}\}$
Step 3 – Repeat step 2 until a stopping criterion is reached
Step 4 – Evaluate the final solution s^{best}

A fixed size Q tabu list is used and updated on a First In First Out basis. As a result, whenever a solution s is accepted at iteration t, its features are added to the tabu list and features of the oldest solution identified at iteration $t - Q$ are removed.

Two local moves are considered in our approach: **MoveArc** by replacing an arc $\langle n, n' \rangle$ by another arc and **MoveNode** by replacing a node n by another node. As each node of the process model corresponds to an event cluster, let $cl(n)$ be the cluster of node n. The following features are recorded: (cl, cl') with $cl = cl(n)$ and $cl' = cl(n')$ if the MoveArc is used and cl if MoveNode is used. The features (cl, cl') forbid all arcs (cl, cl'). The feature cl forbids nodes n with $cl(n) = cl$.

The stopping criterion is simply a given number of algorithm iterations. At Step 4, the final solution s^{best} is evaluated exactly using the relation 3.2 (Section 3.4) with respect to all traces of the event log. To speed up the tabu search, in step 2.2, the selected neighbor solutions are evaluated using the faster Monte-Carlo approach of relation 3.8 (Section 3.4) with independent random samples for different solutions. In the present work, the sample size K (number of sampled traces) is selected to ensure that the 95% confidence interval is smaller than a given threshold. From Remark 5, the confidence interval half width CI_{HW} has the

following properties:

$$CI_{HW} = t_{(K-1,1-0.5\alpha)}\sqrt{\frac{S(K)^2}{K}} \leq \frac{t_{(K-1,1-0.5\alpha)}}{\sqrt{K}}$$

where $S(K)^2$ is the standard variance estimate and $1-\alpha = 95\%$. The upper bound of the confidence interval can be refined by the following:

$$\sqrt{S(K)^2} \leq max(1-E[R(PsM,\sigma)], E[R(PsM,\sigma)])$$

3.7.2 Initial solution

Two approaches were used to generate an initial solution. The first is to generate a random model. It requires two parameters: the maximum number of elements in the model and the initial number of nodes. Each node of the model is randomly assigned an event cluster. Then, as many arcs as needed to respect the total number of elements in the model are randomly added between the nodes (only arcs with at least one occurrence in the log are considered). The resulting model often has a very low replayability value, especially when the number of possible clusters is large.

The second initial solution is found by the Integer Linear Programming model of our preliminary work (Section 3.5). The obtained model was proven optimal with regard to an aggregated quality criterion based on direct flowing relations represented. Although the replayability function described here is different from this criterion, ILP approach still provides a good initial solution. It significantly reduces the computation time of the tabu search compared to the random initial solution. With the randomly generated initial solution, many iterations are needed to go from the initial poor quality solution to a promising area of the space search.

3.7.3 Local moves

Our tabu search algorithm uses the two different moves to generate promising neighbors from a current solution.

1. **MoveArc**: it replaces an existing arc of a model by a new arc
2. **MoveNode**: it replaces the cluster of a node by a new cluster.

Both moves work similarly: as the total number of nodes and edges in the model is bounded, both moves first remove a part of the current model and then add new elements after. To decide which arcs and nodes to remove (or to add) without evaluating all the possible cases, we define an "artificial" **performance measure** equal to the number of occurrences of the elements in the log. This performance needs to be measured only once, when the log is initially read. The idea behind this measure is to identify very quickly, i.e. with no need for extra computation, the most promising arcs and nodes. As a result, clusters can be sorted from "best" (highest artificial performance) to "worst" (lowest artificial performance). Pairs of clusters (arcs) can also be sorted from "best" to "worst".

MoveArc is a two-step local move to generate X neighbor solutions: first it removes the worst arc of the model. It then replaces it with some better arcs, i.e. arcs with the highest artificial criterion. The latter is done from the list of arcs sorted in decreasing order of their artificial performance measure. The X "best" non tabu arcs of the list are considered leading to X non tabu neighbor solutions

MoveNode is a four-step local move to generate X' neighbor solutions. First, the "worst" node n of the model and all its arcs are removed. Second, each incoming arc (n',n) of the removed node n is replaced by

the non tabu arc $(n', n")$ between remaining nodes with the highest "artificial" performance if such an arc exists. Third, one of the X' "best" non tabu clusters is assigned to the node n. Fourth, till complexity limit is reached, the "best" incoming $(n", n)$ or outgoing arcs $(n, n")$ of the new node n are added, selected from the list of non tabu arcs sorted from "best" to "worst". These 4 steps are illustrated in Figure 3.5.

Moves *MoveArc* and *MoveNode* generate respectively X and X' neighbors for any given model. Step 2.1 of our tabu search algorithm merges the two neighborhoods to create the final set of $X + X'$ neighbor solutions. The two moves could also be used alternately instead of jointly, but the search would be forced to use each type of move every two iterations. Instead, it may turn out to be more relevant to replace several clusters in a row, followed by several arc replacements in a row. It depends on the search space and the current model. The replayability of the $X + X'$ newly created neighboring models is evaluated and the best model is chosen as the new current solution of the search. The methodology of using a pre-computed performance measure saves the effort of evaluating the replayability of each possible new model obtained by replacing any node or by replacing any Cluster (e.g. thousands of possible models). Instead, only $X + X'$ promising models are evaluated (e.g. tens of models).

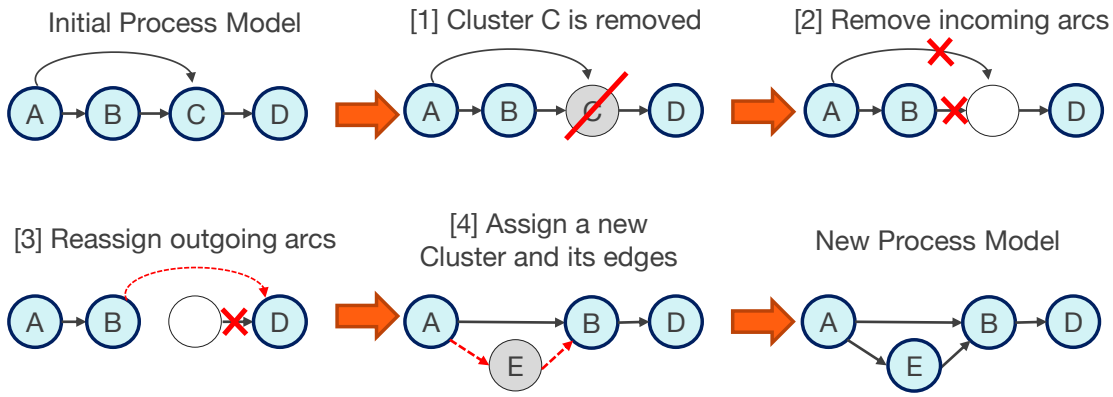


Figure 3.5: Illustration of *MoveNode*: 4 steps to create a neighboring model

3.7.4 Summary

Global methodology and main contributions of this chapter are presented in Figure 3.6. From an event log, we create the hierarchical structure for Clusters of event classes as presented in Section 3.4.2, using aggregation methods based on significance value or experts' knowledge (step 1). It leads us to a set of traces that is used to create an initial solution (i.e., a process model). Such model may be created randomly or through integer linear programming, as described in Section 3.7.2 (step 2). Then we execute the tabu search: new process models are computed using two proposed moves (*MoveArc* and *MoveNode*, see Section 3.7.3), resulting in a set of neighbors to evaluate (step 3). The replayability is evaluated using an original method based on a Monte-Carlo sampling strategy, allowing to decrease the computational complexity (step 4). The search continues following the tabu algorithm until the stopping condition is reached.

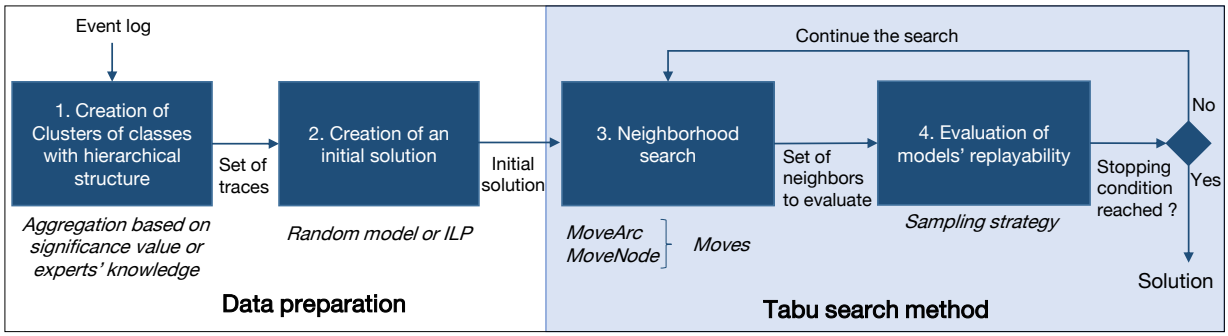


Figure 3.6: Global methodology of our approach to solve the optimal process discovery problem

3.8 Computational experiments

This section reports the results of computational experiments for comparison of our tabu search algorithm with the commercial software DISCO developed by Fluxicon (version 1.9.1) in terms of the replayability scores defined in this chapter under various process model complexity constraint using the same Causal Net notation without duplication. A random process model generation approach is also considered to serve as basis to evaluate the benefit of optimal process mining. A sensitivity analysis is performed to set the parameters of the tabu search algorithm. The different approaches are tested on a real-life case study and event logs randomly generated by a log generator to be presented. All experiments described in this section were performed on a PC with an Intel Core i7 processor (2.1 GHz), 4GB RAM and Linux OS. The tabu search algorithm was coded in C++.

3.8.1 Log generation

A log generator enables to test process mining algorithms on logs of different complexity. A log complexity can be defined in various ways, such as the total number of traces, of events, of different traces (variants) or of event classes. Here, we focus on attributes highly correlated with spaghetti-like models, so we define **a log complexity as the number of event classes** it contains. It is easier to discover a compact process model from a log with few event classes and many traces than the opposite because the total number of nodes in a model can never exceed the number of classes in the log. Figure 3.7 illustrates two process models discovered from two separate logs. Both models have a replayability score of 1. Even with four times fewer traces, the process model of Figure 3.7-b is more complex than the model 3.7-a because the original log had twice as many classes.

In our experimental design, we generated 5 types of logs with increasing complexity, ranging from 15 to 130. Note that most existing studies consider event logs with less than 30-40 classes (also named activities or tasks) (van der Aalst, 2004; van der Aalst et al., 2005; Mendling et al., 2007; Rozinat et al., 2008; Rozinat and van der Aalst, 2008; Huang and Kumar, 2012; Rebuge and Ferreira, 2012; van der Werf et al., 2008) and only a few try to deal with more than 100 classes (Gunther and van der Aalst, 2007; Prodel et al., 2015).

Logs are generated following three steps. We first randomly create a process model *PsM* with a given size (number of nodes and arcs). Then, we generate traces that perfectly match this model. Finally, we add noisy events to the traces. Input parameters are *the size* of *PsM*, the number of possible *event classes*, the number of *traces* and the *percentage Z of noise*. The only constraint is to have a number of classes greater than the number of nodes in the process model. The 3 steps are described in detail below.

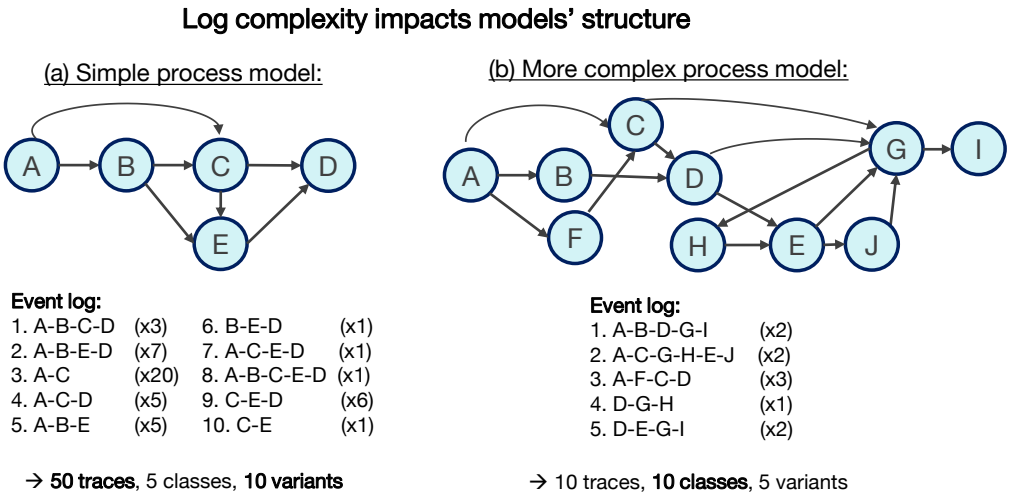


Figure 3.7: Process models complexity versus event log complexity

- **Step 1:** A process model is created by randomly assigning one cluster to each node. Any arc is assumed possible. Arcs are recursively added by randomly assigning one outgoing arc to each node until the size threshold is reached. In the end, nodes almost have the same numbers of arcs. This assignment procedure ensures a connected graph. See Figure 3.8 for an example of a generated model of size 26.

- **Step 2:** Each trace σ is generated by randomly choosing a starting node in the model. The event class in the node is the first element of σ's sequence. Then, we compute the probability for σ to have one more event class in its sequence. This probability is a decreasing function of σ's current length n_σ: $p_\sigma(\textit{one more event}) = a \times n_\sigma^2 + b$, where $a = -0.001$ and $b = 1$. We then randomly choose an arc of *PsM* which goes from the current node to a new one. The class in the new node is added to σ's sequence. This procedure is repeated until no more event is needed and until we reach the desired number of traces. The set of all the created traces is denoted L. Finally, we have $R^i(PsM, \sigma) = 1$ with $\sigma \in L$ and $i \in \{5, 5a, 6\}$.

- **Step 3:** We add noise in traces by randomly adding event classes in their sequence. Noisy event classes are chosen among classes that are not assigned in *PsM*. The number of added noisy classes is $\left[Z \times \sum_{\sigma \in L} n_\sigma \right]$.

Five groups of event logs are generated with process model of complexity (15, 30, 50, 75, 130) with noise coefficient equal to (5%, 10%, 20%, 20%, 20%) and node-arc complexity of (5-10, 25-50, 30-60, 40-80, 40-80). 20 logs of 100 000 traces are generated for each group. As a result, 100 logs were obtained.

3.8.2 Preliminary analysis of the tabu search

The tabu search parameters of Table 3.3 are used throughout the thesis. The confidence interval is used to set the number of Mont Carlo samples such that the length of the 90% confidence interval is at most 0.01. This section focuses on analyzing the impact of the number of tabu search iterations and the size of the process models.

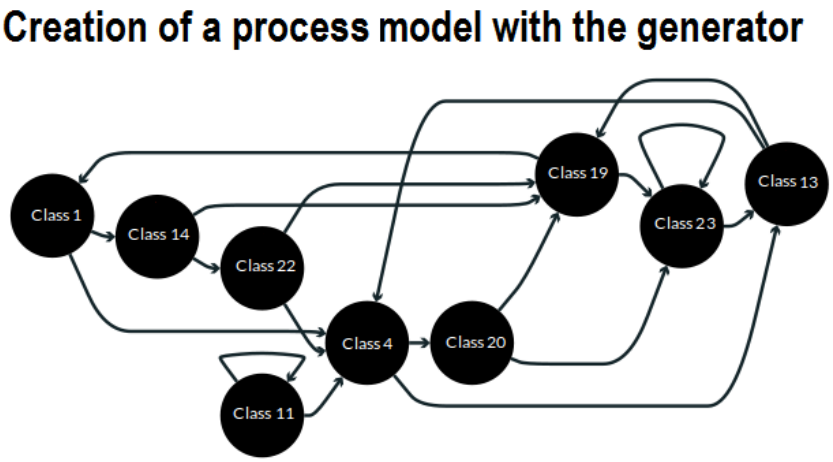


Figure 3.8: A process model of size 26 created by the log generator

Table 3.3: Tabu search parameters

Parameter name	Value
Size of tabu list	15
Number of MoveArc per iteration	5
Number of MoveNode per iteration	5
Confidence interval	0.01
Number of tabu search iterations	[1-500]

The tabu search is evaluated by computing the average replayability of the best found model over several replications. For a given log, 20 replications are needed to ensure significant results, due to randomly created initial solutions. Only replayability score functions R^5 and R^6 are considered. Score R^{5a} is an extension of R^5 and behaves similarly to R^5. Mean replayability score and 90% confidence interval are shown in Figure 3.9.

The following observations are made. The **size of the model** to mine has a clear impact on the best found solution's replayability score. Figure 3.9-a shows the replayability value of the best solution found by our tabu search algorithm versus the size of the discovered model. Though proved to be irregular in Theorem 1, the R^6 curve suggests that the optimal R^6 score is practically regular: having more elements in the model implies a higher replayability. There is a fast increase in replayability for small to medium size between 30 and 60, then the increase slows down for sizes between 60 and 80 and becomes almost null for sizes greater than 100 elements. This convergence is explained by the fact that adding more elements in the model does not provide much improvement after a certain level. The shape of the R^5 curve illustrates that the optimal R^5 score is not regular as models with more elements are proven to have a slightly lower score.

The second result on replayability versus the **number of algorithm iterations** is important to set the stopping criterion of tabu search. More iterations give the opportunity to improve the best solution by exploring more models but is costly in computation time. Impact of the number of iterations on the best found solution's replayability score is shown on Figure 3.9-b. Values ranging from 5 to 500 iterations were tested, but only the range from 5 to 200 is displayed as no further improvement was observed. Test data is still a set of 10 logs with a complexity of 100 classes. Under this parameter setting, the graph shows that

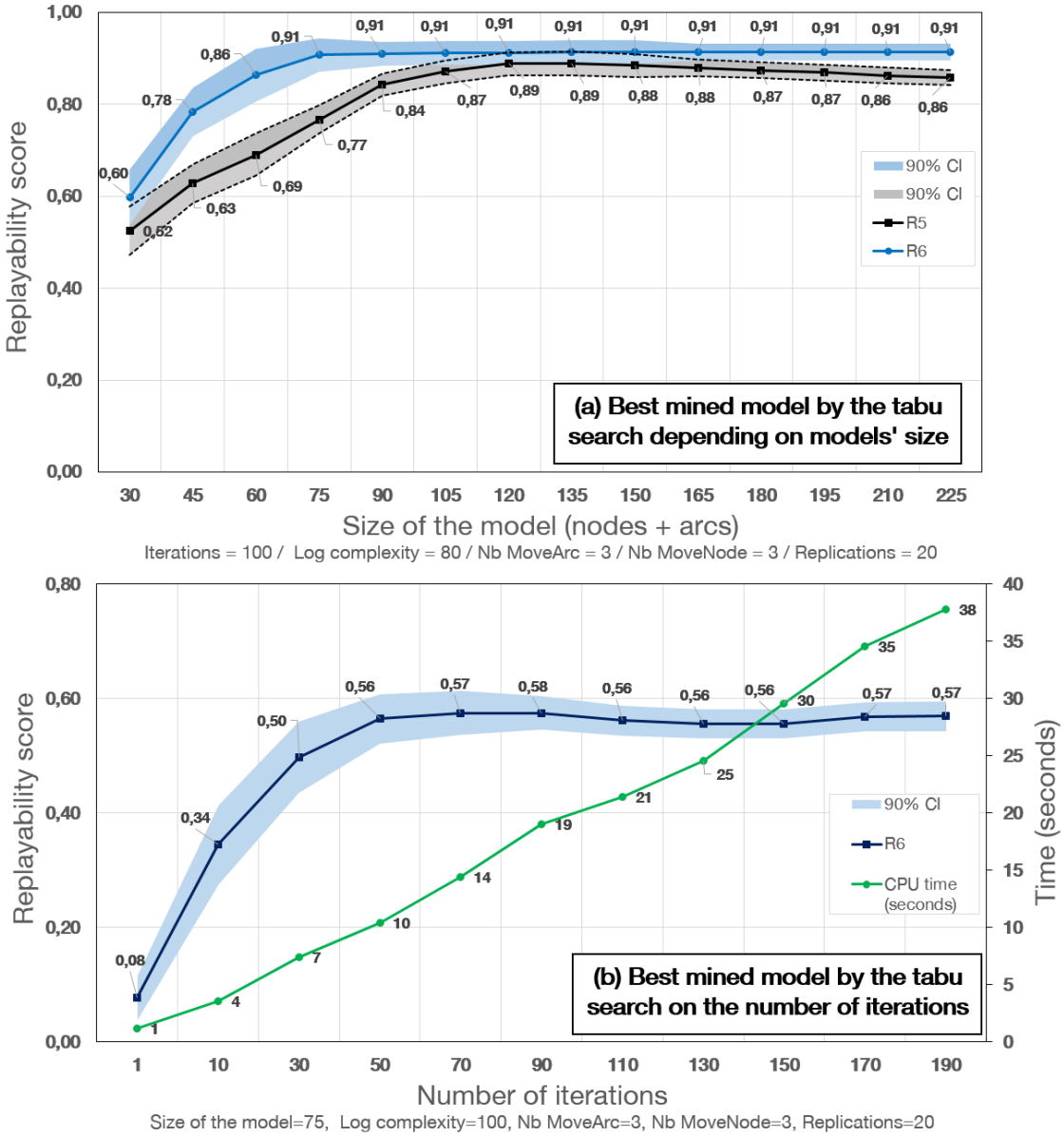


Figure 3.9: Replayability versus size of the mined model and number of iterations

no significant improvement is possible beyond 90 iterations whereas it linearly increases the computation time. In the remaining of this chapter, for any parameter configuration, the number of iterations is 200.

3.8.3 Comparison with the commercial software DISCO

This section presents a comparison between our approach and the commercial software DISCO. DISCO is an enhanced version of the fuzzy miner presented in (Gunther, 2009). It is suitable for process discovery on real-life logs with noise, flexible traces, lots of variants and lots of traces. Discovered models are created as Causal Nets.

In this Section, we choose to use a random model as an initial solution for the tabu search. This choice is motivated by the fact that DISCO cannot take advantage of results from the ILP model (Prodel et al., 2015): it would create an unfair comparison bias to use an initial solution generated by the ILP model.

The comparison was made on 5 sets of 10 logs each, each set corresponding to a given complexity (i.e., number of event classes in a log). For each log, we built the best possible process model for 5 different sizes. Ranges of possible sizes depend on the log complexity as the size (number of nodes) is necessarily lower than the total number of classes in the log. We also computed results of a random approach where nodes and arcs were assigned randomly. Results of DISCO, the tabu search and the random approach are shown in Table 3.4. Column "AVG" is the average replayability of the best mined models over the 200 runs (10 logs times 20 independent replications). Column "STD" gives the average standard deviation over the same runs.

Table 3.4 shows that our approach performs better than DISCO for small and middle complexity (15, 30 and 50 Classes). Our tabu search performs similarly as DISCO for higher complexity (75 and 130 Classes). It is important to notice how both approaches perform very well on various types of logs. It confirms their capability to deal with flexible real logs. Both methods also significantly outperform the random approach which shows the important of optimal process mining.

Table 3.4: Replayability function R^6 of DISCO, a random approach and our tabu search

LC	Size	DISCO		Random			Tabu search		
		AVG	STD	AVG	STD	DIFF	AVG	STD	DIFF
15	5 / 10	48.86	10.12	31.28	2.79	-35.98%	68.09	6.35	39.36%
	8 / 16	78.71	5.05	49.11	2.19	-37.61%	83.79	3.27	6.45%
	11 / 22	89.42	2.39	70.69	4.09	-20.95%	89.97	2.37	0.61%
	14 / 28	94.80	1.90	92.81	1.06	-2.10%	97.01	0.24	2.33%
30	10 / 20	53.39	7.13	24.66	1.20	-53.80%	60.37	8.33	13.07%
	15 / 30	72.79	6.26	35.58	1.07	-51.11%	76.79	7.49	5.49%
	20 / 40	83.67	4.04	45.83	1.08	-45.23%	87.15	5.68	4.16%
	25 / 50	90.15	1.60	55.31	0.87	-38.65%	94.18	5.05	4.47%
	30 / 60	92.19	1.11	65.08	0.31	-29.40%	98.18	5.45	6.50%
50	10 / 20	44.44	6.76	15.27	0.70	-65.65%	52.95	5.86	19.15%
	15 / 30	61.96	5.46	21.04	0.86	-66.04%	70.15	3.81	13.22%
	20 / 40	73.54	3.26	26.5	0.87	-63.96%	80.40	1.28	9.33%
	30 / 60	79.79	1.38	37.27	1.01	-53.28%	84.73	0.35	6.19%
	40 / 80	81.64	0.97	47.37	0.85	-41.97%	86.84	0.30	6.37%
	50 / 100	84.21	0.47	57.59	0.24	-31.61%	87.49	0.32	3.89%
75	10 / 20	49.96	7.60	10.79	1.54	-78.41%	50.11	4.28	0.30%
	20 / 40	74.21	4.20	18.66	1.07	-74.85%	74.85	2.41	0.86%
	30 / 60	81.77	1.28	25.92	1.52	-68.30%	81.99	0.42	0.26%
	45 / 90	83.01	1.24	35.81	0.71	-56.86%	83.42	0.38	0.48%
	60 / 120	84.33	1.06	46.17	1.26	-45.25%	85.18	0.31	1.01%
	70 / 140	85.44	0.48	52.61	1.05	-38.43%	86.39	0.22	1.11%
130	10 / 20	31.83	7.09	6.55	1.12	-79.44%	40.15	6.08	26.14%
	20 / 40	56.81	5.80	10.81	0.44	-80.98%	65.77	4.64	15.77%
	35 / 70	77.91	2.50	17.51	0.91	-77.53%	79.90	2.29	2.55%
	50 / 100	81.57	1.62	23.64	1.39	-71.02%	82.98	1.50	1.73%
	80 / 160	82.96	0.80	35.91	1.18	-56.71%	85.25	0.78	2.76%
	110 / 220	84.35	0.87	47.31	0.93	-43.91%	85.8	0.76	1.72%
	130 / 260	86.02	0.34	55.06	0.15	-36.00%	86.5	0.41	0.56%

A graphical comparison of DISCO and our tabu search is also displayed in Figures 3.10 and 3.11 respectively. 3 models for each approach with respectively 10, 25 and 50 nodes are displayed. Both methods are suitable to deal with models in a range from 1 to 50 nodes without suffering from the spaghetti curse. To the best of our knowledge, DISCO is the only process mining algorithm capable of generating a process model which is not "spaghetti-like".

3.9 Conclusion and future research

In this chapter, we proposed a new methodology to compute process models from complex event logs. The scientific contribution is multiple: (1) the first rigorous mathematical formulation of the optimal process mining problem; (2) a hierarchical representation of event relations; (3) an event-log-size independent Monte-Carlo simulation approach; (4) properties of optimal process models; (5) an efficient tabu search algorithm for process model optimization. The method was tested on a wide range of both generated and real event logs. It has proven to perform well in terms of convergence and computation time. The proposed method outperforms both random process creation strategies and state-of-the-art process mining algorithm, the Fuzzy Miner heuristic implemented in commercial software DISCO. A realistic case study is presented in Chapter 6. It also provides qualitative results. The validity of our models has been confirmed by health practitioners. Finally, we proposed an innovative rigorous mathematical framework which can be used to build and compare solutions using objective criteria.

For future works, we intend to improve process model computation by taking into account domain specific parameters and by adding weights on Clusters, classes and/or arcs. Based on experts' knowledge, it may be possible to converge quickly to realistic process model. A more in-depth study of the relationship between the replayability scores and the event log information captured by the resulting model is highly needed. We also intend to automatically convert Causal Net process models into executable models, by creating notation equivalences with other formalism such as Petri nets or state-charts, so that it can be directly injected in a simulation model. This topic is addressed in Chapter 5.

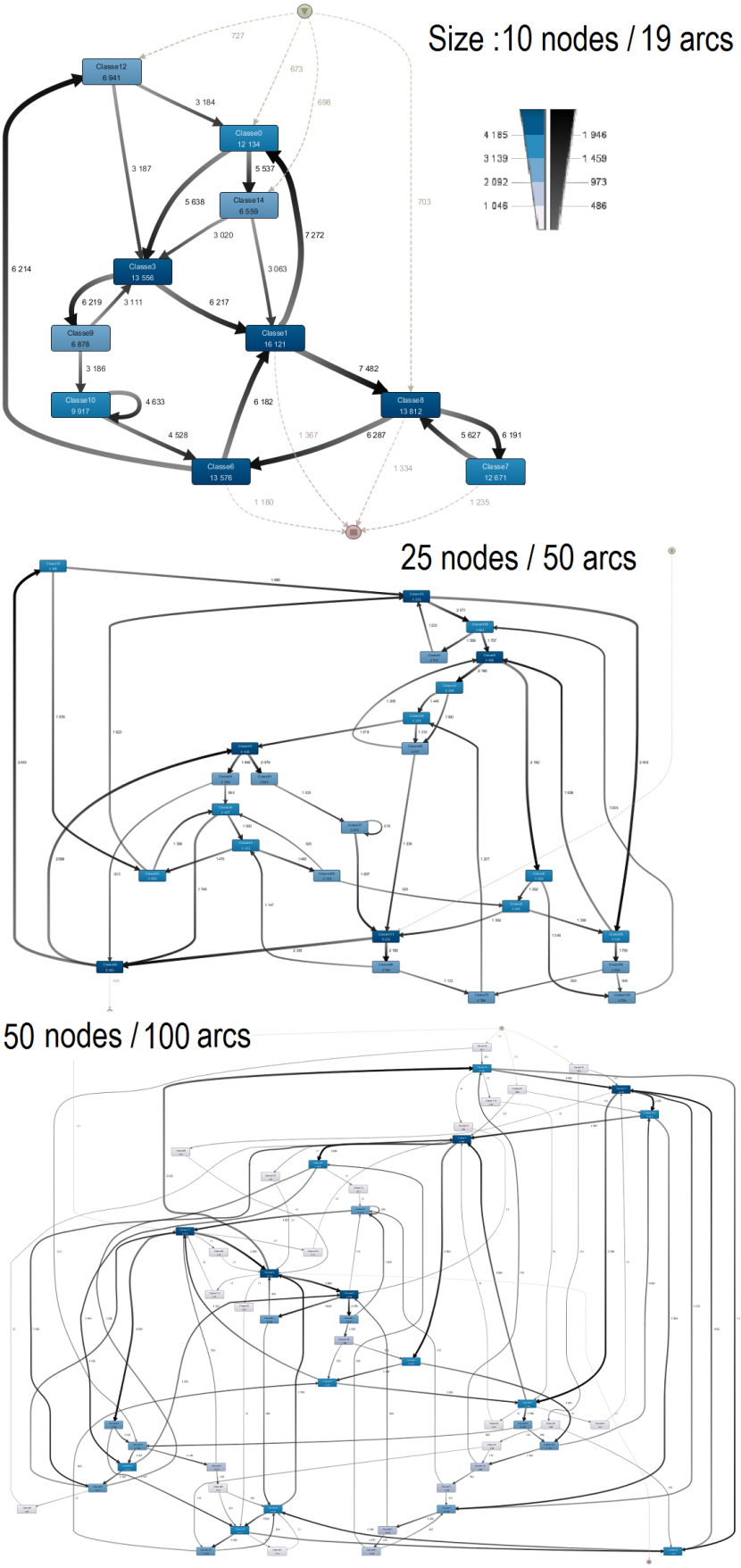


Figure 3.10: Models discovered by DISCO - 3 size of models

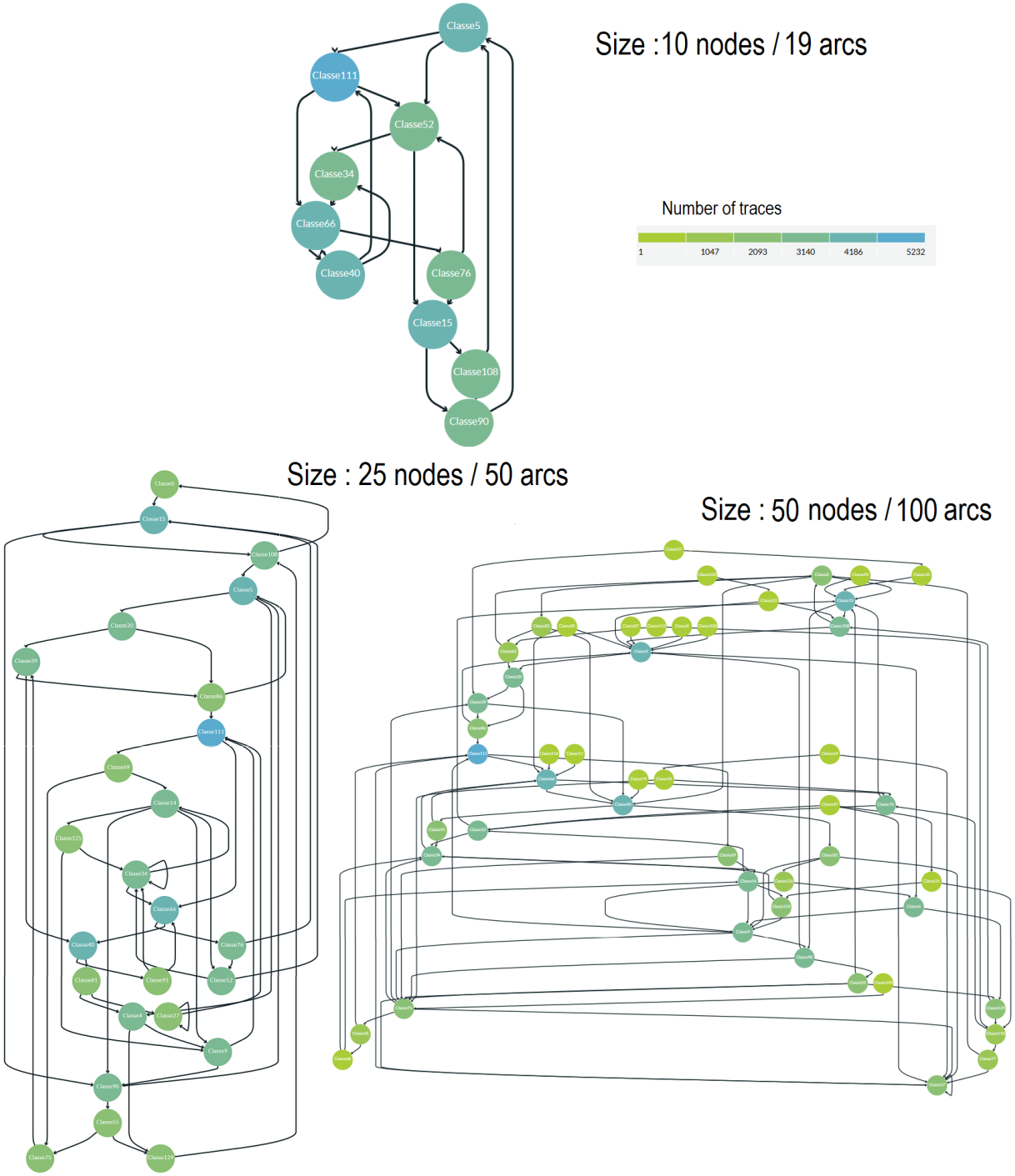


Figure 3.11: Models discovered by our tabu search - 3 size of models

Chapter 4

Health-care Analytics

Contents

Abstract

This chapter addresses the problem of enriching a process model which represents a clinical pathway. We specifically focus on the study of two perspectives: the decision point analysis and the time perspective. The decision point problem aims at finding relations between data attributes and the routing choices in the process. We formulate this problem as a classification problem where the outcome to predict is the next event in a patient's clinical pathway. We present the challenge that we face when using a noisy and heterogeneous log, such as health data. We tackle the problem related to the mismatch bias between the model and the log when learning a classifier. For that, we propose an innovative methodology which combines methods from the bioinformatics (sequence alignment algorithms based on similarity matrices) and the data mining fields. It enables the modeling of the interactions between a patient's features, his/her medical history, the natural evolution of a disease and his/her clinical pathway. In addition, we complete our clinical pathway model with duration thanks to classical distribution fitting. The entire methodology is generic and automated, so that it can be re-used straight on new data sets.

4.1 Introduction

In the previous chapter, we saw how to discover a process model from large and complex event logs. A process model can be used as such, as a first approach, for a control-flow analysis and a search of possible improvements. However, we can go further in the process analysis. The next step after discovering a process model is to build a simulation model. Using the health-care database of all the hospital events of numerous patients, our final objective is to create a simulation model of their clinical pathway. A simulation model gives life to the static control-flow process model. Several additional elements are needed to enhance a process model into a simulation model:

- The model's structure: we need to convert the nodes and the edges of the process model into the actionable states of a simulation model.
- A set of rules to choose a path at the output of each node, it is the **decision point analysis**.
- A set of probability distributions for the time spent in each state.
- A set of probability distributions for the time spent between two states.
- A set of probabilities to assign the first state to a simulated patient's sequence.
- A set of probabilities to decide if the simulation stops after each state.

In this chapter, we specifically focus on the analysis of decision points and of probability distributions. To this extend, we propose an *analytic toolbox*. The next chapter is dedicated to the conversion procedure of a process model into a simulation model, including the model's structure and the set of probabilities.

The idea of adding other perspectives into a process model has already been addressed in several ways. In addition to the control-flow perspective, an event log may contain valuable information about the organizational perspectives, the entity perspective or the time perspective. The general approach that consists in adding such perspectives is referred to as *model enhancement* (van der Aalst, 2011, chapter 8). The extension of a model is done by cross-correlating elements of the process model with the log. On this topic, most works of the literature have specifically focused on the **organizational perspective** (Rozinat and van der Aalst, 2006a; van der Aalst, 2011, 2010). The social network analysis enables to

determine the relationships among the organizational entities (resource, person, role, department, etc.) that are involved in the different activities of the process. It gives insights about how often these entities work together to achieve a common task and if there are handovers (Rozinat and van der Aalst, 2006a). The analysis of resource behavior is also possible if the appropriate time-stamps are present in the event log. A performance evaluation of the resource can be performed, such as utilization rates, response times or shortages (van der Aalst, 2010). In most organizations, the matter of resources efficiency is of major interest. For instance, identifying and reducing resources' idle times on a production line is both cost saving and increases productivity. In our application domain, our goal is to study clinical pathways at a macroscopic level. A patient is followed over several years and the smallest unity for an event is a hospital stay. An event duration is between half a day to several weeks. At this scale, we do not consider the hour by hour management of a patient within a hospital. Hence, we do not include any organizational perspective for our model enhancement. We assume that a patient can always find a hospital with the required facility to take care of him/her since he/she can move across a large territory. In our case studies (Chapter 6), patients come from any location of the French territory and are allowed to choose the hospital they want. Our approach here is driven by the specificity of the health-care domain.

Another perspective that one may want to add to process models is the **time perspective**. By definition of an event log, all the events have a time-stamp. Moreover, in practice, they almost always have a duration. The analyses of the time spent by entities in each event or between events can serve several purposes: finding bottlenecks, analyzing the service level, monitoring resource utilization or predicting the remaining process time of ongoing cases (van der Aalst, 2011). Section 4.6 of the present chapter presents how we derive statistical distributions from the historical data for the duration and for other entity attributes. Finally, the perspective that we are the most interested in here is the mining of rules that explain the **decision points** of the model. The topic of *decision mining* has been much less addressed compared to the discovery and the conformance checking of a process model. The scientific contribution of this chapter is the combination of several existing methods from different fields (process mining, data mining and bioinformatics) to solve the decision point problem in a health-care context. The decision points that we are facing in a clinical pathway can represent either medical decisions coming from doctors or the evolution of a disease. To the best of our knowledge, being able to model and explicit such decision points automatically from event logs has not been done in the literature.

The remainder of this chapter is organized as follows. Section 4.2 discusses existing literature on decision mining. Section 4.3 introduces a formal definition of the decision point problem that we want to solve. It also presents our motivation to develop a new methodology that can avoid a mismatch bias due to complex and heterogeneous logs, as found in health-care. Section 4.4 explains how we use a sequence alignment algorithm to align each trace of the original log with a fictitious trace generated from the model. Then, the decision point problem is solved using these traces in Section 4.5. Finally, the model enhancement is completed in Section 4.6 thanks to statistical distribution fitting.

4.2 Literature review

In the field of process mining, process discovery is the most studied approach (Gunther and van der Aalst, 2007; van der Aalst, 2004; van der Aalst et al., 2005; van Dongen et al., 2005; Weijters et al., 2006). Process discovery is the necessary and initial step when starting to analyze an event log. Many algorithms were developed in the last two decades to address various situations based on the type of event log (small, large, noisy, heterogeneous, etc.). Our work on optimal process discovery perfectly illustrates the ongoing search

for process discovery methods that can deal with large data sets, which are more and more encountered in practice. Once a process model has been discovered, we want to analyze how data attributes influence the choices made in the process by looking at the historical data. The topic of finding dependencies between the data attributes and the routing choices of entities in the process is referred to as *decision mining* or *decision point analysis* (Rozinat et al., 2009; Rozinat and van der Aalst, 2006b; Suriadi et al., 2013).

One of the first work dealing with the complementary combination of a process mining technique for process discovery and of a data mining algorithm for decision point analysis was done by (Rozinat and van der Aalst, 2006b). The idea was to take benefit of the available and unused data attributes of the log to enrich the knowledge about the decision points. It was comforted by several data mining (or machine learning) algorithms that had become widely used and proven efficient to extract knowledge from large data sets (Mitchell, 1997; Witten and Frank, 2005). No new data mining had to be developed. The originality is to propose a combined use of process mining and data mining. The very first step to analyze decision points is to identify them in the process model. The work of (Rozinat and van der Aalst, 2006b) performs this task using a Petri Net formalism: a decision point corresponds to a place with multiple outgoing arcs. Figure 4.1 illustrates the decision mining approach developed in the context of Petri Net (Rozinat and van der Aalst, 2006b). Our approach here follows the same logic, but for a few differences discussed above. As shown at the top of Figure 4.1, the starting point is to have an event log where at least a case ID, a time-stamp and a labeled activity are known for each event. Then, a process model is discovered using a conventional process mining algorithm (e.g. the α algorithm (van der Aalst, 2004)). Then, all the decision points are identified and analyzed with a classification algorithm. For each decision point, it produces routing decision rules. The overall approach of (Rozinat and van der Aalst, 2006b) was implemented as a *decision mining* plug-in in the open software ProM which is dedicated to process mining and its extensions (Rozinat and van der Aalst, 2006c). The methodology presented in Figure 4.1 was reused as an intermediary step for the conversion of a Petri Net into a simulation model (Rozinat et al., 2009). The topic of simulation model conversion is addressed in Chapter 5.

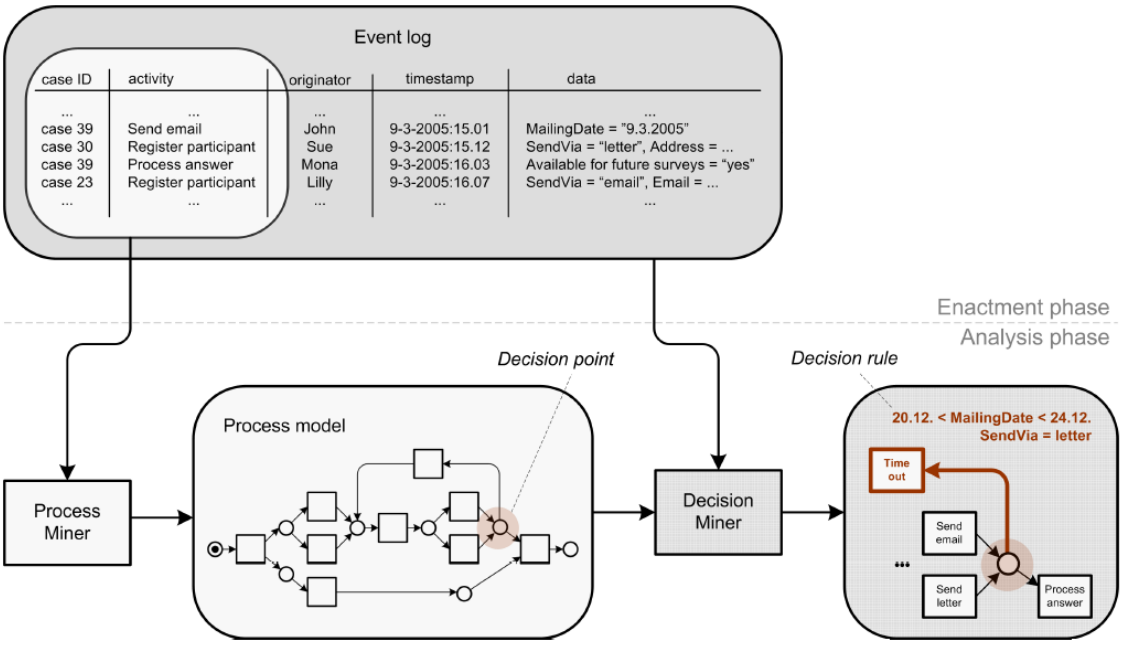


Figure 4.1: The general approach of decision point analysis (Rozinat and van der Aalst, 2006b)

In (Rozinat and van der Aalst, 2006b), the authors highlight that the first challenge that practitioners face when applying decision mining is the quality of the data and their correct interpretation. Noise in the data is an obstacle to the mining of proper rules. The need for a noise-robust algorithm and the nature of data (volume, type, heterogeneity) will drive the choice of the data mining algorithm to apply. A good knowledge of data attributes is also necessary to understand and interpret the resulting decision rules. Whatever the case study, this prerequisite is unavoidable. Data knowledge can be acquired during the upstream step of data preparation, in which the practitioner must dig into the data and their meaning. This is even necessary before starting process mining. Once all the traces' attributes have been properly defined (at the very first step when looking at the available event log), we assume that the selected attributes are relevant to the studied process. In this chapter, we propose an innovative methodology to transform the original, and possibly noisy traces of the log, into so-called perfectly replayed traces (Section 4.4). It avoids the under-sampling of historical observations and the creation of a biased classifier. It also provides a perfect mapping function between the log activities and the process model.

Among the variety of existing data mining algorithms (Witten and Frank, 2005), a decision tree classifier is proposed in (Rozinat and van der Aalst, 2006b). This choice is motivated by the capacity of decision trees to handle continuous, discrete or categorical variables, and also missing variables. It is a guarantee for better practical use. They also discuss a way to handle the cases of invisible activities and **duplicate activities** when met in the context of Petri Net. Invisible activities are specific to this formalism, but duplicate activities also occur in other process model formalism. There are two types of duplicate activities: (i) two nodes of the same model can represent the same activity but performed at different moments of the process, (ii) there is a loop in the model that allows to come back a second time to a node. In both cases, it raises the question of whether or not to use the same classifier in the duplicate activities. Here, as described for the optimal process discovery, we made the assumption that duplicate nodes in a process model are not allowed whereas loops are. We consider that the same decision point classifier is used whatever the amount of visits. We assume that the difference is taken into account in the traces' attributes, such as in the "number of already visited nodes" and "total time spent in the process so far". Then, attribute values will be higher at a trace's second visit of this node, which may result is a different routing choice if the classifier integrates this parameter for the decision. This assumption implies that a trace for the log that had several times the same event is considered as several separate observations for the learning of the classifier.

The limitations of (Rozinat and van der Aalst, 2006b) and (Rozinat and van der Aalst, 2006c) regarding the decision point analysis are the following. First, only Petri Nets are considered. They do not consider other types of process models. We propose a complementary approach using causal nets. Second, no performance measures are proposed to discuss the quality of the learned classifier. The initial data are usually split in two, one for the learning phase and one the testing phase. Testing the algorithm on unseen data reveals the predictive capacity of the learned rules. It helps detecting over-fitting or other undesired behaviors (Witten and Frank, 2005). Furthermore, only one data mining algorithm is considered for the classification problem, a decision tree based algorithm. The rich literature in the data mining field has proven that no supervised learning algorithm outperforms the others on all problems. Hence, testing several algorithms can improve the quality of the learning results. If an exhaustive comparison of all the existing methods is unfeasible on each practical case study, it is recommended to compare the performances of the most popular algorithms (Decision Trees, Support Vector Machine, Neural Networks, Naive Bayes, etc.) (Witten and Frank, 2005).

Due to its potential benefit for an in-depth analysis of processes, decision mining has been taken over in other works (Suriadi et al., 2013; de Leoni et al., 2016). The idea of (Suriadi et al., 2013) is to enrich

event logs to make it as suitable as possible for any upcoming *root cause analysis*. The aim of root cause analysis is to find an explanation for why things happen. It determines the factors that influence the success or the failure of a process. Hence it can be turned into a classification problem. This approach is a general contribution to the analysis of a process model because it focuses on the enrichment and the transformation of a log into a classification-ready log. It may be specifically applied to the case of the decision point analysis. The approach was shown to achieve good performance results to predict on-time process instances when using a decision tree algorithm or a rule-based algorithm. Following the idea of (Suriadi et al., 2013), we propose a new method to enrich a log for the decision point problem. Based on the observation that original logs might be noisy and too heterogeneous compared to a discovered process model, we propose a log transformation to make the classification problem feasible (Section 4.4). The work of (de Leoni et al., 2016) goes one step further by proposing a general framework that unifies existing approaches for process-related correlation analysis. This framework defines formally how various data sources related to the process (traces' attributes, resources, control-flow, organizational and time perspective) can be used to correlate, predict and cluster the dynamic behavior of the model. They introduce concepts such as log transformation, trace manipulations or event-filter to solve any process-related question. It gathers existing approaches in one environment.

To sum up, existing literature on decision mining in the context of process models has proven the interest of combining data mining and process mining to get new insights about data interactions. Most works have built their analysis based on the most used formalism for process modeling, a.k.a. Petri Nets. Other types of process models have been supported. Furthermore, the methods would benefit from more feedback from case studies on real data. Following the perspective of the previous chapter, our goal is to propose a reproducible methodology that is suitable for large and complex data sets such as in health-care. The main contributions of the present work to the field is to adapt existing approaches on the causal net formalism, to integrate a sequence alignment algorithm to enrich the log and to propose a real application on a large health data set.

4.3 Decision point analysis

4.3.1 Definition of the decision point problem

A process model is an abstracted view of the reality that intends to represent what happened as described in the data. It is like a picture that shows the different possible sequences in the process. Let $PsM = (N, E)$ be a process model discovered using the approach described in the previous chapter. In this causal net notation, all the incoming joins and the outgoing splits of the nodes are exclusive disjunctions (XOR): exactly one path is chosen in the flow. However, nothing stipulates how a path is chosen over the others. To be able to simulate new traces, we need to know the rules to decide which paths are chosen at the output of each node.

Our purpose here is to determine a set of decision rules, one for each node, that assign the probability of following each outgoing path and which are the most representative of the event log. More specifically, we consider a given log L, a process model $PsM = (N, E)$. The problem consists in determining a set of functions f_{n_i} in order to maximize some likelihood function, where $\forall i \in [1, |N|]$, $n_i \in N$. Any given node $n_i \in N$ may be followed by $\{0 : n\}$ transitions $e_1, ..., e_n \in E$. $\forall n_i \in N$, we define $Successor_{n_i} = \{n_j \in N | (n_i, n_j) \in E\}$ the set of all n_i's successors in PsM. $Successor_{n_i}$ may be the empty set. The number of

elements in $Successor_{n_i}$ is denoted by z_i. Then,

$$f_{n_i} : \begin{array}{ccc} \mathscr{L} \times N & \rightarrow & Successor_{n_i}^{z_i} \times [0,1]^{z_i} \\ (L,n_i) & \mapsto & \{n_1,...,n_{z_i}\} \times \{p_1,...,p_{z_i}\} \end{array} \text{, with } \sum_{i=1}^{z_i} pi = 1 \tag{4.1}$$

where $\mathscr{L}$ is the set of all possible logs. Then, the decision point problem is defined as follows:

$$\underset{\{f_{n_1},...,f_{n_{|N|}}\}}{\arg\max} \; likelihood(\{f_{n_1},...,f_{n_{|N|}}\},L) \tag{4.2}$$

where $likelihood(\{f_{n_1},...,f_{n_{|N|}}\},L)$ is a measure of the likelihood of using $\{f_{n_1},...,f_{n_{|N|}}\}$ functions in the simulation model compared to the actual log L. Such a metric is defined in Chapter 5, Section 5.5, where we assess the validity of the newly built simulation model compared to the log. If $Successor_{n_i}$ is empty, then f_{n_i} is the null function (every probability is zero).

Solving the optimization problem (1) to optimality is out of the scope of the present work, just as an extensive search of all the possible sets of functions $\{f_{n_1},...,f_{n_{|N|}}\}$, but it would be of interest to dedicate future works to it. Indeed, the choice of a set $\{f_{n_1},...,f_{n_{|N|}}\}$ impacts the fidelity of the decision points compared to what actually happened. Here, we want to highlight the existence of a methodological bias that arises when choosing a set of functions $\{f_{n_1},...,f_{n_{|N|}}\}$ and the difficulties that arise when choosing such a set of functions. To illustrate this bias, we introduce a systematic, and rather simple, method to find a set $\{f_{n_1},...,f_{n_{|N|}}\}$.

4.3.2 The mismatch bias for the decision point problem

A simple method to determine the probability of following each path of the process model is to use the historical probabilities found in the log. We consider a log L and a process model $PsM = (N,E)$. Let $n_i \in N$ be a node of PsM, n_j be a successor of n_i. Then, the probability that n_i is followed by n_j is equal to the number of traces in L that actually had this transition divided by the number of traces that had n_i. Formally, it gives:

$$\forall n_i \in N, f_{n_i}(L,n_i) = \{(n_j,p_j)\} \tag{4.3}$$

$$= \{(n_j, \frac{|n_j \Rightarrow n_i|}{|n_i|})\} \tag{4.4}$$

This formulation is rather simple as it only uses the occurrence frequency of a transition in the data to calculate its probability. Still, it suffers from a problem of imperfect matching between the data and the process model. Indeed, when dealing with large and complex logs, not every event class and transition observed in the log can be represented in the discovered process model. Only the most relevant classes and transitions, as defined in the replayability score, are kept to respect the size constraint (Chapter 3). Hence, when looking at the original data to compute the decision path probabilities, many traces are unused. Those traces are not perfectly replayed in the model because all their events are not represented, so they cannot contribute to decision point probabilities.

An example of such a mismatch between a process model and traces is shown on Figure 4.2. The illustrated process model has 3 nodes, including two nodes A and C with only one output edge, and one node B with a decision point. The 2 possible paths after B are C and B itself. The log is made of 4 traces which all had the event B, $\sigma_1, \sigma_2, \sigma_3, \sigma_4$. The analysis of B's decision point is made by counting the traces

which had the event B and each of the 2 possible following paths (B and C). Following a B, σ_1 had a C, σ_2 had a B and D (as it had two B), σ_3 had a E, σ_4 had a E. Finally, the distribution is 20% for C, 20% for B, 20% for D and 40% for E. However, only B and C are possible in the process model, so the final distribution is 50% for C and 50% for B (we ignore D and E contributions). This final result is only based on two observations, instead of the five available. A more in depth analysis of both σ_3 and σ_4 shows that the event B was followed by E, and eventually by C. It suggests that σ_3 and σ_4 would contribute to the path "B followed by C" but they could not because of a slight difference in their sequence compared to the process model. The point is that, as illustrated in the example of Figure 4.2, the difference between the process model and the traces only concerns some specific events and that a slight change in a trace sequence would avoid the mismatch issue.

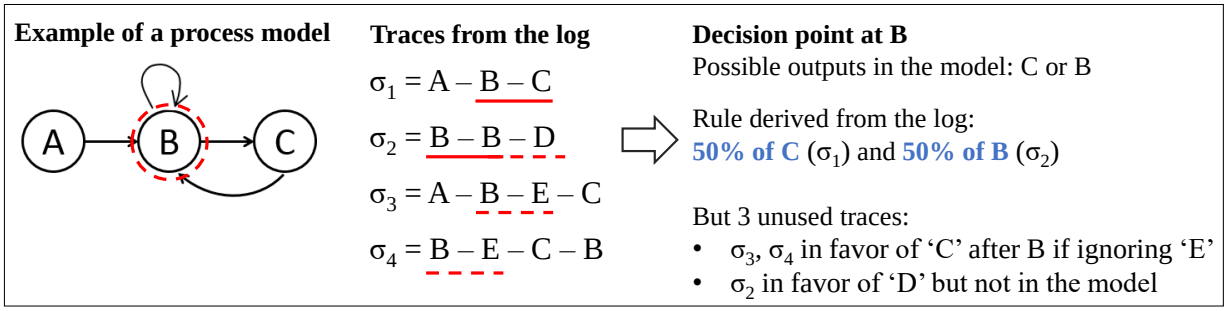


Figure 4.2: Imperfect matching between a log and a process model for decision point analysis

This bias induces two drawbacks in the computation of the decision point probabilities: it creates undersized samples that make the probabilities unreliable, and it does not accurately represent the historical data as most of it is ignored. These two drawbacks are extremely dominant when using large and complex logs. A way to avoid the mismatch bias would be to have a process model representing exactly all the possible behaviors seen in the data. It is possible when the model is extremely large but it is contradictory with the idea that a model must represent as many behaviors as possible while being as small as possible.

In the following, we present an innovative methodology based on the transformation of the original traces to avoid the mismatch bias. This methodology is made of 3 steps: (1) first we generate the list of all the traces that can be perfectly replayed in the process model, (2) for each trace of the log, we find the closest perfectly replayed traces using the Needleman-Wunsch algorithm, along with a similarity matrix of events, then (3) we use the perfect assigned trace instead of the actual trace for the decision point analysis.

4.4 Perfect traces generation and sequences alignment

4.4.1 Perfect traces generation from a process model

For a given process model, our objective is to generate all the possible perfect traces derived from that model. The objective of this step is to know all the possible sequences that could be 100% replayed in the process model, unlike most of the actual traces from the event log. We design an automatic procedure that generates such perfect sequences. The procedure is presented in Algorithm 2. For any node of the model, we create a trace of length 1 whose sequence is made of the event in the node (step 1). Then, for each possible successor of this node, we create a duplicate of the length-1 trace and we add the successor to its sequence. We iterate this process until there is no more successor to a node or until the generated trace has reached a maximal size threshold (recursive step 2). This second stopping criterion is used to avoid infinite-

sized traces. Indeed, as a model can have loops, i.e. there exists a path from a given node to another node of the model which was previously visited in the sequence, it may possibly generate infinite-sized traces. For that reason, a *maximal size* value is imposed to perfect traces. We set this threshold as the size of the longest trace found in the log.

Finally, any sub-trace of the traces generated with the above procedure is also added to the set of perfect traces (step 3). Hence, for each generated trace, we obtain the set of all traces whose size is between 1 and *maximal size*, and which is perfectly re-playable in the model. The number of perfect traces grows exponentially with the number of nodes and edges in the model.

Algorithm 2 Generation procedure of perfect traces from a process model

Require: $PsM = (N, E)$, a process model
1: **Step 1 – Initialization**
2: Let $L = \{\emptyset\}$ be the set of the generated perfect traces (initialized as empty)
3: **for all** $n \in N$ **do**
4: Create $\sigma =< n >$, (a trace with one event)
5: Set $Current_{node} = n$
6: **Step 2 – Recursive generation of all possible traces**
7: **if** ($Successor_{Current_{node}}$ is empty) OR ($lenght(\sigma) > Size_{max}$) **then**
8: $L = L + \{\sigma\}$
9: **else**
10: $\sigma_0 = \sigma$
11: **for all** $n' \in Successor_{Current_{node}}$ **do**
12: Create $\sigma' = \sigma_0 + < n' >$, (duplicate σ and add an event to the sequence)
13: $Current_{node} = n'$
14: Go back to step 2 with $\sigma = \sigma'$
15: **end for**
16: **end if**
17: **end for**
18: **Step 3 – Sub-traces**
19: **for all** $\sigma \in L$ **do**
20: Add every sub-trace of σ to L
21: **end for**
22: **Step 4 – End of the procedure**
23: Return L, the set of all the perfectly replayed traces in PsM

This procedure has proven to be exhaustive as it provides all the perfect traces whose size is between 1 and *maximal size*. However, it is not optimized from an algorithmic point of view. The same sequence is duplicated as many times as there are routing choices in the model, which requires hard memory. A tree structure would be more memory saving than storing each possible trace individually. Further improvements could be brought to improve the computational performance of the procedure. It would be especially needed when the number of possible traces rises to several million.

4.4.2 Sequence alignment for trace mimicry

The purpose of generating all the perfect traces derived from the process model is to align them with the traces from the original log. For each original trace, we want to find the perfect trace with the closest sequence. So far, a trace was defined as a sequence of events. We now define a new type of trace, an

enhanced trace, as the combination of a trace and a set of features. A trace's closest perfect trace will inherit the features of the original trace and will be used to solve the decision point problem. This method takes benefit of more information from the original log by avoiding the mismatch bias.

Definition 1 (Features of a trace)**.** *Let T be a set of events, L a log over T and $\sigma \in L$ a trace. We define $F = \{f_1, ..., f_x\}$ a finite set of attributes (age, gender, medical history, medical cost, follow-up duration...), $x \in \mathbb{N}$. The features of σ, denoted $\sigma(F)$, is a set of x values defined on F, one for each of the attributes.*

Definition 2 (Enhanced trace)**.** *Let T be a set of events, L a log over T. An enhanced trace σ^+ is a 2-tuple composed of a trace and its features: $\sigma^+ = (\sigma, \sigma(F))$.*

The alignment of two sequences requires the definition of a score measure that quantifies the distance between them (similarity matrix). Then, we use the Needleman-Wunsh algorithm to find the best alignment regarding this score. The best alignment between two sequence is found by trying to add (or remove) one or several events in one of the sequence, or by inserting a gap, in order to get two identical sequences. Finally, the distance between two sequences is the amount of operations (add, remove or insert) that is needed. The following presents in detail the score function and the Needleman-Wunsh algorithm.

Alignment score and similarity matrix

The first step to align two sequences of events is to be able to quantify the closeness between two sequences of the same size. To compare two sequences and evaluate their closeness, we need to be able to compare two events. There are 3 possible different situations when comparing one event of a sequence to one of another sequence: they are the same (= match), they are different (= mismatch) or one event aligns with a gap in the other sequence (=gap insertion). For instance, when comparing the sequences A-B-C and A-B-D, we see that 2 events match out of 3. The next question is to know if A-B-C is closer to A-B-D, to A-B-E or to A-*(gap)*-C. For that, we propose a method to evaluate an alignment score between two sequences.

Such score enables to quantify the closeness of two sequences of the same size. The closeness score of two sequences is equal to the sum of the closeness of their events taken one-by-one. The higher the score is, the closer the sequences are. Figure 4.4 presents an example of a similarity matrix and of the closeness score of two sequences. In this case, there are only 3 different events (A,B,C) and the gap factor is set at -5. Three sequences are compared by computing the two-by-two closeness score. The closeness score between sequences 1 and 2 is 41 and is higher than the score between sequences 1 and 3, and 2 and 3. It means that sequences 1 and 2 are the closest sequences in this example. To compute the alignment score between two sequences, we need to define a similarity matrix: it is a square and positive matrix whose size is equal to the number of different events in the log. For any pair of events (e_1, e_2) the matrix gives a similarity score between e_1 and e_2. The higher the score is, the more similar e_1 and e_2 are.

This matrix can be created in two ways: the first way is on a case-by-case basis where a domain experts chooses a closeness value for each possible pair of event. This can be a tedious task when there are hundreds of possible events, and the result heavily depends on the expert's opinion. The advantage is to include a strong domain expertise that is impossible to have otherwise. In the medical field, in-depth knowledge is required to assess the similarities and differences between two diagnosis or two surgeries. The second way to create the matrix is to use a hierarchical structure of data. This approach is recommended for large scale data. The idea is to create clusters of similar events with different levels of aggregation. For instance, in health-care, "magnetic resonance imaging" and "radiography" can be gathered in a cluster "imaging activities", which again can be gathered with "blood tests" into a super-cluster "non-invasive

medical examinations", and so on. The closeness between two events corresponds to the inverse of the distance, as the number of clusters in the hierarchy between them.

$$Closeness(e_1, e_2) = \frac{1}{\text{Length of the shortest path from } e_1 \text{ and } e_2 \text{ in the hierarchy}} \tag{4.5}$$

This definition makes the similarity matrix symmetric (the distance between A and B is the same as the distance between B and A). Figure 4.3 shows an example of such a hierarchical structure with health-care events. The event "magnetic resonance imaging" is only 1 cluster far from "radiography", but 3 clusters far from "magnetic resonance imaging". Some events, like "radiography" and "'neurosurgery' cannot be compared at all, then their closeness is null (equivalent to say that their distance is infinite).

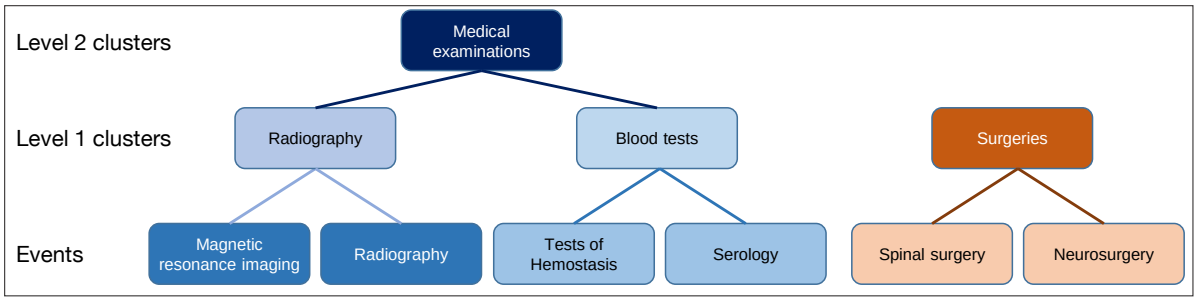


Figure 4.3: Example of a hierarchical structure of health-care events

In addition to the similarity matrix, we use a *gap factor* that penalizes the presence of a gap in a sequence. This factor is necessarily a negative number. Finally, the similarity matrix and the gap factor enable to quantify the closeness of two sequences of the same size.

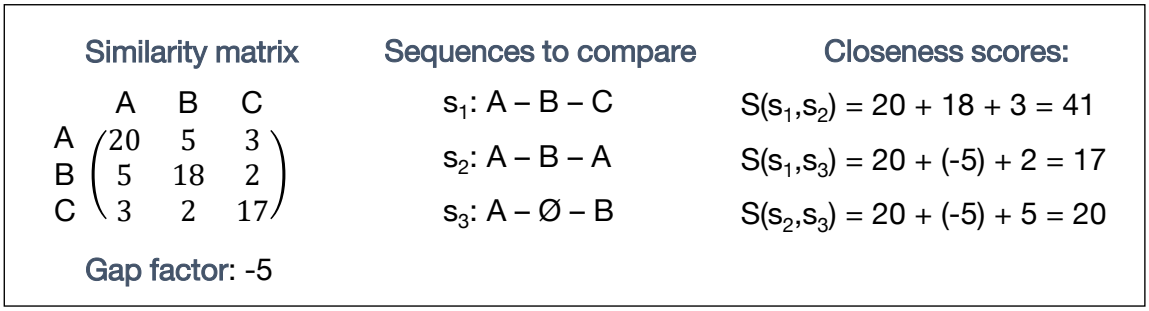


Figure 4.4: Example of closeness scores between two sequences of events

Once we can compute the closeness score of two sequences of the same size, we are interested in a generalization for any size of sequence. For two given sequences of any size, we need to find the best possible alignment of these two sequences that will end up in two sequences of the same size. For instance, if we search to align the sequences A-B-C and A-C, we see that one possible best alignment is to transform A-C into A-*(gap)*-C. Then, we can compute the closeness score of A-B-C and A-*(gap)*-C as explained previously. Finding such best alignment between two sequences is not trivial and requires a dedicated algorithm. In the following, we introduce the Needleman-Wunsh algorithm to do so.

Sequence alignment with the Needleman-Wunsh algorithm

The Needleman-Wunsch algorithm is an algorithm that was originally developed in 1970 and used in bioinformatics to align protein or nucleotide sequences (Needleman and Wunsch, 1970). The goal was to perform a global sequence alignment between two sequences and to find out structural or functional similarity. In the field of DNA analysis, when a new sequence is found, the structure and function can be predicted thanks to sequence alignment. It relies on the belief that a sequence sharing common ancestor would exhibit similar structure or function. Hence, higher the sequence similarity, greater is the chance that they share similar structure or function.

The Needleman-Wunsch algorithm is a dynamic programming approach which divides the global problem into smaller independent sub problems. It saves tremendous computation time as it does not need to enumerate all the possible alignments between two sequences. The complexity of the algorithm is $O(n^2)$, n being the length of the longest sequence. The algorithm browses the two sequences event by event and determine the outcome score of three possibilities based on the given position in the sequence: it is a perfect match (diagonal elements of the similarity matrix), it is a mismatch (other elements of the similarity matrix) or a gap is inserted (gap factor). At the end of this procedure, the algorithm produces a 2D rectangular matrix whose size is the one of the two studied sequences. It contains the values of the best action among the 3 for any relative position of the two sequences. Finally, the optimal alignment is obtained by finding the path from the top left element to the bottom right element of the matrix that maximizes the total score. (See (Needleman and Wunsch, 1970) for a more in depth explanation of the dynamic programming algorithm).

By searching the highest scores in the matrix, the best alignment can be accurately obtained. The advantage of using this algorithm is its guarantee to find the optimal alignment between two sequences and its low computational complexity. The main challenge of using such method is that it heavily relies on the choice of a reliable similarity matrix and of a gap factor. When dealing with a domain such as DNA sequencing where the sequence elements are well documented (there are only four possible nucleotide bases A, C, G and T) and ruled by biological facts (e.g. probability of a mutation between two nucleotides may be evaluated following chemical properties), the corresponding similarity matrix and gap factors can be more easily derived. In our case, we are dealing with much shorter sequences than DNA, but with a much larger variety of events (several hundreds for a typical clinical pathway case study compared to four bases for DNA and 26 amino acids for proteins). Hence, finding a reliable similarity matrix is much more of a challenge. This is why we decided to use the hierarchical structure of the events to compute the similarity matrix. Our approach relies on a well established medical classification, but it can accept some afterward modifications based on expert opinion. The choice of a similarity matrix can also be evaluated when validating the simulation model that will be ultimately built at the end of the chain. Indeed, the behavior of the model will be compared both to the perfect traces and to the original traces.

Eventually, each trace of the original log is aligned with each perfect trace that was generated. The perfect trace whose alignment induces the highest score is elected as the best matching sequence for the original trace. Each perfect trace is converted into an enhanced trace by adding the features of its original counterpart. The original log is thus replaced by a new log with these enhanced perfect traces. The newly created log can now be used to solve the decision point problem without a mismatch bias.

Discussion on sequence alignment

The aforementioned method enables to align two sequences of events in order to quantify their similarity. We propose to use this to transform original logs into logs with only perfectly replayed traces. This new log

is more suitable to solve the classification problems of a model decision points. Nevertheless, this method could also be used outside of this context. Comparing two sequences of events is a broader problem than data preparation for classification. In health-care (the same applies for other domains), comparing the clinical pathway of two patients is not straightforward when they have had tens of medical events. Here, we propose an objective and formalized way to achieve this goal, with the advantage that the alignment rests upon the strong medical expertise that is captured in the similarity matrix. A two-by-two sequence comparison can also be used to compare a patient's pathway with a guideline pathway. This would then bring the same type of responses as *conformance checking* (van der Aalst, 2011).

Summary of the sequence alignment method

To sum up, we propose a method to quantitatively compare two sequences of medical events, which constitutes the first element of our analytic toolbox.

We consider the following hypothesis.

1. H_1: we consider a similarity matrix M to measure how similar two process events are
2. H_2: we introduce a gap factor G to penalize the presence of gaps in sequences
3. H_3: we define a score function which uses M and G to quantify the closeness of two sequences

The analysis method can be summarized as follows:

1. Generate all the perfect traces (set P)
2. Extract all the traces from the log and enhance them (set E)
3. For each trace of E, find the closest perfect trace in P using the Needleman-Wunsch algorithm

4.5 Classification models to solve the decision point problem

Previously, we have defined the decision point problem that we need to solve to determine the routing rules of a process model. We have introduced a first way to solve this problem, which was to use the historical probability that each path is chosen. This method totally applies to the newly created log with perfect traces. The advantages of the probabilistic method are to be simple to compute, straightforward to understand and it does not need any configuration. However, it is a short-sighted approach that only considers a single piece of information to determine the next node: the current state. Indeed, it does not consider the previous elements of the sequence or any feature of the trace. Our ultimate goal is to create a simulation model of the process that is capable of reproducing as much as possible the behavior seen in the data. This is why we now focus on more advanced methods from the field of machine learning to solve the decision point problem.

Machine learning for health-care consists in developing algorithms that learn to recognize complex patterns within valuable and massive data. Challenges related to that topic are numerous, and many scientific fields are involved: computer science, data science, operational research. When applied to health-care, the objectives are often summarized as improving quality and timeliness of care, maximizing financial performance, and decreasing practice variability across organizations. It relies on the following tasks: (i) identify critical features that impact outcomes (allocation of limited resources/time for greater effectiveness); (ii)

seek greater use of treatment evidence to advance the quality and effectiveness of care delivery; (iii) rapid learning and best practice dissemination.

The following is organized as follows. Section 4.5.1 states the decision problem as a classification problem, Section 4.5.2 explains how to prepare the data from the log to perform the learning, Section 4.5.3 discusses the choice of an appropriate machine learning algorithm to solve the classification problem and Section 4.5.4 presents the validation step of the results.

4.5.1 Decision points as a classification problem

A process model is an oriented graph in which a decision point corresponds to a node with multiple outgoing arcs. Since only one path can be chosen by a process instance, each time that several outgoing arcs can possibly be chosen after a node, a rule must be defined to explicit this choice. There is no way to know the number of decision points in a process model based on the number of nodes and edges. It depends on the structure of the process model, especially on the presence of loops or of terminal nodes with no outgoing edges. For each of the identified decision points of a process model, our objective is to determine with certainty what the followed path of an entity based on its features will be. This objective can be rephrased more specifically as a classification problem which includes the role of input variables in the search of the output path (Rozinat and van der Aalst, 2006b; Mitchell, 1997; Witten and Frank, 2005).

Definition 3 (Classification problem)**.** *Given a node of a process model where there is a decision point, the problem consists in finding the most reliable classifier that can derive the outgoing path based on rules including a set of input variables.*

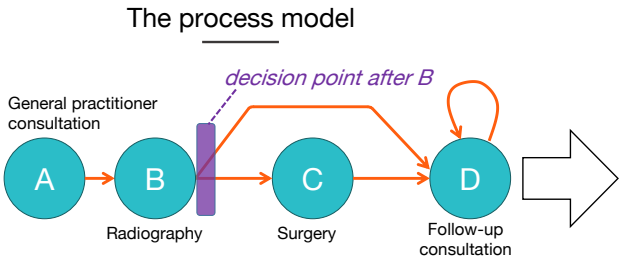
A separate classifier will be created independently for each decision point. The different outgoing paths that can be chosen are the output "classes" to predict and the traces features are used as the input variables. The creation of a classifier is done by learning from the historical observations, i.e. from the traces. The learning process intends to generalize the behaviors seen by discovering patterns between the input variables and the desired outcome. These patterns are formally expressed as a set of logical rules. There are 3 key steps to learn such rules from raw data.

1. Data preparation
2. Choice of the classification algorithm which will learn the rules from past observation
3. Validation of the newly discovered rules on new observations

From a case study perspective, each decision point of a clinical pathway is a critical moment of a patient's care. It can simultaneously model the natural evolution of a disease, the medical choices made by practitioners or the individual choice of a patient to see a doctor. For instance, after a cardiac arrest, a patient may either slowly recover, either have another cardiac event or even die. This choice is ruled by the complex dynamic of cardiac degeneration, the medical history of the patient and his/her features (including life conditions). It also depends on the decision of a physician to send a patient home or to keep him one more day after an acute hospitalization. The modeling of a clinical pathway decision points with a classifier is essential to represent these critical nodes of a patient's life, especially for the follow-up over several years. We do not pretend to provide a model that is capable of capturing all the interactions that take part in a care process. We propose the best possible model based on the available input variables in the historical data. The resulting model brings new answers to the understanding of long term clinical pathways.

4.5.2 Data preparation

For a given node x with a decision point, the learning phase is done by selecting all the enhanced traces (i.e. patients), among the aligned traces obtained at step 3, that had an event x in their sequence. Each selected trace becomes a learning observation where the target variable is "the next event after x in its sequence". The input variables that are used to learn the classifier are the traces' attributes. An important point is that only the trace attributes that were known until the x point are included in the analysis (e.g. the total time spent by the entity in the process is the time between the beginning and x, so is the total number of occurred events). If a trace's sequence contains twice the same event, then the trace is considered as two separate observations. An example of a ready-for-classification table of observations is shown in Figure 4.5. A process model with 4 nodes and 5 edges was discovered. There is one decision point that was identified, it is the routing after node "B" (radiography). Then, we extracted from the log the 5 patients that had a "B" event in their sequence. They are displayed on the right side of Figure 4.5. Regarding the technical part of the data selection, a program that automatically selects the relevant observations for each decision point of a given model was implemented in Python (version 2.7). This program is interfaced upstream with the output of the C++ program performing the optimal process discovery, and downstream with the Python machine learning library.



Observations for the decision point after B

Patient ID	Medical events sequence	Gender	Age	Radiography results	Next event after B
1	A-B-C	Female	50	Tumor	C
2	A-B-C	Male	48	Tumor	C
3	A-B-D	Male	51	Nothing	D
4	B-C-D-D-D	Male	35	Tumor	C
5	A-B-D	Female	85	Tumor	D

Figure 4.5: Example of a ready-for-classification log for a single decision point

4.5.3 Selection of a machine learning algorithm

The choice of a machine learning (or data mining) algorithm to perform a learning task is not straightforward. It depends on the data (size, quality, nature), on the question that we want to answer, on the available computational power, etc. Usually, we do not know the best algorithm before we try some of them. There are many available machine learning algorithms that can be used to perform different tasks (Caruana and Niculescu-Mizil, 2006; Witten and Frank, 2005). We present a list of 25 of the most popular machine learning algorithms in Appendix A. Algorithms are grouped according to the task they perform (regression, two or multi-classes classification, anomaly detection) and ordered by their number of parameters, either or not they use linearity and by their usual trend in accuracy and training time. Most of them are available in open source tools. For the practical use of the work presented in this thesis, we decided to use *Scikit-learn*, a free software machine learning library for the Python programming language. Scikit-learn includes all of the most used algorithms by researchers and practitioners, namely Support Vector Machines, Decision Trees, Stochastic Gradient Descent, Naive Bayes, Generalized Linear Models, Nearest Neighbors, Ensemble methods and Neural network (Wu et al., 2008).

Regarding the decision point problem that we want to solve, it is a supervised learning (the class is known for historical observations) and a multi-class classification problem. Here, **data quality** is not an

issue anymore because we performed a log transformation that ensures that each trace can be perfectly replayed in the model. Hence, at each decision point, the only possible paths are those present in the model. The question which cannot be answered with certainty here is the **number of observations** found in the log for each decision point of the model. It mainly depends on two factors: the size of the process model and the number of occurrences of each event in the log. The impact of the latter on the number of training observation is immediate. The size of the model (i.e. the number of nodes and edges) has an indirect impact. The bigger the process model, the higher the number of nodes and edges, and thus the number of possible paths between two points of the model increases. It means that the traces generated for sequence alignment will take a larger variety of paths in the model. It induces a mechanical diminution of the number of traces that specifically enter in each decision point. This phenomenon is described in a qualitative manner here, a precise quantification of the amount of observations shall be proceeded for each case study. However, based on the assumption that a discovered model is only valuable and kept if it is representative of the log, it means that each element of the model was sufficiently seen in the log compared to the number of actual traces. By nature of a data mining approach, there is preconceived rule that could be applied to determine the minimum number of observations required for a classification task. Instead, past experience and afterward validation are used to produce recommendations on the size of the learning sample (Mitchell, 1997).

In the following, we discuss 2 data mining approaches for our decision point problem: decision tree algorithms, similarly as the method proposed in (Rozinat and van der Aalst, 2006b), and ensemble methods to increase learning performance. We will specifically present random forests.

Decision Trees: popular and explicit classifiers

Decision tree learning is one of the most practical and widely used methods to perform inductive inference (Mitchell, 1997). It can be used to solve classification or regression problems. The learned function can be graphically represented by a tree made of a single root node that can lead to several leaf nodes. An illustration of a decision tree is shown in Figure 4.6. It represents a possible decision tree classifier's output for the decision point B of the model in Figure 4.5. The tree has 3 levels. The first level is the root node and includes all the patients. This node is then split in two nodes based on the input variables "radiography results". The left group is split again using the "age of the patient" and a threshold at 80 years old. In the end, the three has 3 leaves, each having a prediction value for the next event. A decision tree can also be represented as a set of "if-then" rules. The popularity of decision tree is due to its capability to handle noisy data, large volumes of data and any type of variables. Moreover, it has been successfully applied to a broad range of learning tasks in various domains, from medical diagnosis to facial recognition and credit risk assessment for loans.

In our case, similarly to (Rozinat et al., 2009), we decided to use decision trees for 2 main reasons. First, they can handle any type of variables, from continuous values (e.g. length of stay) to binary values (e.g. presence of an infection), categorical values (e.g. medical diagnosis) and discrete values (e.g. number of previous consultations). Moreover, many improvements of decision tree learners have been proposed in the literature to avoid over-fitting, to handle missing value, and to speed up the learning phase. It makes decision trees extremely valuable for practical use. The second motivation for the choice of decision trees is for the clear interpretation of the result. The graphical representation of the tree enables the understanding of the most discriminating input variables that were used to distribute the learning observations in the leaves of the tree. Hence, not only does the model provide reliable predictions, it also gives explanation about the outcome. In the perspective of discovering unknown relationships from the data, this aspect is of major

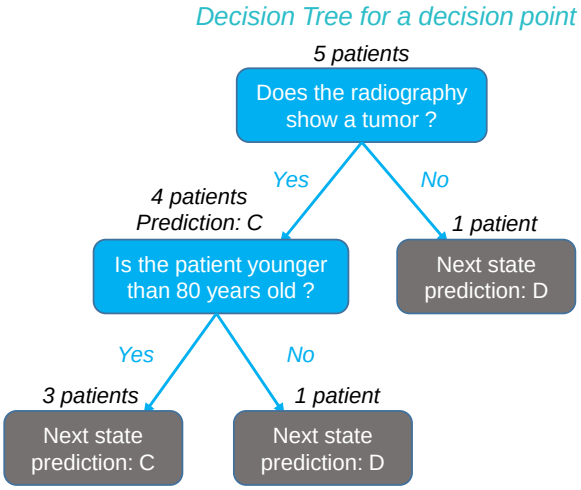


Figure 4.6: Example of a decision tree learner's output for the process model of Figure 4.5

interest for our case study applications. It is also a strong demand coming from the final recipients of our clinical pathway models, health practitioners (hospital staff, drug/device makers). In health-care, relatively few models exist for modeling the long-term (during several years) follow-up of patients (Jensen et al., 2014). Hence, proposing new and explicit classification models at each step of the care process is valuable, even if some trees are rather big (several tens of levels and hundreds of nodes).

A decision tree classifies instances by sorting them down the tree from the root to some leaf node, where each split of branches is made using a rule on a single input variable. In practice (see Chapter 6, case study), we decided to use the CART (Classification and Regression Trees) algorithm. It is very similar to the C4.5 algorithm (an algorithm used to generate a decision tree developed by Ross Quinlan (Quinlan, 1993)), but it differs in that it supports numerical target variables (regression) and does not compute rule sets. Instead, CART constructs binary trees using the features and thresholds that yield to the largest information gain at each node[1]. We implemented a Python program to automatically identify all the decision points of a process model and transform the event log into as many data sets of traces observations. Then, a decision tree learner is run for each decision point. The output tree is converted into a set of if-then rules and is incorporated in the C++ program that we developed to convert a process model into a simulation model (next chapter). These aspects do not fall within the scope of scientific challenges, but they enable an automated reuse of the entire methodology on new data and still bring a valuable contribution for practitioners.

The only obstacle for a perfectly automated re-use of this decision point analysis using a machine learning technique is inherent to any learning technique: it needs to be configured. Decision Trees require at least the set up of 5 parameters. A short description of these parameters is presented in Table 4.1. A relevant tuning of the parameters will avoid the classical pitfalls of machine learning (overfitting, bad accuracy).

Finding the set of parameter values that produces the best (e.g. the most accurate) decision tree is not trivial because of the high number of possible combinations. Instead, in practice, parameter values are adjusted empirically by the user, by trial and error, until the desired level of accuracy is reached. Such approach is time consuming, user dependent and becomes fastidious for models with numerous decision points. In the scope of the present thesis, we focused our efforts on the development of an innovative

[1] reference: http://scikit-learn.org/stable/modules/tree.html

Table 4.1: The 5 main input parameters of a decision tree learner (CART algorithm of scikit-learn)

Parameter name	**Short description**
split criterion	The function to measure the quality of a split (e.g. Gini impurity or information gain).
maximal depth	The maximum depth of the tree (distance between the root and a leaf).
minimum sample split	The minimum number of samples required to split an internal node.
minimum sample leaf	The minimum number of samples required to be at a leaf node.
minimum impurity split	Threshold for early stopping in tree growth. A node will split if its impurity is above the threshold, otherwise it is a leaf.

combination of process mining and machine learning techniques to investigate clinical pathways from large scale databases. The search for an optimal tuning of a decision tree classifier requires advanced skills in the field of machine learning. Still, our goal is to propose a modeling methodology which is highly reusable and as much automated as possible for health data. Although we could not investigate the matter of the optimal tuning of a decision tree learning in this thesis, several existing approaches propose to use classical optimization methods such as genetic algorithm (Camilleri and Neri, 2014; Camilleri et al., 2014; Coroiu, 2016). We also propose a brief discussion about the growing research field of *discriminant analysis using mixed-integer programming* in the perspectives, which proposes complementary approaches to solve typical machine learning problems.

Ensemble methods: the quest for performance

The goal of ensemble methods is to combine the predictions of several base estimators built with a given learning algorithm (e.g. a decision tree algorithm) in order to improve generalizability and robustness compared to a single estimator. The idea of ensemble methods is that "unity is strength". The average of several independent estimators is usually better than any single estimator. Similarly, other ensemble methods (boosting methods) build each base estimator sequentially, so that each new estimator tries to reduce the bias of the combined estimator[2]. The search for the best way to generate a pool of accurate and diverse base learners and for a way to combine their output for a maximal results is an ongoing research field (Zhou, 2012).

Random forest is one ensemble method where base estimators are decision trees[2]. The creation of each tree is different from the the one of a single decision tree learner. Each tree in the ensemble is only built from a sample drawn with replacement from the training set (i.e., a bootstrap sample). In addition, when splitting a node during the construction of the tree, the chosen split is no longer the best split among all features but among a random subset of the features (Breiman, 2001). Random forest are extremely useful in practice because they correct the trend of single decision trees to over-fit the data. Moreover, the general performance of the classifier increases systematically with the increase of the number of trees in the set, until a threshold is reached and no more improvement is achieved. This gain is obtained at the cost of computational power. Finally, Random Forest perform extremely well compared to other supervised learning

[2]reference: http://scikit-learn.org/stable/modules/ensemble.html

algorithms, even for high dimensional data (Caruana et al., 2008). This is even truer when balancing the accuracy, the Area under the Curve and the required amount of time to get the model. The drawback of random forest, and of ensemble methods in general, is their black box effect. No more interpretation of the model is possible and no understandable rules can be derived after the classifier is built. The only available information when predicting the class of a new observation is the *feature importance*. It tells how much each feature contributed to the resulting prediction.

In this thesis, we pursue two objectives. The first is to discover a complete model of clinical pathways that we can simulate. To this end, we propose a methodological framework to automatically perform a process discovery, a decision point analysis and a conversion into a simulation model. Then, new patients are generated through the model. The use of random forest as a supervised machine learning algorithm fits our approach to create a model including all the perspectives. Simultaneously, the second objective is to find the determinant factors that best explain why pathways happen the way they do. For that purpose, a single decision tree learner provides more explanation of the routing choices.

4.5.4 Validation of classification models

In the previous paragraphs, we discussed the definition of a classification problem, the data preparation, and the selection of a supervised learning algorithm. The last step of a classifier construction is the validation. This task is achieved by testing the classification model on new observations. For any data mining project, some of the historical observations are used to learn the classifier whereas others are used to test and validate it. The validation results will determine if the classifier's predictions are good enough on unseen data. If they are not, then the parameters of the algorithm will be tuned until it reaches satisfactory results. This creates an optimization loop where the objective is to maximize the quality of the classifier and the variables are the algorithm's parameters and the choice of features in the data. This validation process is part of a more general scheme, the CRISP-DM reference model (Chapman et al., 2000). CRISP-DM is a comprehensive data mining methodology that provides anyone, from novices to data mining experts, with a complete blueprint for conducting a data mining project.

In our study, creating a classification model is not an end in itself, it is only an intermediate step to enrich the process model. As a result, the validation of this classifier can be carried out in two distinct ways. The first is to validate the classifier independently of the rest of the clinical pathway model. The validation is done exclusively by evaluating the performance of the model on new observations (distinct from the learning observations). Separate validations procedure are run for the classifier of each decision point. The second approach to validate the quality of the learning algorithm can be achieved at a later step of our methodology. It is done when the simulation model is completed. Then, it is done by evaluating the overall behavior of the process model seen as a whole. Global measures about the routing choices can be made after new traces are generated in the simulation model. The simulation model's behavior is compared to the original log, based on several key performance measures. An iterative feedback loop can be implemented to search for the data mining algorithm that would produce the best validation measures. The advantage of this validation approach is to consider metrics related to the entire process (e.g. average number of hospital stays by patient during their pathway), instead of local ones (e.g. number of well classified path for patients after their a given stay X). In conclusion, we can say that the two validation approaches are complementary. In the case study (Chapter 6), we present the results of both.

4.6 Statistical distributions from the event log

In addition to the decision point analysis, we add two other perspectives to a given process model so that it can then be dynamically simulated. These perspectives are the time aspect (within event duration and between events duration) and the probabilistic distributions of trace features.

4.6.1 Characterization of clinical pathway components

In a clinical pathway, **time** is crucial. The elapsed time between the first symptoms and the beginning of a treatment often makes a difference between recovery and death. In a clinical pathway model, we distinguish two measures of time: time spent in a state (hospital stay) and time spent between two states (patient at home). The first one is of the order of a few days whereas the latter can extend from one day to several months. The length of stay or the duration between two stays is heavily stochastic. It depends on the patient, on his/her condition and on other external causes. Based on the event log, we can derive duration measures of the elapsed time between any two moments of the pathway. Historical data shows the variety of possible values. A solution to use historical data in the modeling of the duration phenomenon is to fit these observations with a **random theoretical distribution**. A separate function shall be used for each state and each transition.

The second set of phenomenons that we want to incorporate in the simulation model are patient features. Again, we consider two types of features. First, there are the **process-related features**. Examples of such features, defined for any patient, are "the number of hospital stays since the first hospitalization", "the total time spent since the initial diagnoses" and "accumulated cost of care". These features do not need to be modeled as such because they are mere variables which need to be updated when a new trace (patient) is run in the model (clinical pathway). The second type of features are the **inherent values of each patient**. Examples are "age", "gender", "weight", "height", "level of white blood cell" and "presence of diabetes". Most of these features may be known as soon as the first hospital event of each patient but they are not fixed. They can be modified by the occurrence of specific events (e.g. a bariatric surgery will change the obesity status of a patient) or across time (e.g. age increases and so does the natural death rate). We are interested in knowing the distribution of these features in the historical data. Then, in the simulation model, we will be able to generate new patients who follow this distribution. Both the process-related and inherent features are useful for the analysis of decision points described below because they play an important role in the routing choices. They bring a tremendous added-value to the model compared to simple probabilistic routing choices. Patients' features can also be used to fit several distributions for a given hospital stay. For instance, the random distribution for the length of stay of a knee surgery will not be the same for patients under 25 years old and over 70. Using such splits can improve the level of detail, and thus the quality of the model.

4.6.2 Selection of the best fitting distribution

A way to model a phenomenon such as duration or features distribution is to fit it with a random theoretical distribution. Based on the event log, we assume that we were able to compute a set of historical observations for any variable that we want to model. Not every theoretical distribution is suitable to model any phenomenon. Instead, it is necessary to first look at the characteristics of the historical values. The analysis of input data from such a sample can be split up in three steps (Law and Kelton, 2000, chapter 6):

1. Identifying the appropriate probability distribution (normal, log-normal, beta, Weibull, exponential, etc.)
2. Estimating the parameters of the hypothesized distribution
3. Validating the assumed statistical model by a goodness-of fit test, such as the chi-square or Kolmogorov-Smirnov test, and by graphical methods.

The choice of an appropriate distribution is essential and relies on a thorough study of few basic components of the historical data. The very first step is to determine if the variable has discrete or continuous variable. For instance, the "number of previous medical consultations" is a discrete variable whereas the "cost of care" is continuous. Specific distributions shall be chosen accordingly. Then, one shall look at the presence of a symmetry around the mean value (do lower and higher values spread similarly around the mean). This criteria is rarely observed in actual data. The third element to check is the presence of a lower bound and/or of an upper bound. For instance, any duration is always greater than zero and the risk of being obese is lower than 100%. Finally, the last component to observe in the data is the likelihood for extreme values. For instance, regarding patients' age, it is common to have patients between 0 and 2 years (babies) but much more rare to have patients over 110 years old. Based on the investigation of these 4 elements, one may be able to narrow down the set of possible theoretical distributions. In practice, normal distributions requirements are rarely met by historical data. It makes this choice of distribution very poor compared to many other possibilities, but it is often made for its simplicity.

Once a distribution is chosen (e.g. log-normal or Weibull), the next step is usually to estimate its parameters. Different parameter values will result in different shapes of the curve, which will more or less fit the historical data. A parameter configuration is validated by testing the difference between the theoretical distribution and the data with a statistical test (Law and Kelton, 2000, chapter 6).

In practice, this whole data fitting procedure can be found in most commercial or open-source software related to data analysis or simulation models. In this thesis, we used the Input Analyzer of Arena software. It performs an automatic selection and parameter tuning of the best fitting distribution among tens of the most commonly used. An example of distribution fitting for the variable "age" is shown in Figure 4.7. Based on the characteristics of the data (continuous, lower bounded by zero, no upper bound, asymmetrical), the best theoretical distribution was found to be a log-normal with a parameter value 5.2. We implemented a C++ program to automatically build the observation sets of each variable from the event log. Then, each observation set is automatically processed by the Input Analyzer, returning the best theoretical distribution.

Generation of random variables. Within the simulation model, the generation of a random variable from a theoretical distribution consists in getting one observation of a random variable based on the desired distribution. We do not present the subject any further here as the topic was extensively addressed in (Law and Kelton, 2000). The reader is referred to (Law and Kelton, 2000) to read about the available methods used to generate random variables. These methods are fully widespread in most commercial simulation software.

4.7 Summary and future works

In this chapter, we explored several aspects of process analysis. The main focus was to add several perspectives to a process model, in the form of a causal net, so that we can simulate new traces (patients). We first

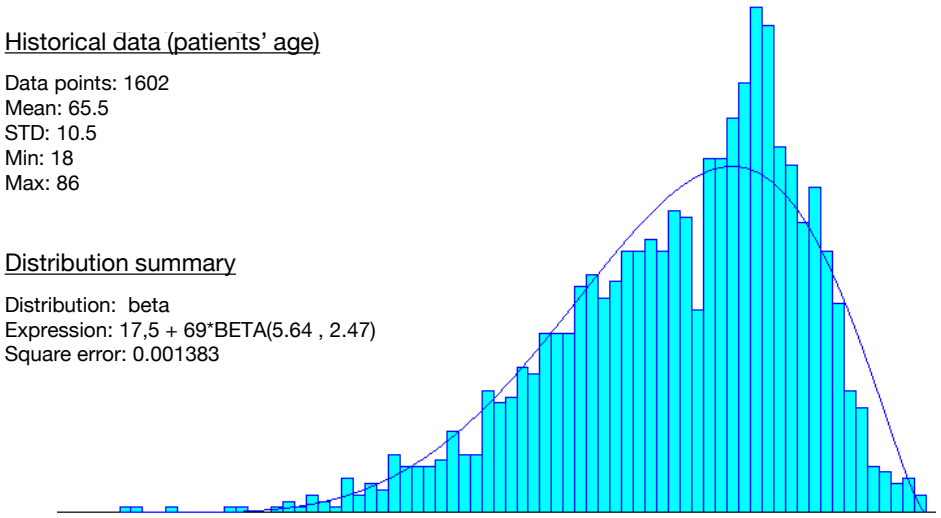


Figure 4.7: Example of a distribution fitting for patient age

formally defined the decision mining problem and the challenges that arise when using the traces from the log to solve it. Our solution to the decision mining problem differs from existing studies where most works use probabilistic models (Markov chain). We introduced an original way to combine sequence alignment using the Needleman-Wunsh algorithm along with similarity measures applied to process modeling, perfect trace generation and classical data mining algorithm to achieve the task of learning reliable decision point rules. We also introduce an already proven way to learn statistical distributions from data, so that we can integrate the time perspective in our process models. The complete methodology was proposed in a generic way to ensure reuse.

4.7.1 Contributions

In this chapter, we propose a *health analytic research toolbox* which encapsulates 3 contributions.

A method to quantitatively compare two sequences of medical events. This step was presented here as an intermediary step before performing classification tasks, but it is also a significant contribution as such to field of medical aid. The alignment of two sequences was only possible after the creation of a meaningful similarity matrix for event classes. Medical events are far too complex to apprehend and to classify for people with no medical knowledge. For that reason, and based on our collaboration with physicians and health data experts, we propose to use a hierarchical structure of data. Such a structure can be built manually by medical experts during data preparation. We also propose to use the 10^{th} revision of the International Classification of Disease, a consensual and already hierarchical reference, to automatically create a hierarchical structure of events. It is extremely valuable when dealing with hundreds or thousands of event classes, in a log where classes were derived from the diagnosis of each hospital stay. Case studies are presented in Chapter 6. Finally, we have an available method to give the closeness score of two sequences of medical events. One sequence can be a guideline pathway that patients are supposed to follow and the second a specific patient that a doctor is following.

A predictive model of the next step in a clinical pathway. At any node of a process model where there is a routing choice, we propose to use a classification algorithm which predicts the path that an entity will follow based on its features. We identified two traditional machine learning algorithms to perform

these tasks. First, we propose to use decision tree learners. Their major advantage is to produce intelligible rules that human being can understand. It helps understanding the factors that most influence each choice of a clinical pathway. That way, we continue enriching our knowledge discovery from health data. The other approach is to use an ensemble method (e.g. Random forest), which will produce better predictions but which lacks interpretation capability.

A complete methodology to produce the two previous points, to which we add the time perspective discovered with distribution fitting, in an generic manner. It enables a straightforward reuse of the approach on new data sets. Even with different data attributes, the methodology stands. In the end, we have an automated way to convert raw data into a process model (Chapter 3), enriched with additional perspectives such as decision points and the time (present chapter).

4.7.2 Limitations

We distinguish two limitations of the presented work. The first one relates to the data quality and the second one to the choice of a machine learning algorithm.

Data quality is a major concern for any person using it. So far, we referred to the "data" as an event log where all the events are labeled, they have a time-stamp and they are related to a process instance. We also considered that other attributes could be available for each event. These attributes can relate to the event itself (e.g. duration, location, number of seized resources) or to the instance (e.g. patient's age). Here, we do not want to expand on the subject of data quality from the collection, the missing values, the noise or the inconsistencies point of views. Such subjects have been treated in the literature (Bose et al., 2013). We want to highlight the quality issue from the data meaning point of view. In our approach, not only did we assume that many attributes were available in the data, but also that these attributes were relevant for studying decision points. This assumption is a strong one, and especially when dealing with health data. A full study could be dedicated to the analysis of raw data so that it becomes useful in making decisions. It includes both classical data preparation (cleaning, sorting, description, type) and a dissection of all the domain-related attributes. For instance, during a hospital stay, patients vital signs can be recorded several times. When a patient is followed for several years and for tens of hospitalizations, the amount of recorded information can be substantial and mostly useless regarding the patient's current treatment. On the other hand, important factors of a clinical pathway, such as a patient's age and his/her medical history, might not be recorded, thus leading to a shaky model of the decision points. To sum up the limitation that we address here, our modeling approach rests upon the availability of relevant data attributes. We did not address this data quality issue much in our methodology here, but we are well aware of it and we propose a thorough description of the data that we use in our case study (Chapter 6).

The second limitation of our approach lies in the choice of a machine learning algorithm to solve the classification problems. We only proposed two supervised learning algorithms (decision trees and random forest), which is insufficient to guarantee the best possible predictions. Based on the classification problem that we defined, a performance benchmark of machine learning algorithms on health data sets would benefit to our methodology. The study could focus on the determination of the most commonly found attributes in hospital event logs (e.g. the list of patient's attributes that are systematically recorded for the medical care). Then, based on several logs of different sizes from real life cases, a standard comparison of the prediction algorithms could be performed. Such a benchmark comparison of machine learning algorithms would help practitioners to select the most appropriate algorithm when studying clinical pathways. It would also contribute to the proposition of a general framework for decision aid in a health-care context.

4.7.3 Future works: medical decision aid

Resource utilization

The present work builds the ground for further enrichment of a process model. For instance, the analysis of the resources at stakes in the process, also called the organizational perspective, is a topic that has already been addressed by the process mining community. For that, it requires that each event of the log is related to a resource (human, facility, machine, etc.). Then, one can study resources utilization rates or the handover among several resources. Here, we modeled clinical pathways at a macroscopic scale. The smallest unit of description is a hospital stay (an event). We think that promising improvements could be brought to our approach by modeling medical procedures within each stay. In the current model of a clinical pathway, we modeled the rules that determine the next stay of a patient based on this stay attributes and on the patient's features. However, each stay could be described as a care process itself composed of several sequential steps (arrival, nurse consultation, medical consultation, imaging, departure). Each step is a sub-event which requires specific resources (human or material). Patients who are hospitalized at the same time would share these resources.

Medical decision aid

Furthermore, during a stay, physicians make several decisions (for instance the choice of operating a patient, of early dismissal or of treatment shift). It would bring the overall model to the next level to successfully model medical decisions and resource dynamics.

DAMIP approaches

Particle swarm optimization techniques for feature selection, coupled with an optimization-based approach called discriminant analysis using mixed-integer programming (DAMIP) is an emergent and promising field of research. It can be applied to identify a classification rule with relatively small subsets of discriminatory factors that can be used to predict resource needs or outcome for treatment. For example, (Lee et al., 2012) proposed a clinical decision tool for predicting patient care characteristics: patients returning within 72 hours in the emergency department using such approach, and demonstrated that optimization achieves better results than classical machine learning techniques. Such approach was also used for modeling and optimizing clinic work-flow (Lee et al., 2016).

Chapter 5

Simulation of clinical pathways

Contents

Abstract

This chapter presents the final methodological step to automatically create simulation models of clinical pathways, starting from raw databases. We introduce an automatic procedure to convert a process model, discovered with process mining, into a simulation model. The concept of state chart is used and enriched to incorporate the distinctive features of health-care processes into the model. Hence, we introduce a new subclass of state chart, called "Clinical Pathway State Chart", with the required properties to simulate a cohort of patients while taking into account the pathways discovered using process mining techniques presented in Chapter 3 and the features found using the health analytic toolbox presented in Chapter 4. The clinical pathway simulation model is used to simulate new patients' sequence of events. The resulting model is validated by comparing key performances indicators with the historical data. Finally, we use the model to perform sensitivity analyses and what-if scenario evaluations. The simulation process is automated and can be used with any process model and any set of data as defined before.

5.1 Introduction

In Chapter 4, we proposed a new approach to discover clinical pathways (CP) from a national hospital database using process mining. The objective was to create the most representative process model of the event log under a constraint on the size of the model. In the literature, CP analysis of recorded data was mainly done using either data mining or process mining techniques. Such approaches receive an increasing attention in the field of medical informatics. The next step of this research consists in (i) proposing a model that can be executed using simulation and (ii) testing what-if scenarios. Scenarios can be related to various decisions, such as a change in the medical treatment of certain patients, the launch of new medical devices supposed to be more effective to cure certain diseases, or a change in hospital activities financing.

This chapter provides a comprehensive methodology to analyze and simulate such CPs. It uses an existing process model discovered from an event log and a set of features found using the health-care analytics toolbox. For that, we propose (i) a new procedure to automatically build a simulation model of patients' CP from an event log of hospital stays, and (ii) a new subclass of state charts called "Clinical Pathway State Charts" (CPSC) to capture all the required material to efficiently simulate and evaluate the performances of any clinical pathway.

Such methodology may be applied using any database as data input and may be applied for any cohort of patients, which constitutes a significant scientific contribution. Simulation of clinical pathways brings new knowledge and allows the evaluation of scenarios through design of experiments. Along with the simulation model, an automated set of analyses can be performed, including formal validation procedures and sensitivity analysis. The latter provides immediate insights on the variables of the case study which have the highest impact on key performance indicators. Target users of our approach are numerous:

- **hospital managers**: predict the results of investments in new care services or management strategies;
- **health-care practitioners**: test the relevancy of new treatments at certain steps of the care pathway of the patients under study;
- **pharmaceutical firms**: extrapolate the impact of a new drug or a new medical device on the patient care pathway by taking into account the cost of hospital stays.

The goal of modeling and simulation is to reproduce the behavior of the original traces found in the log. Hence, we want to reproduce the sequencing of events, the random or controlled aspects of a path choice or the dynamic evolution of an instance's features. So far, based on the definitions of the previous chapters, we do not consider resource sharing in our model. Thus, patients do not compete with each other for resource access. Indeed, our main application case study (see Chapter 7, case study) is the analysis of clinical pathways at a macroscopic point of view. We assume that a patient can always find an hospital which meets its need (operating room, medical specialist, imagining technology, etc.) as he/she can move on a vast territory (national scale). Our description of clinical pathways is not limited in time and space. It means that we study the sequencing of hospital events for millions of patients over several years. The delay between two events ranges from days to months and the total follow-up duration often reaches 5 to 10 years. At this scale, we assume that a hospital event is the smallest descriptive unit. We do not study the care process within an hospital stay, with hour by hour steps (MRI, passage in the emergency room, nurse care, medical care, etc.). Many research works can be found in the literature on the topic of modeling and analyzing clinical pathways within a single hospital stay, from the entry point to the discharge (Lucidi et al., 2016; Franck et al., 2015; Perdomo et al., 2006; Augusto and Xie, 2014). In Section 5.8, we discuss the merging into a single model of both high-level and daily management perspectives of clinical pathways.

The remainder of this chapter is organized as follows: a literature review related to the automatic creation of a simulation model and to its performance evaluation is given in Section 5.2. We then introduce the concept of clinical pathway state charts, a new subclass of state charts, that we use for the simulation in Section 5.3. We also present a methodology for the automatic conversion of causal net into CPSC. Then, Section 5.4 presents the simulation procedure that we perform to generate output measures and Section 5.5 explains how we validate the model with these measures. Sections 5.6 and 5.7 are dedicated to the use of the simulation model (sensitivity analysis and what-if scenario). Possible extensions and perspectives are discussed in Section 5.8.

5.2 Literature review

The main motivation for an automated creation of simulation models is that most simulation models are handmade models. Handmade models are built based on available documentation, observations of the modeler and on interviews of experts. This is a time consuming approach and a partial view of the processes. The perception of the actual process is influenced by the experience of the human studying it. It introduces a bias which may impact the key performance indicators and the overall results of the simulation study. To avoid these biases, the idea of integrating various process mining results to automatically generate a complete simulation model was first done by (Rozinat et al., 2009). Such idea of a complementary input for simulation models was then advocated by (Martin et al., 2014b) and (Martin et al., 2014a). In (Rozinat et al., 2009), the authors focus on the validation of a simulation model (whether generated or hand-made) since its quality is crucial for drawing conclusions from a simulation run. Finally, they highlight the challenges that are faced when discovering simulation models from event logs. It includes creating not too complex models to have usable results, adding other perspectives to the flow perspective (e.g., patients' features or human resources), and adjusting the model for real-time simulation. They show an example of their modeling methodology using Petri Net as the formal representation of their process models. It allows for a strong formalism of the modeling framework, but it lacks the capacity of dealing with very heterogeneous and eccentric behaviors.

The work of (Zhou et al., 2014) describes a case study in health-care where process mining and sim-

ulation are used together. They use the fuzzy miner algorithm for process discovery (Gunther and van der Aalst, 2007). They specifically study the pathway of patients during a single hospital stay, starting from the admission to the release. The process included the key steps of the process, such as *check in*, *medical consultation and diagnosis*, *waiting* and *check out*. Nevertheless, if a patient is readmitted later, he/she is considered as a new patient starting the process for the first time. Our work is driven by health-care case studies with numerous patients. For that reason, we need a flexible modeling framework, which is the *CPSC* formalism presented in the next section. Our approach intends to show the clinical pathway of patients during several years. A patient is followed over a long period of time and across a national territory. It leads to a complete description of care pathways at a macroscopic view. In (Augusto and Xie, 2014), a modeling and simulation framework dedicated to health-care systems was proposed. The main contribution was an automated procedure to convert a set of UML models to an actionable Petri net. The framework, named MedPRO, was successfully applied to various case studies such as the operating theater, the hospital pharmacy logistics or a neuro-vascular service organization. In this thesis we propose an automated procedure; however, we also automatically create process models using process mining instead of relying on experts with hand-made models in the MedPRO approach.

The creation of a simulation model is not an end in itself. The real purpose is to run the model and to evaluate the behavior of the system. For that, a design of experiment must be defined. The experimental phase is twofold: preliminary experiments are dedicated to the validation of the simulation model, and once a model has been validated, new experiments can be driven to fully exploit the benefits of the model. It includes a sensitivity analysis of all the input variables and the evaluation of new scenarios. The validation of a simulation model is essential to ensure its practical utility. A model does not need to be absolutely valid, instead it has to be valid regarding a specific purpose (Sargent, 2011). Validation techniques are multiple and no approach outperforms the others systematically. The choice of a validation technique essentially relies on the available information related to the model (several sets of historical data, experts' opinions, graphical visualizations of the process, etc.) (Sargent, 2011). In this thesis, we propose a rigorous and automated procedure to validate the simulation models using the information related to the traces.

Once a model is validated, it can finally be put at work. A simulation model is a powerful tool that allows to test a large variety of configurations in order to see their impact on the model's outputs. The impact of a new configuration can be studied in two ways. The first is to perform a sensitivity analysis and the second is through what-if scenarios evaluation. A sensitivity analysis (SA) is the study of how the variations of input parameters impact the model's outputs. SA approaches are either local or global (Maljovec et al., 2016). Local SA are dedicated to the study of a model's responses over the variations of a single parameter while other parameters remain fixed (Saltelli et al., 2008). On the other hand, global SA study the output changes when all the parameters vary simultaneously (Saltelli et al., 2008). Global SA are more complex to handle because of the large number of possible combinations when dealing with numerous parameters and because of the interpretation of the results. Here, we only focus on local SA. Moreover, in the continuity of our methodology presented so far, we propose a procedure for the automatic generation of a simulation model SA. The process of performing a sensitivity analysis is often hand-made and domain dependent (Wu and Mortveit, 2015). It requires (i) to identify variables of the model whose impact on the outcomes has a relevant meaning, (ii) to determine the range of possible values that these variables can take, and (iii) to analyze the significance of the discovered relationships. In (Wu and Mortveit, 2015), the authors introduced a general framework for experimental design and sensitivity analysis of simulation models. They generalized the concept of uncertainty quantification and SA for simulation models from different domains. The most important part of their work is the development of a tool that allows the user to perform

SA without having any knowledge about the model specifications and formalism. However, it still requires the user to choose a SA technique and the variation ranges. To the best of our knowledge, proposing an automated procedure to generate input-output sensitivity analyses from well-formalized simulation models, while being domain independent, is innovative and has not been done.

The final step of the model utilization is the evaluation of what-if scenarios. This aspect of simulation is extremely documented in the literature as it is the first motivation to create simulation models. Almost every article about simulation model applied to a case study deals with scenarios evaluation. Examples of scenarios evaluation using a simulation model in the health-care domain can be found in (Augusto et al., 2015; Augusto and Xie, 2014; Franck et al., 2015; Pehlivan et al., 2013b; Perdomo et al., 2006).

5.3 State chart definition and conversion framework

This section presents the formalism that we use to define a simulation model and our conversion procedure to transform a process model into it. A state chart (also called a finite-state machine) is a mathematical model that describes the behavior of a system. It is an excellent way to model the process steps of an entity, the activity of a resource or the coordination of several entities. A state chart is made of states and transitions. Then, at any moment, any instance of the state chart can only be in one state at a time. This state is called its *current state*. The instance can change from a state to another when a specific condition is met, this is called a transition. State charts can be used to model systems in a large variety of domains and problems.

5.3.1 State chart definition

To simulate the clinical pathway of new patients, we use the general concept of state charts. It includes the definition of states, transitions, probabilities of activating a transition and a state duration. We then enrich this state chart definition with two new concepts: wait-states and care-states. It allows us to better model the behavior of a clinical pathway. Eventually, we introduce a new subclass of state chart that encapsulates all the specific features of a clinical pathway and simulates it.

Definition 4 (State chart)**.** *A state chart (SC) is a 4-tuple* $M = (S, V, \zeta, \tau)$ *where* $S = \{s_1, s_2, \ldots, s_n\}$ *is a finite set of states,* $V \subseteq (S \times S)$ *is a finite set of transitions,* $\zeta : V \to [0,1]$ *is the probability of activating a transition, and* $\tau : S \to \mathbb{N}$ *is the time spent in a state.*

Our goal is to use a state chart to model the clinical pathway of a patient. A patient is modeled using the concept of entity, where each entity is defined by a set of features and an active state.

Definition 5 (Entity)**.** *An entity is a 3-tuple* $u = (M, f, s)$*, where* $M = (S, V, \zeta, \tau)$ *is a SC,* $f = \{f_1(u), ..., f_x(u)\}$ *is a set of assigned values for attributes from F (the set of trace's attributes, Chapter 5) and s is its current state,* $s \in S$*.*

Two types of states are defined to distinguish states related to a stay in a hospital and states related to a waiting period between two hospital stays.

Definition 6 (Care-state)**.** *A Care-state is a 2-tuple* $s^c = (l, B)$ *where l is a unique label and, with* $n \in \mathbb{N}^*$*,* $B = \{(f_1, v_1), \ldots, (f_n, v_n)\}$ *is the list of entities' features* $\{f_1, ..., f_n\}$ *to be updated in this state with new values* $\{v_1, ..., v_n\}$*. It includes at least a state-related cost that will be used as a performance indicator in the simulation.*

A care-state is related to a change in a patient's health condition and requires a medical response process. During this process, the entity's attributes may change according to the set B.

Definition 7 (Wait-state)**.** *A wait-state is a singleton $s^w = (l)$ where l is a unique label.*

Finally, we propose a new subclass of state chart to describe the clinical care pathway of patients, denoted Clinical Pathway State Chart.

Definition 8 (Clinical Pathway State Chart)**.** *A Clinical Pathway State Chart (CPSC) is a 6-tuple $CPSC = (S, V, \zeta, \tau, p, q)$:*

1. *$S = S_w \cup S_c$ where S_w is a finite set of wait-states and S_c is a finite set of care-states*
2. *$V \subseteq (S_c \times S_w) \cup (S_w \times S_c)$ is the set of transitions (vertexes) of the CPSC*
3. *ζ gives the probability of activating each transition given a state s and a set of features $F = \{f_1, ..., f_x\}$:*

$$\begin{array}{rccc} \zeta & : & S \times F & \rightarrow & V^{|V|} \times [0,1]^{|V|} \\ & & s \times \{f_1, ..., f_x\} & \mapsto & \{v_1, ..., v_{|V|}\} \times \{p_1, ..., p_{|V|}\} \end{array} \text{, with } \sum_{i=1}^{|V|} p_i = 1$$

4. *$\tau : S \rightarrow \mathbb{N}$ is the time spent in a state.*
5. *$p : S \rightarrow [0,1]$ is the probability that the simulation starts at a given care-state, $\sum_{s \in S} p(s) = 1$*
6. *$q : S \rightarrow [0,1]$ is the probability that the simulation stops after reaching a given state*

A Clinical Pathway State Chart (CPSC) is a state chart whose underlying graph is a bigraph (bipartite graph) and which has the 5 following properties:

1. The two types of states of the bipartite graph are care-states and wait-states. By definition, two states of the same type cannot be linked by a transition. E.g., after the care-state "surgery during an hospitalization for heart failure" there must be one of the 3 following wait-states, "die", "recover partially" or "recover fully".
2. Each wait-state has exactly one input transition and one output transition: the probability of activating the output transition of a wait-state is always equal to 1.
3. The probability of activating a transition in a clinical pathway model depends on the current state of an entity, but it also depends on this entity's features.
4. The sum of all the output transitions probabilities of a care-state is equal to 1 (definition of ζ).
5. In an actual clinical pathway, any state can be the starting point or the stopping point of a patient care process (the condition of an ill patient can be much advanced or not when he is seen for the first time, and a patient may die at any time). A *CPSC* needs two further components to capture such behavior:

According to Definition 8, a care-state may be followed by $\{0; n\}$ wait-state(s), meaning one of n options will be realized according to the probability of the mapping function ζ. A wait-state is always followed by exactly one care-state: the probability of the transition between a wait-state and a care-state is equal to 1.

5.3.2 Conversion procedure

An algorithm for the automatic conversion of a process model into a Clinical Pathway State Chart was developed. In this section, we present the conversion procedure step by step. The conversion is done sequentially: first, the structure of the model is created. Each node and arc of the process model is converted into a specific state of the *CPSC*. Then, for each newly created state, a decision point classifier and statistical distributions are added. In the end, we obtain a fully operational *CPSC*.

First part: creation of the state chart's structure
Input: A process model $PsM = (N, E)$ composed of a set N of nodes and a set E of arcs
Output: A state chart $SC = (S, V, \zeta, \tau)$

1. Initialization: Let S be the set of states and V be the set of transitions. S and V are empty.
2. For each node $n \in N$, add a care-state s_n^c to set S.
3. For each arc $e \in E$ having $n \in N$ (resp. $m \in N$) as origin node (resp. destination node): (i) add a wait-state s_e^w to set S; (ii) add the two transitions $\{(s_n^c, s_e^w), (s_e^w, s_m^c)\}$ to set V. The number of transitions in V is always equal to twice the number of edges in E.
4. Let ζ, τ all be the null function on their domain of definition.

To execute the resulting state chart *SC* in a simulation model, we need to define the functions ζ (transition probabilities) and τ (duration in states). We also need to add two probabilistic distributions to our newly created state chart: p (starting states probabilities) and q (stopping probabilities in states).

Second part: probabilities and state duration of the CPSC
Input: $SC = (S, V, \zeta, \tau)$ + decision point classifiers (Chapter 5, section 7) + statistical distributions (Chapter 5, section 8)
Output: A complete Clinical Pathway State Chart $CPSC = (S, V, \zeta, \tau, p, q)$

1. For each care-state $s_c \in S$, build a decision point classifiers C_{s_c} as defined in Chapter 5, then $\zeta(s_c, \{f_i\}) = C_{s_c}(s_c)$. For each wait-state $s_w \in S$, as s_w only has one output transition v_w by definition (probability of v_w being triggered is 1):
$\zeta(s_w, \{f_i\}) = \{v_1, \ldots, v_{(w-1)}, v_w, v_{(w+1)}, \ldots, v_n\} \times \{0, \ldots, 0, 1, 0, \ldots, 0\}$.
2. For each care-state s_c and each wait-state s_w of S, fit the historical observations of state duration with the best possible random distribution (Chapter 5, Section 8). The result is τ.
3. For each care-state $s_c \in S$, compute the probability of being the first state of a CPSC instance based on the historical data: $p(s_c) = (nb\ of\ traces\ who\ started\ in\ s_c)/(nb\ of\ traces\ in\ the\ log)$. For each wait-state $s_w \in S$, $p(s_w) = 0$ (an instance cannot start in a wait state).
4. For each care-state $s_c \in S$, compute the probability of being the last state of a CPSC instance based on the historical data: $q(s_c) = (nb\ of\ traces\ who\ finished\ in\ s_c)/(nb\ of\ traces\ who\ had\ s_c)$. For each wait-state $s_w \in S$, $q(s_w) = 0$ (an instance cannot stop in a wait state).

Using such a procedure, the resulting state chart $CPSC = (S, V, \zeta, \tau, p, q)$ is a Clinical Pathway State Chart as formally defined in Definition 8. This model can be executed to run new instances of its process. The following presents an example of the conversion procedure used on a small process model.

Example. We consider the process model given in Figure 5.1. This process model is formally defined as a causal net by the sets $N = \{A, B, C, D\}$ and $E = \{e_1, e_2, e_3, e_4, e_5\}$ (4 nodes and 5 edges). The first part of the conversion procedure produces the state chart $CP = (S, V, \zeta, \tau)$ presented on Figure 5.1 with:

- $S = \{s_A^c, s_B^c, s_C^c, s_D^c, s_1^w, s_2^w, s_3^w, s_4^w, s_5^w\}$ where state s_i^c is a care-state related to node i and state s_j^w is a wait-state related to edge e_j. Care-states refer to hospital stays and wait-states to waiting between two stays.
- $V = \{(s_A^c, s_1^w), (s_1^w, s_B^c), (s_B^c, s_2^w), (s_2^w, s_C^c), (s_B^c, s_3^w), (s_3^w, s_D^c), (s_C^c, s_4^w), (s_4^w, s_D^c), (s_D^c, s_5^w), (s_5^w, s_D^c)\}$
- ζ and τ are initialized as null (they are defined at the second step).

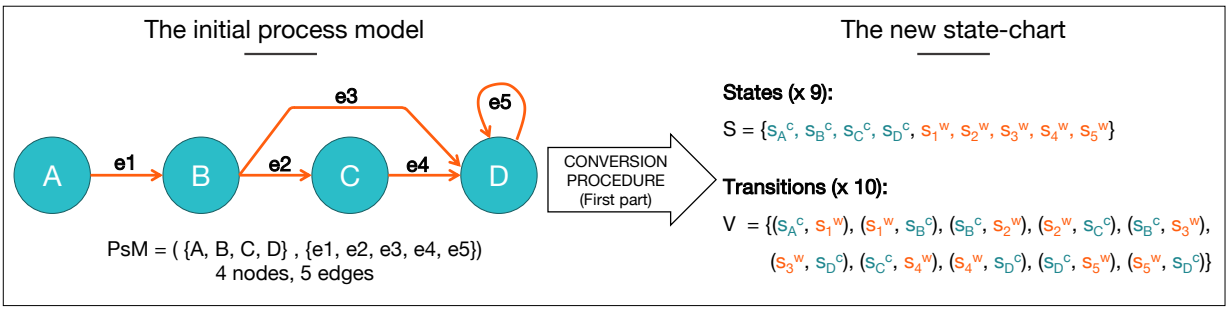


Figure 5.1: Illustration of the first step of the conversion procedure on a simple process model (4 nodes, 5 edges)

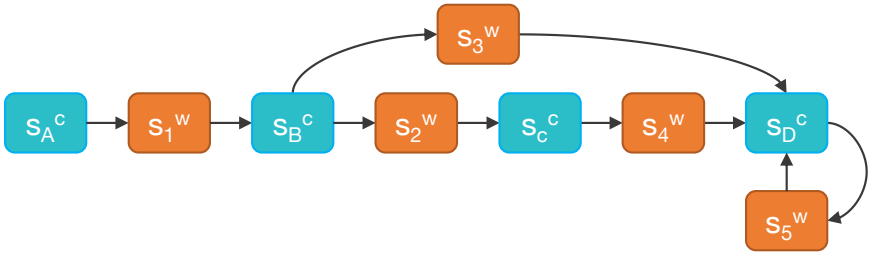


Figure 5.2: Graphical representation of a CPSC: care-states (blue), wait-states (orange) and transitions (black)

The second part of the conversion procedure produces a complete Clinical Pathway State Chart $CPSC = (S, V, \zeta, \tau, p, q)$. An example of the computation of ζ, τ, p and q is presented on Figure 5.3. The computation of decision point classifiers, state duration, start and stop probabilities is performed using the health-care analytics toolbox described in the previous chapter.

5.4 Simulation setting up

This section presents the simulation procedure used to run a newly converted simulation model. Such a model can now be used to simulate new enhanced traces, as defined in Chapter 4.

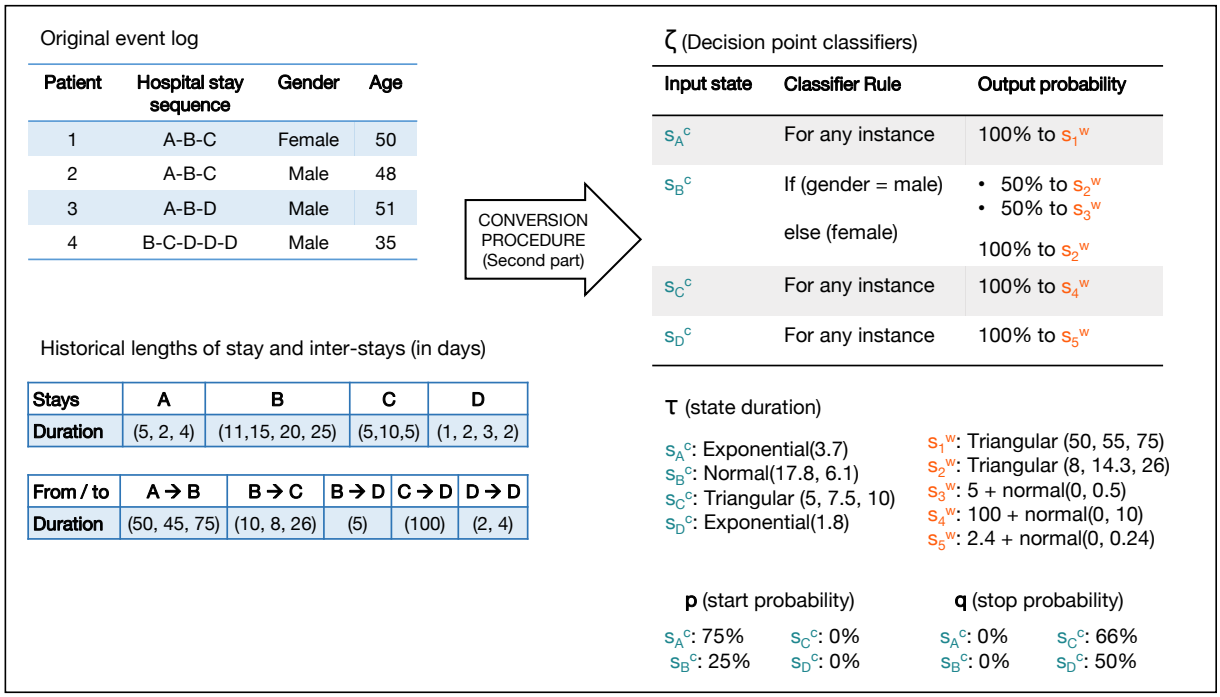


Figure 5.3: Illustration of the second conversion step based on a simple event log (4 traces, 4 event classes)

5.4.1 Simulation procedure

As explained in the introduction of the present Chapter, the simulated entities do not interact with each other (e.g. no resource sharing among patients and no contagious disease are modeled). Hence, each entity can be simulated independently and the same procedure is repeated to simulate an entire cohort with numerous patients. As entities are independent and normally distributed, our experiment meets the criteria of a Monte-Carlo experiment (Raychaudhuri, 2008). Monte-Carlo methods are a broad class of numerical simulation models that are used to solve difficult problems with randomness. The general concept of Monte-Carlo methods can be summarized in 4 steps:

1. Define a domain for the inputs
   Example: a CPSC defines a domain of possible sequences of care
2. Generate inputs randomly from probability distributions over the domain
   Example: Each patient is an input with specific values drawn from distributions
3. Compute the deterministic output for each input
   Example: each KPI is measured for each simulated entity
4. Aggregate the results
   Example: compute the average and the error of each KPI

The simulation procedure for a single entity is described in Algorithm 3. First, a new entity is created (line 2 of Algorithm 3). Its initial values of features (line 3) and its initial state (line 4) are drawn from the right random distributions. Then, the procedure computes the time spent in the current state (line 7) and the next state based on the classifier (line 11). This is repeated until a stopping criterion is reached (line 10). Three stopping criteria are used. The first possibility is when a state with no outgoing transition

is reached (line 18), the second is the natural probability that a sequence stops within a given care-state (both possibilities are included in the function q of the *CPSC*), and the third is when an entity's sequence reaches the threshold of the maximal number of events (line 10). This threshold is set empirically by looking at the size of the longest trace sequence of the original log. The threshold avoids to have extremely long simulated traces that would be unrealistic (and probably due to a repeated loop of events). When an entity enters into a new care-state, its features are updated accordingly (line 15). For instance, if the entity has an attribute "health condition", it can be updated to "good" after a successful post-surgery follow-up with a favorable advice of the doctor, or to "bad" after an hospitalization for heart failure after a passage through the emergency room. Similarly, other attributes such as costs, age or medical history are updated. In addition, in any state of the model, the entity's time-span is incremented of the duration spent in this state (line 14).

Algorithm 3 Simulation procedure of a new entity

Require: a Clinical Pathway State Chart $CPSC = (S, V, \zeta, \tau, p, q)$ and the maximal threshold on the number of states an entity may have $M \in \mathbb{N}^*$

1: **Step 1 – Initialization**
2: Create a new entity $u \leftarrow (f, s)$, $s \leftarrow \{\emptyset\}$ (current state), $f \leftarrow \{\emptyset\}$ (features)
3: For each feature, draw a value x from the random distribution and $f \leftarrow f + \{x\}$
4: Draw a random number x in $[0,1]$, $p^{(-1)}(x)$ gives the matching starting state s_0: $s \leftarrow s_0$
5: $TS(u) \leftarrow 0$ (time spent by u in states)
6: $NB(u) \leftarrow 1$ (number of visited states)
7: Draw a duration value x from state s random distribution: $TS(u) \leftarrow TS(u) + x$
8: Draw a random number $x \in [0,1]$, to compare with the stopping probability of q
9: **Step 2 – Simulate a pathway**
10: **while** $(NB(u) < M)$ and $(x > q(s))$ **do**
11: compute the next state s_{next} of u with $\zeta(s, f)$
12: $s \leftarrow s_{next}$, the new state
13: $NB(u) \leftarrow NB(u) + 1$
14: Draw a duration value x from state s random distribution: $TS(u) \leftarrow TS(u) + x$
15: **if** $s_{next} \in S_c$ **then**
16: update the entity's features: $f \leftarrow B_{next}(f)$
17: **end if**
18: Draw a random number $x \in [0,1]$, to compare with the stopping probability of q
19: **end while**
20: **Step 3 – End of simulation**
21: return u

5.4.2 Simulation output

The above procedure describes the simulation of a single entity. It represents one observation of a random process. Our objective is to evaluate the global performances of the clinical pathway that we modeled. For that, we first need to define a set a key performance indicators. Then, we will simulate a large number of patients as described in the Algorithm 3. As patients are independent and equally distributed, the procedure meets the Monte-Carlo simulation criterion (Raychaudhuri, 2008). The key performance indicators are first measured for each patient and they are then averaged on the entire cohort to be significant. Finally, we will be able to provide a standardized simulation report.

Key performance indicators

The key performance indicators (KPIs) are used for 2 purposes. It will be used to validate our simulation model behaviors and to test new situations. For the validation, we compare the indicators measured in output of the simulation model with the same measure directly derived from the original log. The validation of the model is unavoidable before running further tests. Indeed, a model is useless if it does not reproduce the actual system behavior.

Many KPIs can possibly be measured at the end of a simulation run. Most of the time, KPIs are specifically chosen for each case study. For instance, for the care process of patient with a lung cancer, the time between the diagnosis and the death is of major interest. In the case of the follow-up process after a spinal surgery, the percentage of patients who develop an infection is more relevant. However, based on the definition of a *CPSC*, we propose a set of generic KPIs that can be used for any case. For each simulated entity of a $CPSC = (S, V, \zeta, \tau, p, q)$, we measure at least the following:

- **KPI #1:** The total (cumulative) time spent in care-states
- **KPI #2:** The total time spent in wait-states
- **KPI #3:** The number of visited care-states
- **KPI #4:** The number of times that state s_i was visited, $\forall i \in S$

Remark 12. We do not need to measure the number of visited wait-states because it is equal to the number of visited care-states minus 1.

Remark 13. KPI #4 represents several measures. There are as many KPIs as there are states in S.

These KPIs are measured for each simulated entity. Then, to evaluate the global behavior of the cohort, we compute the average value of each KPI. Hence, KPIs #1, #2, #3 and #4 become (i) the average time spent in care-states by a patient, (ii) the average time spent in wait-states by a patient, (iii) the average number of visited care-states by a patient and (iv) the average number of times that state s_i was visited by a patient. In addition, we add another KPI for the global evaluation of the cohort:

- **KPI #5:** The average number of different entities that visited state s_i at least once, $\forall i \in S$

KPI #5 is slightly different from KPI #4. The difference relies in the fact that an entity can be in a given state s_i more than once during its clinical pathway. We are interested in knowing the total number of times that a state was visited (KPI 4) and the number of different entities that visited it (KPI #5). The difference between KPI #4 and KPI #5 is not relevant for all the states, it depends on each case study. In practice, some states were observed to be recurrent states. For instance, during the care process of patients with a cancer, the chemotherapy sessions are repeated multiple times over several weeks. So, in a cohort of patients with diabetes where the goal is to study the clinical pathway related to the management of diabetes, few patients may have a cancer and may undergo chemotherapy. These patients will be the only to have tens of chemotherapy sessions. On a global view of the cohort, it may appear that each patient has on average one chemotherapy session. However, this would not be true as only a small part of the patients gather all the sessions. This is why KPIs #4 and #5 bring different information about the care process.

Finally, we want to emphasize the need for more KPIs to be defined for each case study. We will see examples in the Chapter 7 (case studies). Still, the set of 5 generic KPIs presented here already captures well the behavior of a cohort simulated in a *CPSC*. It describes the main aspects related to the time spent in states and to the trajectory of entities in the model (number and types of the visited states).

Simulation stochasticity and confidence interval calculation

The simulation of a single entity's clinical pathway represents one observation of a random process. At the different steps of the simulation (starting state, features initialization, decision points routing, etc.), random draws were made from theoretical distribution. Our objective is to get a good estimation for each KPI. For that, we repeat the simulation procedure for a large number of entities. The KPIs are estimated as the empirical mean on the sample of simulated entities.

For a given sample of n independent and normally distributed entities, the estimation $\bar{x}$ and the error ε of a KPI x at a confidence level $(1-\alpha)$ are:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \varepsilon = t_{\alpha}^{n-1} \times \frac{\bar{\sigma}}{\sqrt{n}} \tag{5.1}$$

where $\bar{\sigma}$ is the empirical standard deviation of the sample and t_{α}^{n-1} is the t-distribution value of the Student law with $(n-1)$ degrees of freedom and a confidence level of α (see Appendix B for the table of t-distribution values). By the law of large numbers, the error on the KPIs' estimation converges to zero as the number of simulated entities grows. This is the main advantage of the approach: in theory, the method always converges. In practice, the available computation time is often a limitation. Hence, the final results is a balance between an affordable computation time and an acceptable level of error.

Simulation results summary

At the end of a simulation run (i.e. the simulation of numerous entities), a report with the results is generated. It includes the estimated mean and the error of each key performance indicator chosen by the user. It also includes detailed information about the simulation run elements (e.g. number of simulated entities, memory usage, elapse time, etc.). Such reports are useful for stakeholders to understand the output of the simulation model and should be carefully built upon the relevant key indicators for each of them.

5.5 Validation of the simulation model

The concern of whether or not a model and its results are "correct" is addressed through the model **verification** and **validation**. Verification is the process of determining that a model implementation and its associated data accurately represent the developer's conceptual description and specifications. Verification addresses programming-oriented issues. It ensures that the conceived model was successfully transposed in a computer language. A detailed discussion about verification is out of the scope of the present work.

Validation is the process of determining the degree to which a simulation model is an accurate representation of the real world, specifically from the perspective of the intended uses of the model (Schlesinger, 1979). Model validation is critical in the development of a simulation model because a model that cannot be validated is simply useless. There is no need for a model that cannot reproduce the real system, and if it is easy to demonstrate that a model is false, it is much harder to ensure that it is reliable. Figure 5.4 shows the relations between a real system, a conceptual model and a simulation model. The verification is made between the conceptual model and the simulation model, whereas the validation that we are interested in here is the operational validation between the simulation model and the real system.

There is no generic method to determine how a model should be validated (Sargent, 2011). Each model is unique and must be validated accordingly. A model is always built regarding a specific application and it must be validated with respect to that objective. It requires defining the model output variables related

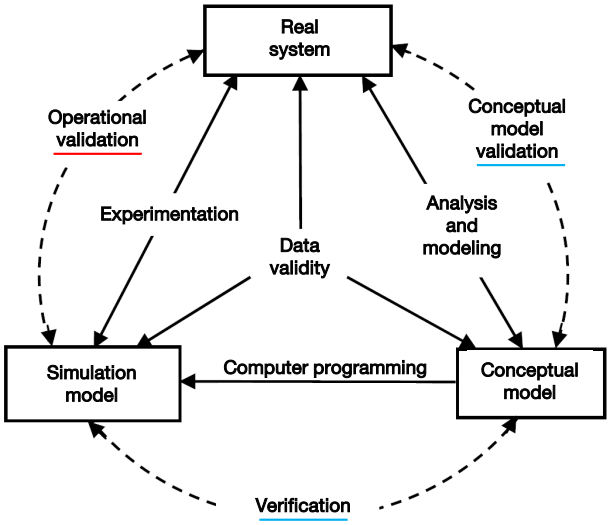


Figure 5.4: Simplified version of the modeling process and of the verification-validation process (Sargent, 2011)

to this application. This is exactly the role of the key performance indicators (see Section 5.4.2). For each indicator, we need to define an acceptable range of possible values. If the results are within this range, the model will be validated. The choice of a range is often based on the actual value of the real system plus or minus a margin of error.

5.5.1 Validation techniques

There are several ways to validate a simulation model. In (Sargent, 2011), the author summarizes and presents 15 of the most commonly used validation techniques. These techniques can be used independently or combined together. They show the variety of aspects that validation can take. The reader is referred to each technique's specific literature for more details. All 15 validation techniques are detailed in Annex C. In this work, we focus on two techniques in particular:

1. *Face Validity*: Individuals knowledgeable about the system are asked whether the model and its behavior are reasonable.
2. *Historical Data Validation*: If historical data exist, part of the data is used to build the model and the remaining data are used to test whether the model behaves as the system does.

The choice of *historical data validation* is motivated by the fact that it is one of the methods that can be done automatically and which does not need any human intervention. Indeed, our leitmotiv through this thesis is to propose methods that are generic and reusable on any data set. Comparing the output KPIs of our simulation model to the historical value found in the event log is the best way to fully take advantage of the database. So far, in our methodological process, aggregate measures about the studied process were not used. Using historical data validation is a fast and objective way to validate the model. Moreover, several validation techniques enlighten the need for the validation of the model's assumptions. Here, the modeling hypothesis are those of the process mining algorithm that was used to discover the model. The strong hypothesis were the choice of a model notation (the causal net) and of a replayability measure which can

assess if a model is representative of the event log. These two aspects were discussed in detail in Chapter 4. Then, between the discovered process model and the simulation model we applied several modeling choices (perfect trace generation, trace alignment, statistical distribution, conversion procedure) that brought their part of new assumptions. Each choice was motivated and explained regarding a conceptual or a technical challenge. Our goal is not to validate each step independently, but to evaluate the final behavior of the resulting model. That is why historical data validation and a sensitivity analysis appear to us as the most appropriate way of doing so.

In addition to the historical data validation of the KPIs, we are well aware of the potential benefit of using *face validity*. Experts with knowledge about the actual process can provide efficient feedback about the model behavior. It is even truer in the health-care domain where the understanding of disease evolution or medical decisions is difficult. In the present work, for the application of our methodology on health data, the advice of medical experts were solicited at two specific points of the analysis: first, at the very beginning when preparing the data (event labeling and definition of hierarchical classes), and then for the interpretation of the sensitivity analysis. Indeed, as presented in the next section, the results of the sensitivity analysis (i.e. the test of numerous possible configurations) would require an important amount of data from the real system to be confronted and validated. Instead, the opinion of experts from the application field can provide thoughtful insights about these results.

5.5.2 Model validation and calibration

Validation is usually achieved through the calibration of the model. A model calibration is an iterative process which is done by comparing the model to the actual system behavior and by using the discrepancies between the two to improve the model. This process is repeated until model accuracy is considered acceptable enough. Figure 5.5 presents a schematic view of such a calibration process. New values of input parameters are tested until the validation is satisfied.

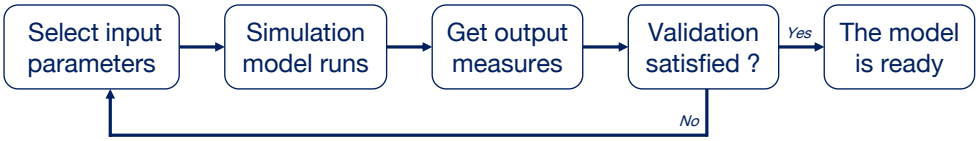


Figure 5.5: Schematic view of a simulation model calibration process

Most models have many input parameters and several output results, so finding the set of parameter values that provides the most accurate model is not trivial. One way to do it is to run a simulation experiment for any possible value of the input variables. However, in most practice application, it is unfeasible because of the tremendous computational time it would take. It can be due to the large number of different values and combination to test, or to the complexity of running the model for non-optimal values (especially for extreme case values). Rather than trying all possible values, the idea is to use an optimization method to find optimal values for input variables. The problem of finding the set of parameters that induce the best model regarding the accuracy of several outputs is a multi-objective optimization problem. To solve that problem, new approaches were developed and gathered in the field of "simulation-based optimization" (Carson and Maria, 1997). Such approach can be used to simulate health-care systems (Lucidi et al., 2016). In (Lucidi et al., 2016), the author proposes a simulation-based optimization approach that makes use of a discrete-event simulation model to model hospital services. This model is then combined with a derivative-free multi-objective optimization method to solve a resource allocation problem.

Here, we developed a simulation model where entities are simulated independently. It allows us to use parallel computing and to use a limited amount of computing time (few seconds for a million of entities). Hence, in our case studies (Chapter 7), we were able to use a **brute-force checking of all the possible input values**. However, in the perspectives of integrating other aspects of the process such as resources and patient-to-patient interactions, parallel computing would not be possible anymore. Discrete-event simulation and multi-agent simulation would be good modeling alternatives. This is why we wanted to mention simulation-based optimization as a way to tackle the challenge of a good model calibration.

5.5.3 Model validation with historical data

The validation of our simulation model is done by comparing the output values of key performance indicators with those of the historical data. In our case, we have two separate data sets that can be used for validation. The first is the **original log** and the second is the **aligned log** created afterward. Indeed, in our methodology to build the simulation model, we converted the original event log into an ad-hoc event log that we called an aligned log (Section 4.4, Chapter 4). This choice was motivated by the fact that a model cannot represent every possible behavior seen in the original event log, it must generalize and be synthetic. So, we faced a matching bias between the original log and the model when studying the decision points and the statistical distributions. We proposed the solution of using a log with perfectly aligned traces to avoid this bias.

By definition, the aligned log is much closer to the model than the original log because each trace of the aligned log can be perfectly replayed in the model. It is not the case of the traces from the original log. So, the model validations with the two logs would provide different results in most cases. For instance, if we want to compare the performances of the model on KPI #4 defined previously (the average number of times that a state s_i was visited by simulated entities), we face several pitfalls. Again, some events of the original logs may not be represented in the model (by a care-state). Hence, there is no possible way to evaluate a KPI #4 for these events. Similarly, the transitions (as input or output) of these events, as observed in the original log, cannot be seen in the model. Furthermore, when counting the number of times that a given transition was observed in the original log versus the number of times the related wait-state was used in the simulation, the noise of the log will disturb the result. Indeed, the transition $A \rightarrow C$ may not appear often in the log because of a noisy event B that is not represented in the model ($A \rightarrow B \rightarrow C$ is seen in the log). The resulting value is biased. These issues do not occur when using the aligned log. In the following, we present a standard validation procedure using the aligned log. We discuss a way to perform the validation with the original log in the future works section.

Validation of a single KPI with the aligned log

The model validation with the aligned log is straightforward. Each output value of a KPI obtained from the model is compared to the same measure from the log. More formally, let *CPSC* be a clinical pathway state chart, let L_a be an aligned log and let $R = \{KPI_1, \ldots, KPI_n\}$ be the set of key performance indicators chosen to validate *CPSC* with L_a, with $n \geq 1$. Then, for each KPI we compute the absolute difference between the model value and the log value:

$$\delta_i = |KPI_i^{CPSC} - KPI_i^{L_a}|, \quad \forall i \in [\![1, n]\!] \tag{5.2}$$

where KPI_i^{CPSC} is the average value of the Monte-Carlo replications for the KPI #i, and $KPI_i^{L_a}$ is the value computed from the aligned log. The simulation also produces an error value ε_i which gives the

confidence interval $[KPI_i^{CPSC} - \varepsilon_i, KPI_i^{CPSC} + \varepsilon_i]$around the value of KPI_i^{CPSC} (See Section 5.4.2 for the error calculation). The confidence interval gives the range of values in which we are confident that the KPI value obtained with the simulation is.

A first approach for the general validation: binary measure

Based on the difference δ_i between the model and the log, the first way to assess the model validity is to use a binary validation process: if the KPI value obtained from the log belongs to the confidence interval of the simulation, we can conclude that the model is valid regarding this specific KPI. Formally, for each KPI we define a dedicated validation function v_i:

$$\begin{array}{rccl} v_i & : & \mathbb{R}^2 & \rightarrow & \{0,1\} \\ & & (\delta_i, \varepsilon_i) & \mapsto & \left\{ \begin{array}{ll} 1 & \text{if } \delta_i \leq \varepsilon_i \quad \forall i \in [\![1,n]\!] \\ 0 & \text{else} \end{array} \right. \end{array} \tag{5.3}$$

In other words, if $v_i(\delta_i, \varepsilon_i)$ is equal to 1, we can conclude that the model's behavior regarding the KPI #i is the same as the one observed in the log. The approach is repeated for each KPI of R. In most cases, and this is the main interest of a simulation model, there are several KPIs to evaluate (from few to several tens). Once we have computed the validation function of each KPI so that we know if the model is valid for each KPI independently, we must aggregate these results to determine if the model is valid globally. One way to do it is to choose a threshold on the minimum percentage of KPIs on which the model shall be valid. Moreover, all the KPIs are not equally important for the validation. We introduce a set of weighting factors $\beta_i \in [0,1]$ for that purpose. Let $T_{min} \in [0,1]$ be such a threshold, then a simulation model *CPSC* is valid if the inequality 5.4 is verified:

$$\sum_{i=1}^{n} \beta_i . v_i(\delta_i, \varepsilon_i) \geq T_{min} \quad \text{with} \quad \sum_{i=1}^{n} \beta_i = 1 \tag{5.4}$$

Remark 14. A model is declared valid under a given configuration (= a set of parameter values), not for any configuration.

Remark 15. If we impose that $T_{min} = 1$, the model is valid if and only if it is valid for every KPI.

The choice of using $T_{min} < 1$ is motivated by the important necessity of avoiding overfitting. Overfitting occurs when a model tends to copy to much the behavior of the data it was supposed to learn from, so that it cannot generalize the observed patterns. It is like memorizing each trace of log and then repeating those traces when simulating new patients. It does not distinguish between the noise and the frequent patterns. In our case, overfitting may occur because of the decision point classifiers or the distribution fits chosen to build our model (Chapter 5). If the classification models overfit the data locally in each care-state, the resulting *CPSC* will express the same behavior globally. It reinforces the need for a selective choice of the most appropriate machine learning algorithm for the local classification problem. Still, choosing the appropriate value for T is not trivial and may appear subjective. The choice depends on the size of the case study. For instance, for a model with only 5 different KPIs to evaluate, it appears reasonable to expect that the model is valid regarding all the 5 indicators ($T_{min} = 100\%$). On the other hand, for a model with hundreds of KPIs, more flexibility can be admitted ($T_{min} = 80\%, 90\%, 95\%$) because no model would possibly fit all KPIs. In any case, T_{min} can be seen as the minimum number (or percentage) of KPIs that a model must validate locally to be valid globally.

Remark 16. For any case study where a simulation model is created and must be validated, it is important that the value of T is determined beforehand of any run. The choice must not be dictated by the outcome results of the simulation to guarantee the objectivity of the results.

A second approach for the general validation: closeness measure

The second way to assess a model's validity is to use a validation function that computes the distance between the simulation result and the log. Formally, the set of v_i functions becomes:

$$\begin{array}{rccc} v_i & : & \mathbb{R}^2 & \rightarrow & \mathbb{R}^+ \\ & & (\delta_i, \varepsilon_i) & \mapsto & |\delta_i - \varepsilon_i| \end{array} \tag{5.5}$$

Similarly to the binary validation, the results of the v_i functions need to be aggregated to assess the global validity of the model. A model is valid if the distance to the log is small enough. Again, a threshold value T_{max} is used to objectively determine if the validity criterion is met. A simulation model *CPSC* is valid if the inequality 5.6 is verified:

$$\sum_{i=1}^{n} \beta_i . \frac{v_i(\delta_i, \varepsilon_i)}{\mathcal{N}_i} \leq T_{max} \quad \text{with} \quad \sum_{i=1}^{n} \beta_i = 1 \tag{5.6}$$

where $\mathcal{N}_i$ is a factor used to normalized v_i, so that KPIs measured on different scales are comparable and summable. The advantage of the closeness validation is to allow for a broader range of possible v_i values. When using the binary validation, there is no difference between the cases where the difference between the model and the log is very slightly over the ε value and the case where the difference is extremely higher than the ε value. In both cases, the model is not valid. With the closeness validation, we intend to avoid this abrupt cut-off effect. A model that is rather close to the log regarding each KPI, but not close enough to be lower than the ε value, is a model of interest. First, it may be the best model that we are able to find. Validation results obtained with the closeness validity allows for a comparison between several models. Second, it seems more promising to change a model that is almost valid than any non-valid model to finally get a valid model. The advantage of the binary validation is to be simpler to interpret. A model is considered as valid if it is locally valid for a sufficient number of KPIs. The value T_{min} directly expresses this "sufficient" threshold. Concerning the closeness validation, the choice of a value for T_{max} is more difficult to interpret and normalization factors need to be found.

An illustrated example of validation

Figure 5.6 shows an example of validation results for a small size model. The *CPSC* is made of 16 care-states and 35 wait-states. The validation is done by evaluating the value of the KPI #5 (the average number of different entities that visited a given state at least once). Here, only the results for the 16 care-states are shown, not those of the wait states. It produces 16 different KPIs. For each KPI, the value from the simulation (blue histogram), the value from the log (orange histogram) and the simulation error ε (red bars) are displayed. The original log included 3 450 patients and we simulated 10 000 new ones. The simulation KPI values were standardized for a population of 3 450 patients to be comparable with the original log. Graphically, we can see that the confidence interval of 12 KPIs does include the value of the original log (states 2, 0, 6, 8, 25, 10, 7, 57, 11, 13, 5, 14). Based on the binary validation approach described above, if we use uniform weighting factors of 1/16 for each KPI, the validation score of the model is $\frac{12}{16} = 75\%$. If the validation threshold lower than 75%, the model would be considered valid.

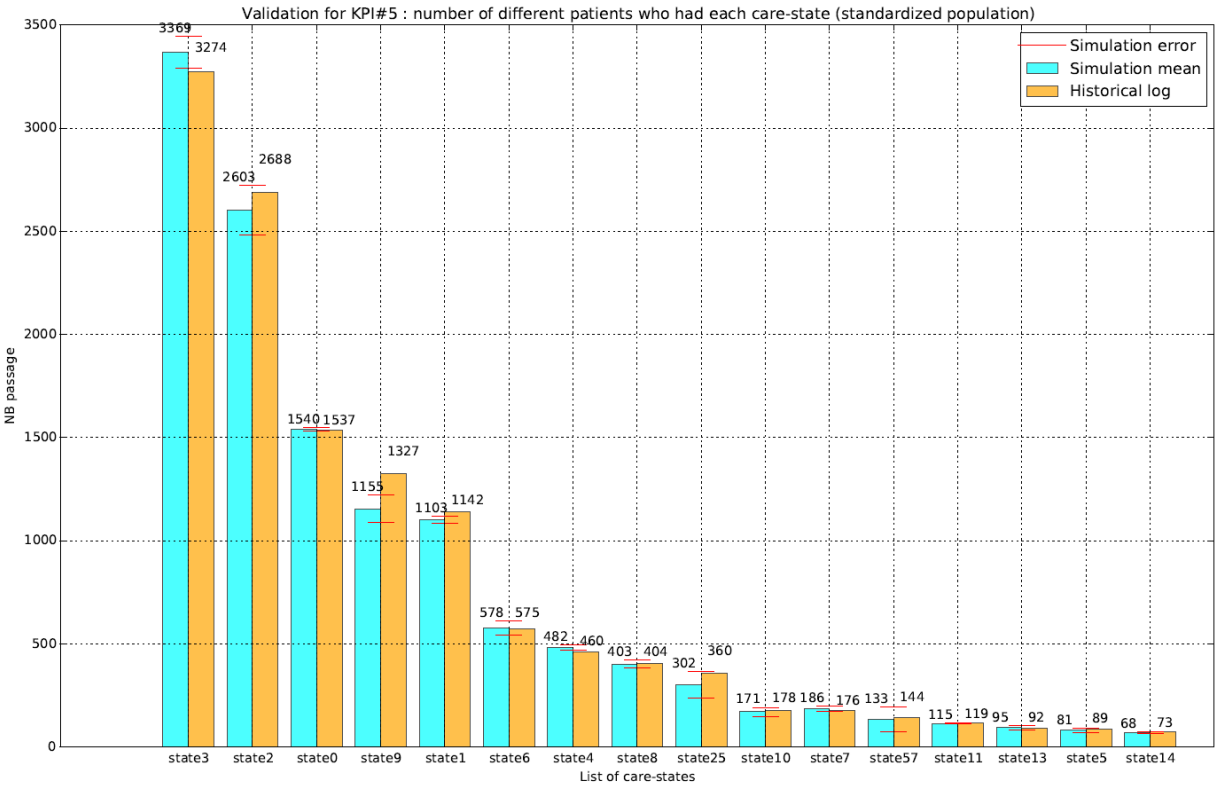


Figure 5.6: Illustration of validation results on 16 Key Performance Indicators

5.5.4 Summary of model validation

In the above section, we have seen the challenges related to a simulation model's validation, a broad range of possible validation techniques and the specific case of historical data validation. Historical data validation allows for an objective and quantified measurement of a model's validity. In the following, we consider that we have a simulation model that was calibrated and validated with such approaches. To sum up:

- The calibration is performed automatically by brute-force checking of all possible input values since simulation experiments are run in parallel.
- New validation approaches are proposed. A binary measure against the aligned log is proposed on predefined KPI's using a validation threshold proposed by the user (for example 85% is sufficient to validate the model presented in the case study in Chapter 6). A closeness measure is also proposed to avoid the cut-off effect of binary.

At this point, we obtain a validated simulation model and we can focus on its utilization.

5.6 Sensitivity analysis

Once a simulation model has been validated, we can start thinking about its utilization. The goal of creating a simulation model is to replicate the behavior of the actual system, so that we can then analyze how a change affects the output of the process. The relationship between the input changes and the outcomes is

Table 5.1: List of the variables independent of the case study

Modeling step	Variables
Process Mining related variables	Size of the discovered model
	Number of tabu search iterations
Variables from the simulation model's creation	The similarity matrix used for trace alignment
	Each parameter of the classification algorithm used for decision point analysis
	The maximal length of the perfect traces sequence
Variables from the simulation execution	Number of simulated entities
	Confidence level α used for the error measurement
	The validation threshold (T_{min} or T_{max})

not straightforward because it is not described by a function. Instead, we use the model to represent the entire care process with several intermediate random distributions and advanced classifiers.

A sensitivity analysis (SA) is the study of how the variations of input parameters impacts the model outputs. It is a technique used to determine how different values of an independent variable impact a particular dependent variable. SA can be used for different purposes. For instance, it can be used to calibrate a model. When the exact value of an input parameter in unknown, we can test a range of possible values to determine the most credible value based on the produced outcomes. SA is also a validation technique. A model is validated by ensuring that it produces the desired outcomes for several configurations (i.e. several sets of input values). SA can be used to ensure that a model is robust in case of random variations, i.e. that it does not produce huge output variations under very small changes in the input conditions. SA also serves as an optimization tool if it is used to determine the values that produce the maximal/minimal outcomes.

Here, we are interested in producing a sensitivity analysis of an already valid model in order to get new insights about the impact of various input parameters of the case study. In the continuity of our methodology for an automatic creation and validation of a simulation model, we propose an automatic generation of a SA. For that, we first need to select eligible input variables that may impact the model outcomes. For each selected variable, a range of possible values to test is then determined. Finally, the results of the SA are presented graphically (Tornado diagram or individual impact curve).

5.6.1 Automatic selection of variables to evaluate

We distinguish two groups of possible variables to test in the SA.

Variables that do not depend on the underlying data and case study. These variables were used all along the technical construction of the simulation model, from the very first step of process mining to the final validation. An extensive list of these variables is presented in Table 5.1. The analysis of case study-independent variables in a sensitivity analysis is a focus on the model's behavior. Indeed, it would help quantifying the changes in response to a different configuration. It could determine if the model is robust and if some variables have a higher impact on the model's behavior. However, this analysis is partly redundant with the calibration and the validation of the model. A search for the best values of simulation input parameters was performed to find a valid model. The search was only done for the simulation model

parameters. The impact of all the variables used prior to the simulation model creation was not assessed. We performed several local validations of the used parameters, but not globally on the final outcomes of the simulation. The advantage of proposing an automatic procedure to transform raw data into outcome results of simulation model is to allow for such a feedback loop. It was not designed in the current study but will be discussed as further extension. So, we do not include these variables in the SA.

Specific variables related to the case study. These variables are either **event attributes** or **trace attributes**. They were used for different purposes. We used trace attributes to learn decision point classifiers. We used event attributes to generate random distribution fitting. We combined event attributes and trace attributes to model the relationship between the two (e.g. a patient that has the hospital event "obstetric surgery" will not have the attribute "obesity" anymore). In health-care, examples of hospital event attributes are the length of stay, the medical diagnosis, the severity of the stay, the injected dose of drugs, the cost and the biology results. Examples of patient attributes are age, gender, size, weight, presence of comorbidities and medical history. For each trace attribute, we derived a statistical distribution from the original log. The distribution is either a random distribution for numeric variables (ex: age) or a discrete probabilistic distribution for categorical variables (ex: male/female dividing).

Whatever the group of a variable is (independent or not of the case study), it can be identified as one of the three following types: textual, categorical or numeric. They are presented in Table 5.2. Here, we do not consider textual variables. Once we have identified the variables that can be included in the sensitivity analysis, we need to determine their variation range.

Table 5.2: The 3 different types of variables

Type of Variable	**Textual**	**Categorical**	**Numerical**
Description	Made of words, cannot be easily interpreted (requires knowledge about the language semantic) and the studied domain	Values of a categorical variable cannot be compared and ordered	Values can be compared, ordered and whose differences are explainable
Possible values	Almost an infinite number of possible values, like languages	Limited number of possible values	Any value within a finite or infinite interval
Example	A doctor's consultation reporting	Gender takes the value male or female, the blood type of a person is A, B, AB or O	Costs, age, length of stay and body temperature
Formalism	*Text mining* is a branch of artificial intelligence that is dedicated to derive high-quality information from text	A categorical variable with K ($K > 1$) possible values is described with a categorical distribution: each value i has a probability p_i and $\sum_{i=1}^{K} p_i = 1$	Either discrete (probabilistic distributions, similar to categorical variables) or continuous (probability density function, e.g. normal distribution).

5.6.2 Variation range of the variables

The variation range of a variable depends on its type. The variation range of categorical variables and of discrete numeric variables can be determined automatically. These variables can be described by a probabilistic distribution where each probability belongs to the interval $[0-1]$. Let x be such a variable and $X = \{x_i\}$ be the set of $K(K > 1)$ possible values that x can take. Then, the probability distribution of x is:

$$\begin{aligned} &\text{Support: } x \in X \\ &\text{Parameters: } p_1, \ldots, p_K \\ &\text{where } \forall i \in [\![1,K]\!], p_i = p(x = x_i), 0 \leq p_i \leq 1 \text{ and } \sum_{i=1}^{K} p_i = 1 \end{aligned}$$

The range of possible values for each p_i is $[0,1]$. We propose the procedure described in Algorithm 4 for the sensibility analysis of a variable x.

Algorithm 4 Automatic variations of categorical or discrete variables for sensitivity analysis

Require: A simulation model *CPSC*, a categorical or discrete variable x described by $p_1, \ldots, p_K$
1: Choose an incremental step $\Delta \in]0,1]$
2: **for all** $i \in [\![1,K]\!]$ **do**
3: $p_i \leftarrow 1$
4: **for all** $j \in [\![1,K]\!] \backslash \{i\}$ **do**
5: $p_j \leftarrow 0$
6: **end for**
7: **while** $p_i > 0$ **do**
8: Run a simulation of *CPSC*
9: $p_i \leftarrow p_i - \Delta$
10: **for all** $j \in [\![1,K]\!] \backslash \{i\}$ **do**
11: $p_j \leftarrow p_j + \frac{\Delta}{K-1}$
12: **end for**
13: **end while**
14: **end for**

This procedure allows to test a large variety of configurations for variable x. We are well aware that it does not represent an exhaustive test of all the possible combinations of values for the p_i. Still, it tests high values of each p_i to see the impact of specific values of x. Regarding the choice of the incremental index Δ, it depends on the available computing power. The smaller the values of Δ are, the higher the number of runs to launch is. For a categorical variable with K possible values, the required number of simulation runs is given by Equation 5.7.

$$Number\ of\ runs = K \times \frac{1}{\Delta} \tag{5.7}$$

The variation range of a numeric continuous variable can be determined in two ways. It can be done by an expert that has the knowledge about the possible values that a variable may take. For instance, in the case of a patient's attributes such as the systolic blood pressure, physicians know that the range of possible values is $[70-190]$ mmHg. The necessity for experts' opinion is highly variable dependent, and so case study dependent. The expert's opinion approach is a "manual" way to determine the possible range of values. It must be done for each variable and it may require to cross several experts. The second

way to determine the variation range of a numeric variable is to use the most probable value found in the historical data and to choose surrounding values. This approach is based on the fact that we have numerous observations of values for each variable. Thanks to these observations, we were able to fit the data to find the closest random distribution. Now, we propose to shift the random distribution to get new random draws for the variable. The best fitting random distribution is not changed, we keep the same function (normal, log-normal, exponential, beta, Weibull, etc.) with the same parameters. We only add a translation factor T. That way, it respects the shape of the distribution that is characteristic of the initial data but we explore new possible values. The translation factor is chosen based on the historical data. For a given variable x, we compute the standard deviation σ_x of the observations from the data. Then, we propose that the range of possible values for the translation factor T is $[-\sigma, +\sigma]$. Figure 5.7 shows an example of a Weibull distribution shifting for two values of T (-2 and $+2$).

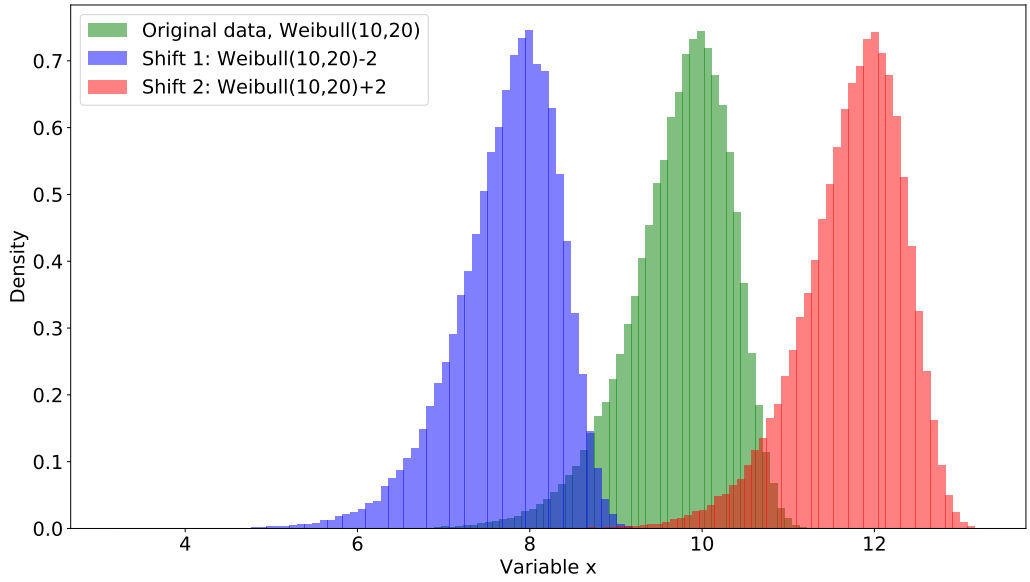


Figure 5.7: Example of shifting for a Weibull distribution

The variation of the translation factor T within the interval $[-\sigma, +\sigma]$ is determined with an incremental step Δ, similarly to categorical variables. Algorithm 5 presents the procedure to perform the sensitivity analysis of a continuous variable. It is done by running a simulation run for each possible value of T within its variation range. A specificity of continuous variables is that their domain of definition is an interval. In comparison, for categorical or discrete variables, the set of possible values is totally known beforehand and no impossible value can be drawn. In the case of continuous variables, the translation shift that we propose may result in the draw of values outside of the variable's domain of definition. For instance, the variable "age of patients" can never be negative, its domain of definition is $[0, +\infty[$. This constraint motivated the choice of using translations withing the range $[-\sigma, +\sigma]$, which provides, by definition, a good insight about the scattering of possible values. A translation that does not exceed standard deviation will likely generate random draws withing the definition domain. Still, we also use truncated random distributions for all variables with a bounded (or semi-bounded) domain of definition. A truncated function returns the value of the lower (upper) bound of the domain of definition if a draw is smaller (higher).

Algorithm 5 Automatic variations of continuous variables for sensitivity analysis

Let *CPSC* be a simulation model
Let x be a continuous variable described by a random distribution $\mathscr{D}_x$
Let σ be the standard deviation of x
Choose an incremental step $\Delta \in]0, 2\sigma]$
Set $T = -\sigma$ (translation factor)
while $T \in [-\sigma, +\sigma]$ **do**
    $\mathscr{D}_x = \mathscr{D}_x + T$
    Run a simulation of *CPSC*
    $T = T + \Delta$
end while

Finally, for any type of variables, the sensitivity analysis provides the set of outcomes from the simulation model that were obtained for a given set of input values for this variable. The interpretation of these outcomes are presented in the following.

5.6.3 Results of the sensitivity analysis

The goal of the sensitivity analysis is to determine the impact of the input variables on the model outcomes. For each input variable, all else being equal, we ran the simulation with several different values. The SA results can be seen in two ways. First, we can individually explore the absolute impact of each variable. Then, we can compare and sort the variables to know their relative impact.

Single input-output relationship

The individual impact of each variable is determined for one KPI at a time. Even if the model produces several KPI in output, we are not in a multi-objective approach here, we want to discover the relationship between one input variable and one output KPI. Our goal is two-fold: (1) we determine if the input-output relationship can be modeled by a regular function and (2) we determine the pairs of input values for which there is a significant difference in output. For each input value of a variable, the model provides an estimation of the studied KPI and an error interval.

The existence of an input-output relationship is tested by fitting the data with standard regression models (linear, polynomial, logistic, exponential). The significant thresholds are determined with the error interval. If the output values obtained for two different inputs are close and their error interval overlaps, then no significant difference is observed. The conclusion might be different depending on the chosen confidence level (See Section 5.4.2 for precision about confidence intervals). The higher the wanted confidence level is, the larger the error interval is and the more difficult it is to have significant differences. A solution to reduce the error is to increase the number of replications, at the cost of more computational time. If we find two outputs for which the estimations are included in intervals that are totally disjointed, then we can assess the presence of a causal relation between the input variable and the output KPI.

Relative comparison of the impact of several inputs on one output

We are also interested in knowing the variables that impact the most a given output. For that, we perform a comparative sensitivity analysis based on the individual results. Even if the input variables are of different natures (age, gender, injected dose, location, etc.) and so are their range intervals, there exist analytic techniques to compare them all. The most used technique is a graphical approach, the *tornado diagrams*.

A tornado diagram is basically a bar chart which plots a KPI's output values for 3 specific values of each input variable: the lower value, the baseline and the higher value. Then, the difference in output values between the lower and the higher values is computed. Finally, the input variables are sorted by decreasing order of this difference. In our case, the baseline value of each variable is the observed value from the historical data. The lower and upper values are those of the variation range defined previously. As a result, a tornado diagram is a tool that gives an idea of which factors are the most important for a specific measurement. A separate tornado diagram must be built for each KPI of the case study.

Example. Figure 5.8 presents an illustration of a tornado diagram for 8 input variables (e.g. proportion of diabetic patients, with hypertension, with kidney failure, with cancer, age distribution, gender distribution). In this example, the variable 8 is the most impacting on the output KPI (e.g. the death rate). The baseline is represented by a vertical line and the impact of lower (resp. higher) values are displayed by orange bars (resp. blue) on the left (resp. right) part of the baseline. We see that a decrease in variable 8 will more significantly decrease the KPI compared to its increase (the graph is not symmetrical).

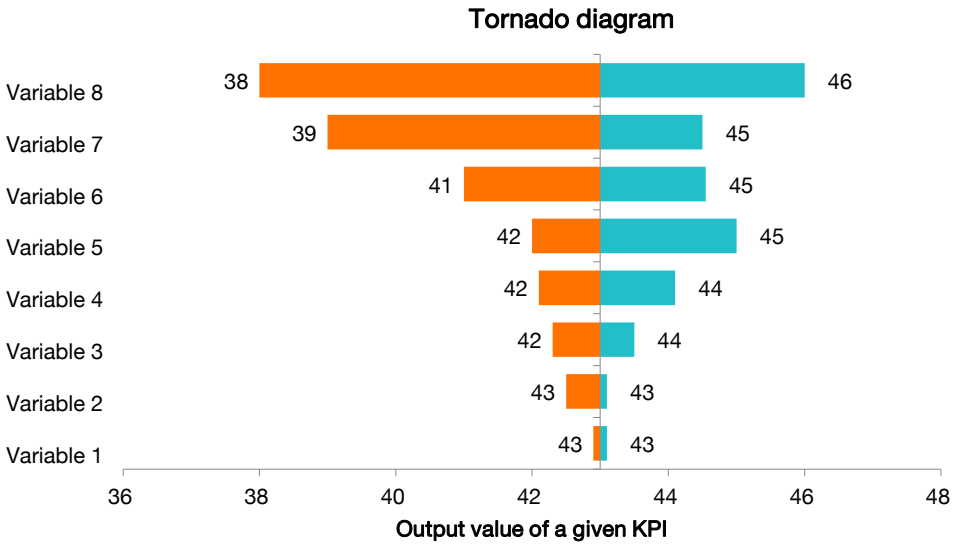


Figure 5.8: Example of a tornado diagram for 8 input variables

The quality of a tornado diagram heavily rests on the selection of a variation range for each input variable. A common mistake would be to vary each variable in the same proportion around the baseline (e.g. to test a $+/-10\%$ variation) without considering that variables are of different natures and that a 10% increase may have no sense for certain variables. For instance, a daily 10% increase in the systolic blood pressure is common and would not impact much on a patient's condition, whereas a 10% increase in a chemotherapy dose could be lethal. Hence, a scrupulous determination of each variable possible range of values is of major necessity. This reason adds to our choice of building our variation ranges based on the historical standard deviation. That way, the range becomes variable-dependent and will allow to test realistic values. In addition to the choice of the lower and upper values used for the tornado diagram, we can assess the risk that this value will be met in real life. This risk is often estimated by experts.

5.6.4 Summary

To sum up, sensitivity analysis helps us to **determine the most impacting variables on each output measure**. Our "automatic sensitivity analysis" package can be summarized in 4 items:

- **Automatic selection of variables to evaluate:** (i) modeling variables (size of the model, similarity matrix elements, confidence level, etc.), and (ii) case study variables, including events' attributes (length of stay, costs, etc.) and patients' attributes (age, gender, obesity, diabetes, medical history, etc.).
- **Automatic generation of a variation range** for these variables. Depending on their type, we developed a procedure to generate relevant intervals.
- **Computation of single input-output relationship** for one output KPI.
- **Computation of relative contributions of several inputs** over one output KPI.

SA gives the decision makers some insights into the uncertainties and their potential impact. It also potentially discovers hidden input-output relationships that were not straightforward to determine without a comprehensive model. Such information can be used to organize an action plan with the most relevant leverages regarding the target (e.g. fighting opportunistic infections in patients with hepatitis C would be much more cost reducing on the long term compared to reducing hypertension for the same patients).

5.7 What-if scenarios evaluation

The second part of the simulation model utilization is the evaluation of what-if scenarios. This part is the main motivation for the creation of a simulation model. A clinical pathway model integrates numerous variables to be representative of the actual system's behavior. These variables represent different aspects of the model (patients' features, decision rules at routing choices, delays, existence of a direct transition between two states, etc.). When simulating a new patient, they can take fixed values (e.g. date of the initial diagnosis) or evolve dynamically through the simulation (e.g. age of a patient). Some of them are more complex to apprehend (e.g. a decision tree modeling the routing choice after a given state). A what-if analysis is the definition of a new scenario that we want to evaluate thanks to the model. For that, we give input variables new values (or new rules for decision trees), thus influencing the model behavior. The measurement of KPIs enables to appreciate the output changes induced by the new input values. The creation of scenarios to test is totally case dependent. Typically, when modeling a hospital service operating, basic scenarios involve adding/removing resources and looking at the impact on the service rate or the waiting time. A scenario with variations of only one variable is a sensitivity analysis. For what-if analyses, we can define more advanced scenarios that incorporate several concomitant variable changes in the model. The scenario is created to represent possible situations that could be faced by the system.

Here, we modeled clinical pathways based on the historical follow-up of patients over several years. The idea is to create a single model for all the patients suffering from a given disease. The interesting scenarios to test depend on the choice of a disease and of its main components (treatment, natural evolution, death rate). Examples of scenarios that can be studied are: (1) to modify the population structure to test the impact of an aging population, (2) change medical guidelines to allow for more patients to undergo an innovative surgery, (3) evaluate the long term impacts (on readmission, costs, death) of a new drug of a new medical device on the market. In the next chapter, we go through an extensive case study related to cardiovascular diseases.

5.8 Summary and perspectives

In this chapter, we proposed a formal procedure for the automatic conversion of a process model, in the form of a causal net, into a simulation model. Our objective was to be able to generate new patients that are close enough to the historical data. We used the concept of state charts to integrate several perspectives of clinical pathways into a single simulation model. After the simulation model creation, we introduced several generic key performance indicators that can be used for model validation. We run the model to simulate the pathway of new patients so that we compare the output KPIs with the historical values from the event log. A validated model is finally used to perform sensitivity analysis and what-if scenarios evaluation. Sensitivity analysis provides insights about the determinant factors (input variables) that most impact the model's behavior (output measures). In the remaining, we highlight the contributions of this work and two possible extensions.

5.8.1 Contributions

The contributions of this chapter are methodological. The framework that we propose to automatically convert a process model into a simulation model is generic. So, it can be applied to any database and to any cohort of patients. This constitutes the main scientific contribution of this work. It includes the definition of a new type of state chart, dedicated to the representation of clinical pathways, of automatic conversion and validation procedures. This last aspect appears to be sometimes neglected or made on subjective opinions. We propose an objective validation of a simulation model thanks to the use of the historical event log. Finally, we present two ways to take benefit of the created model in the context of clinical pathways: a sensitivity analysis for the search of determinant factors and the evaluation of what-if scenarios to test more advanced changes of the current process. The originality of the approach also relies on its combination with the process mining approach. Combined together, the process discovery and the conversion into a simulation model create a full methodology to turn raw data into an actionable model.

The methodology takes into account the features of the health-care data used as input. It proposes a set of predefined tools for accurate validation and extensive sensitivity analysis based on 5 relevant KPI's. All this material constitutes a simulation toolbox that can be personalized by practitioners depending on the case study, as presented in the next chapter.

5.8.2 Future works: further validation and a model of hospital services

Validation with the original log

A limitation of our approach is our choice to validate a simulation model with the aligned log, instead of the original log. This choice was motivated by the necessity to compute key performance indicators from the model that can be systematically derived from the log (e.g. the average number of patients that had a stay Y after a stay X). This was only possible with a log in which traces are perfectly re-playable, such as in the aligned log. In the case of the original log, the transition from X to Y might not even be observed if there are other types of noisy events in-between. In the previous chapter, we demonstrate the relevance of using a rigorous methodology to align the traces from the original log with perfect traces. Then, we could use the closest perfect trace of each original trace to take its place for the decision point analysis, and eventually here for validation. Still, even if the aligned log is a good replacement for the original log, one may want to actually compare the model with the actual data. A model's behavior is expected to be closer

to the aligned log than to the original log. This is due to the definition of the perfect traces (generated traces from the process model). The two types of logs have the same difference to the model in the best case only, when each original trace is exactly identical to its closest perfect trace (i.e. the model represents every possible sequence seen in the data).

The validation with the real log requires the definition of a new metric because we are comparing two objects of slightly different natures (different event classes and transitions). A way to do it could be to take benefit of the alignment procedure that we introduced in the previous chapter. The alignment of two distinct sequences with the Needleman-Wunsh algorithm is a quantification of their difference. The simulation model can be used to generate new patients and their sequence of hospital event. A possible validation metric of the simulation model could be the average value, for a large number of simulated traces, of their best alignment with a trace from the log. In other words, if each trace generated by the model is relatively close to at least one trace from the log, then the model is valid. A threshold value shall be defined ahead for the minimum level of the average closeness value that is expected to conclude positively. Other validation metrics could also focus on other perspectives than the sequencing of events. For instance, the time perspective could be investigated. Again, several convolutions are required to compare the model and the log on equal footing.

Modeling hospital resources and patients' interactions

Two levels of detail for clinical pathways

A second perspective to our approach is the individual modeling of each hospital stay as a small process itself. In our current model, care-states change patients' features in a deterministic way. An exception is made for the time spent in a state by a patient, a random value is drawn for a theoretical distribution. Still, the changes of being in that specific state are the same for all the patients. An interesting extension would be to add a resource perspective and to model medical decisions into the clinical pathway state chart. In the current approach, resources were not taken into account as our main objective was to model clinical pathways at a macroscopic level. The time scale of resources consumption appeared negligible (few hours) compared to the follow-up duration (several years). However, as a second step of the modeling process, we could enrich the model with finer-grained details. For each care-state, patients require hospital resources to be managed. Resources at stake are various, from human resources (nurse, doctor, stretcher bearer) to material resources (bed, drugs, dressing, operating room). Our definition of a clinical pathway state chart was though to be able to integrate resources without having to redefine every concept. Indeed, a care-state is defined as $c_s = (l, B)$ where l is a label and B a set of patient attributes to be updated in this state with new values. In the current version of our *CPSC*, the B set was used to update the features of any entity that goes through this state. B can be extended to other attributes that would be related to the resources, such as the number of available nurses to take care of a patient in state c_s. Wait states remain the same because they do not imply resource consumption as it represents the moment between two hospital events. The patient is out of the care system.

Discrete event simulation

This new integration of resources implies a dynamic management of their seizing and release by the patients. It means that patients cannot be considered independently, they interact with each other through resources. The Monte-Carlo simulation would not apply anymore. Thus, we could extend the modeling of care-states to the next level by proposing that each care-state is a dedicated discrete-event simulation

(DES) model. Such DES models would allow the integration of the resources, but also of patient-dependent decisions. These decisions can be medical (e.g. the choice of operating a patient) or come from the patient (e.g. the choice of being operated in a given hospital). The use of DES to model health care systems, and more specifically hospital services, was extensively done in the literature (Augusto and Xie, 2014). Here, we think that the originality of future works would be in the combination of two models (a state chart and a DES model) to describe clinical pathways at two levels of detail. The resulting model could be provided in the form of a general causal net representing the clinical pathway, as described here, where each node triggers the execution of a sub-model. The patients play the role of "agents" who advance through the clinical pathway and whose health condition evolves. Each variation of their condition is associated with a hospitalization (hospital event) modeled by a discrete-event-simulation model. The medical decisions and the length of the stay impact the condition of patients. Such an approach capable of modeling both the daily care process within a hospital stay and the long term evolution of patients' condition was not found in the literature, to the best of our knowledge. The resulting model would provide more precise results for decision aid in a health-care context.

Chapter 6

Case study: discovery and simulation of clinical pathways using the French hospital database

Contents

Abstract

This chapter presents a comprehensive case study to illustrate the practical use of the approaches introduced in the previous chapters. The French national database of the hospital claims from 2006 to 2015 is used as an event log. It contains the hospital records of several million patients. The case study focuses on the clinical pathway of patients with cardiovascular diseases. In particular, the case of patients suffering from cardiac arrhythmia and who need the implantation of cardioverter defibrillators is addressed. We show how we can discover a model of the clinical pathway and how it brings new information to practitioners. Then, we convert this descriptive model into a simulation model to test new scenarios. In the end, we are able to quantify the impact on the care process of an aging population and of a new implantation strategies. This illustrates the benefit of the method for medical decision aid.

6.1 Introduction to the French database of hospital claims

This section is dedicated to the presentation of the raw material used for real-life applications of process discovery and simulation of clinical pathways: the French database of all the hospital events between 2006 and 2015. We discuss a brief history of that database, its content and examples of existing studies.

6.1.1 Context of health data in France

The French health-care system, as most countries' health-care system, is divided in two parts: the hospital setting and the community medicine. Hospitals play a central role in the overall organization. They provide a broad variety of medical services, from consultations of medical specialists to the most complex surgeries. In opposition, community medicine is dedicated to primary cares and the follow-up of chronic diseases. The main actors of community medicine are general practitioners, specialists, pharmacists and home-health nurse. The overall health-care system is managed by a governmental institution, the Health Ministry. The *Health Insurance* (*Assurance maladie*) is responsible for the reimbursement part. It covers all the costs related to any type of care (general medicine, expensive drugs, laboratory tests, hospitalizations, rehabilitation, vaccinations, maternity, disability, etc.). The health insurance operates on the basis of tariffs that are set by conventions or by the health ministry.

In brief, the health insurance is in charge of managing all the care expenses of the country. Nowadays, care payments and reimbursement processes are fully computerized. Every time that a person visits his/her doctor or a pharmacy drug is reimbursed, a new entry is added to a dedicated and massive information system: the National Health Insurance Information System (Système national d'information inter-régimes de l'Assurance maladie [SNIIRAM]). The **SNIIRAM** is fed by the information generated by the takeover of all health-care consumption and hospitalizations in France. The main mass is provided by 1.2 billion care claims each year, the current volume being 450 terabytes. It is arguably the largest health-care database in the world. It contains the health data related to both hospital activities and community cares. Each person is given an anonymous ID when he/she needs care for the very first time. Then, the same ID is used for his/her entire life. It allows for the follow-up of the cares received by each patient. Although it appears very appealing to use such an exhaustive database for epidemiological studies or for the evaluation of the burden of diseases, its access is extremely restricted, even for research purposes. The main reasons for such restrictions are the risks of misuse of such sensitive data. However, there exist agreements that can be granted to access the hospital part of the SNIIRAM database.

In the context of this thesis, we could only use the hospital part of the SNIIRAM database, which is referred by the acronym "**PMSI**". A special agreement can be obtained through the National Commission for Information Technology and Civil Liberties (CNIL) to access the PMSI. The data are then provided by the Technical Agency for Hospital Information (ATIH). Here, we used the hospital database provided to the company HEVA under the accreditation number 2015-111111-56-18, databases number M14N056 and M14L056. In the following, we present in detail the PMSI database.

6.1.2 A national and medical information system database: the PMSI

A brief history of the database

The Program for the Medicalization of Information Systems (Programme de Médicalisation des Systèmes d'Information [PMSI]) is an integral part of the reform of the French health system to reduce the inequality of resources between health-care establishments. In order to measure the activity and resources of the institutions, it is necessary to have quantified and standardized information, the **PMSI**. The French PMSI took its inspiration from the American model developed by Professor Robert Fetter (Yale University) in which he proposed to create **Diagnosis Related Groups** (DRG). These groups, thanks to both medical and financial homogeneity, allowed for an empirical construction of the costs of hospitalization from standardized information collected on several million hospital stays. The data collected were classified in a deliberately limited number of groups of stays with medical similarity and a similar cost. Nearly 600 DRG groups were established. The PMSI project arrived in France in 1982 with the aim of defining the activity of health facilities and calculating the resulting budget allocation. However, it initially had more of a public health and epidemiological perspective, rather than a financial objective, which differentiated it from the original American model. Still, it was rapidly turned into a budget allocation tool. The French Ministry of Social Affairs adapted the DRG classification to the French system: it bears the name of Homogeneous Groups of Patients (Groupes homogènes de malades [GHM]), a nomenclature updated every year. The collection of data through the PMSI progressively became mandatory for both private and public facilities.

Since 2005, the PMSI has been used for the implementation of activity-based pricing, a new payment system for hospitals, based on their activity. The valuation of this activity within the framework of the PMSI makes it possible to remunerate this activity accordingly. The activity-based pricing mechanically induced a drastic increase in the quality of the information collected. Indeed, missing values or errors led to the absence of a payment for the hospital. Devoted departments and human resources are dedicated to the proper collection, encoding and validation of the data. By extension, "PMSI" is referred as the name of the resulting database, in addition to the legal and technical frameworks necessary to its creation. The PMSI is now used in several medical sectors, with different modes of collection. For MCO (Medicine, Surgery [Chirurgie], Obstetrics) hospital stays, it is based on the systematic collection and the automated treatment of medico-administrative information. Regarding the post-acute care and rehabilitation sector, the collection focuses more on the type of patient care and its degree of dependence. The PMSI is also applied for hospitalization at home, as well as for psychiatry. In the end, the biggest asset of the MCO-PMSI database is its exhaustiveness in France.

The content of the PMSI database

The volumes

The PMSI is made of several databases which gather all information related to hospital claims. These databases contain the medical and administrative information registered in the patient medical records for all the hospitalizations in France. In the context of the present thesis, we only consider the last 10 years of collected data (from 2006 to 2015 included) and the principal acute care database (Medicine, Surgery, Obstetrics). During that decade, **12.5 million distinct patients** were hospitalized each year on average. These patients induced an average of **25.4 million hospital stays** per year. Table 6.1 shows the evolution in the volume of data collected in the PMSI over that period. A regular increase of about 3% is observed each year, from 21.6 million hospital stays in 2006 to 28.9 in 2015. The large increase of 12% more stays in 2008 compared to 2007 is explained by a change in the coding practices. The new financing regulation created a strong incentive for an exhaustive reporting of hospital activities.

Table 6.1: Number of distinct patients and hospital stays in the PMSI from 2006 to 2015

	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015
Patients (million)	10.7	11.0	11.1	12.7	12.8	13.1	13.2	13.3	13.5	13.7
Stays (million)	21.6	21.2	23.8	24.6	25.3	26.1	26.7	27.4	28.2	28.9
1-year rise (stay)	-	-1.7%	12.4%	3.1%	2.8%	3.2%	2.3%	2.8%	2.8%	2.4%

Collected information

A hospital stay is defined as the set of actions and events that occur between the entry and the exit of a patient in a care facility. During a single hospital stay, a patient may visit one or several medical departments, may be treated for one or several diseases and may be seen by one or several physicians. Within a hospital stay, each time that a patient goes into a distinct medical department, a *medical unit summary* (Résumé d'Unité Médicale [RUM]) is produced and recorded in the information systems. Most of the time, patients only visit one department. The RUM contains a limited number of administrative and medical information which must be systematically documented and coded according to standardized nomenclature and classifications in order to benefit from automated processing. At the end of a hospital stay, regardless of the number of RUM generated, only one final summary of what happened is produced and stored in the information system. In the case of several RUM, data are aggregated to fit the final summary's format (e.g. lengths of stay are summed, only the most important diagnosis is kept and all the medical procedures are recorded). In the end, for each hospital stay, the data elements recorded are presented as one line (= one observation) of a structured database. The number of different items recorded is variable but not less than 70 fixed fields. A majority of items are internal codes that are used for the right trans-coding of what happened into the appropriate financing codes. They do not carry information about the clinical pathway. In Appendix D, we present the 28 most useful fields recorded in the PMSI database for the study of clinical pathways. These fields are clustered in 3 groups: the administrative fields (8 items), the patient fields (7 items) and the medical fields (13 items).

Administrative fields provide information about the care facility in which the stay happened, its duration, its cost and the ongoing version of the DRG classification. Most administrative fields contain scattered information about the calculation of the cost.

Patient fields are dedicated to pieces of information that specifically refer to each patient. It includes a unique patient ID, age, gender, home location (ZIP code) and the elapse time since the last hospitalization.

The latter is provided as a replacement of the actual date of the hospital stay. The reason is to reinforce the anonymity of a patient. Thus, we only know the discharge year and month, but not the exact day, and the elapse time (in days) since the last record for that patient (in the same of a different facility). The elapse time is useful for the medical follow-of patients and for the study of clinical pathways. The home location can be used to determine the attractiveness of a hospital in its region, and the required travel time for the patient to reach a care facility. In the PMSI, patients are identified with a unique and anonymous number (ID) that allows for the linking of records relating to any of their hospital stays. This personal ID is of major importance to enable the follow-up of patients across time. The main strength of the PMSI database is to be exhaustive. It contains the records of all the hospital events in France, public and private sectors included. A patient can be followed through his entire hospital history.

Medical fields provide information about the reason why the patient was hospitalized in the first place (*main and secondary diagnoses*) and about the cares which were provided during the stay (*medical procedures*, *expensive drugs*, *admission in intensive care*, etc.). The medical diagnoses are chosen according to the International Classification of Disease, 10^{th} revision (ICD-10^{th})[1]. ICD-10^{th} is the international standard diagnostic tool for epidemiology, health management and clinical purposes, and is updated by the World Health Organization. Each disease is referred by a code (ex: C34 stands for "lung cancer"). Regarding medical procedures, they are coded according to the common classification of medical procedures (classification commune des actes médicaux[CCAM])[2]. The CCAM is a French nomenclature which intends to encode the procedures practiced by physicians. This medical information is then combined with financial elements of the stay to find the Diagnose Related Group (ex: 04M091 stands for "Tumors of the respiratory system, severity 1" and the standard tariff for the care is 2,355€ in the public sector).

Limitations of the PMSI

Despite the incontestable interest of the PMSI database, one must be aware of its limitations. The first limitation is its scope. It only includes data related to hospital events, but not the cares provided outside of hospitals. This limitation is critical for the study of diseases mostly managed outside hospital (e.g. chronic diseases such as diabetes). A patient may not be seen at hospital for 6 months, it does not mean that he did not receive any care. Our objective here is to study clinical pathways, that-is-to-say the sequencing of medical cares received by a patient. Hence, our model would not reflect the reality if it lacks most of the key events that happened outside a hospital. This is why the choice of a case (= a disease) for the study of clinical pathways with the PMSI database must be thoroughly made. The care management of chosen disease must be hospital-centered.

The second limitation of the PMSI is the quality of the data collected. The collection is made by the medical staff with coding knowledge with only one purpose: financing the activity. It induces the following effect: only the elements of a stay that impact the tariff are well filled. The main and secondary diagnoses are essential to determine the diagnosis related group of the stay (and thus the cost), and so are the length of stay, the medical procedures carried out and the expensive drug administrations. However, fields which do not impact the calculation of the tariff are only optional. It was observed that additional diagnoses (in addition to the main and the secondary diagnoses) are imperfectly filled. For instance, if a patient is hospitalized for a stroke, his diabetes may not be mentioned in the report. It is a secondary aspect of the patient's condition that does not directly impact the medical and the economic aspects of the stay. No further resources were deployed to take care of the patient after the stroke episode because of the

[1] International Classification of Diseases, 10^{th} revision: http://www.who.int/classifications/icd/

[2] Common classification of medical procedures: http://www.ameli.fr/accueil-de-la-ccam/index.php

diabetes. Finally, regarding the data quality issue, we provide 2 indicators (average values between 2009 and 2015): the ratio of stays with a missing patient ID was 1.8% and there were 2.5% of stays with no DRG assigned. The third limitation of PMSI lies in the absence of data related to medical examination. The PMSI database is distinct from hospital information systems. The PMSI is the common framework used by every care facility in the country to gather financing information, but it does not replace existing electronic patient files used inside each facility. Among the medical data that would enrich the PMSI, we can list the results of biology tests, of imaging (MRI, PET-scan, X-ray) and of biopsy.

Exploitation of the PMSI database

The PMSI has generated the first permanent and medicalized database at the national level. It is now useful for several purposes.

1. A budgetary allocation tool.
2. A support for setting up strategic dashboards (= tool for organizational management)
3. An access to data on the types and the volume of diseases managed at hospital.
4. An assess to the real cost of hospitalizations in France (essential to economic studies).

Budgetary use of the PMSI

The first goal of the PMSI is to be a budget allocation tool. For this purpose, the activity of care of the patients of all the health establishments is declared. A national tariff, validated by the Health Ministry, is applied to each of the declared benefits. Then, Regional Health Agencies (Agence Régionale de la Santé [ARS])]) locally ensure the good management of hospital and medical expenses. It is a principle of adjustment of the financing to the activity actually observed. The PMSI thus constitutes a tool for reducing resource inequalities among health institutions, among departments and among regions.

Cost of hospitalized diseases (burden of disease)

With the PMSI, the hospital activity can be precisely measured using the definition of GHM (homogeneous groups of patients). Currently, 25 major categories of diagnosis (catégorie majeure de diagnostic) and 3 other major categories (catégorie majeure) have been defined. These categories are the first level of classification of an hospital stay and correspond most often to a functional system (affections of the nervous system, of the eye, of the respiratory system, etc.).

Strategic use of the PMSI

The PMSI can serve as a support for strategic dashboards, which are tools to help strategic decisions of the management of hospital. The PMSI database can be supplemented by the addition of useful information to practitioners and managers. For instance, the knowledge of the volumes of patients hospitalized in each department enables to choose the adequate resources (human or material) and to plan future investments and developments. The use of cartography can also help to identify the attractiveness of a hospital in its region.

Use in pharmaco-economic studies

In addition to the GHM (group of homogeneous patients) and thus of the full cost, another data source can be used to better estimate the breakdown in sub-components of the cost (medical staff, infirmary, consumables, medical procedures, catering, laundry). It is a sample of stays for which the detail of the

costs were specifically measured. This level of detail is the most interesting feature in the field of medico-economic studies. It makes possible to distinguish fixed items from variable items. When one considers the effects of a new care protocol or a new therapeutic approach, it is possible to consider, for a given hospital, that in the short term only variable items will have variations in expenditure. The impact on fixed items will be much more difficult to identify. For instance, reducing the average length of stay for a GHM is of interest to the hospital only if the time saved can be devoted to another activity.

In the present chapter, our objective is to show how our generic methodology can be applied on the PMSI database to extract knowledge about clinical pathways. It works for the study of any disease. In our model, we integrate costs, patient-related data, medical diagnoses and the effect of drugs (or medical devices). In this regard, our approach is at the same time an epidemiology study, a burden of disease and a pharmaco-economic study. We do not currently consider organizational aspects.

6.1.3 Example of studies using the PMSI data

Existing literature

The PMSI database has already been used as a source for numerous descriptive analyses. We gathered existing works into 3 categories based on their general purpose:

- **Epidemiological studies**: the PMSI is used to determine the number of patients affected by a disease and their characteristics (age, gender, location, etc.). The PMSI is particularly reliable for diseases that are managed at hospital (pulmonary tuberculosis (Girard et al., 2014), uterine fibroids (Fernandez et al., 2014), venous thromboembolic disease (Allaert et al., 2016), risk factors for osteoarticular infection (Petit et al., 2016), etc.)
- **Cost-oriented studies/ Burden of disease**: the PMSI is used to determine the cost of a disease from a health insurance perspective. The cost is usually evaluated over a 1-year period and for any care directly related to a given disease (chronic hepatitis C (Abergel et al., 2016), breast cancer (Benjamin et al., 2012), febrile neutropenia (Freyer et al., 2016), spinal tumors (de Léotoing et al., 2015), dengue (Uhart et al., 2016), home bortezomib injection (Touati et al., 2016), thromboembolic events in breast or prostate cancers (Scotte et al., 2015), etc.).
- **Treatment efficiency / adverse events**: the PMSI can also be used to analyze certain adverse events, re-hospitalization risks and hospital death rate ((Chaignot et al., 2015), drug adverse events (Osmont et al., 2013), death cause in atrial fibrillation (Fauchier et al., 2015), predictive scores (Fauchier et al., 2016), lung cancer management (Pages et al., 2016), etc.).

A preliminary approach for clinical pathway modeling: lung cancer

In the context of this thesis and of clinical pathways modeling, we studied the treatment received by patients suffering from lung cancer. The objective was to **analyze the sequences of treatments** received by patients hospitalized for lung cancer. We distinguished two groups of patients, metastatic and non-metastatic patients. From the PMSI database, we extracted all the patients with a lung cancer diagnosis (code C34*) in 2011, and we checked the absence of any prior hospital stay for the same reason between 2009 and 2010. Then, patients were followed for 2 years, until 2013 (each patient is exactly followed during 730 days after his diagnosis). The study was carried out for 5 of the 27 French regions, including Paris region. Here, we only present the results of the *Rhône-Alpes* region.

In Rhône-Alpes, **3,696 patients** were included and generated **65,417 hospital stays**. We considered 14 types of hospitalization (= 14 treatments) that are directly related to the care management of lung cancer. They are presented in Figure 6.1. We gathered any other type of hospital stay in the category "other". We assumed that several hospital stays of the same type are in fact one treatment (e.g. a weekly chemotherapy session is repeated over several months, which generates tens of stays). In order to detect similarities in these patients' sequences of treatment, we used a simple heuristic: (i) patients who received the same treatment during their first stay are gathered together, (ii) in each resulting subgroup, patients are divided again based on their second treatment, (iii) the same operation is repeated until the last stay. The strength of our approach was to provide an innovation data visualization to see all the possible sequences in one graph.

An example of our new graph of sequences is presented in Figure 6.1. The sequence of treatments starts from the inner circle of this *sunburst graph*. For instance, in Rhône-Alpes, 34% of patients started with a checkup, 15% with a curative surgery and 13% with chemotherapy. Then, each additional layer shows the following possible sequences. For instance, after the initial checkup, surgery 4% of patients (of the 3,696) initiated chemotherapy and 5% went in palliative care.

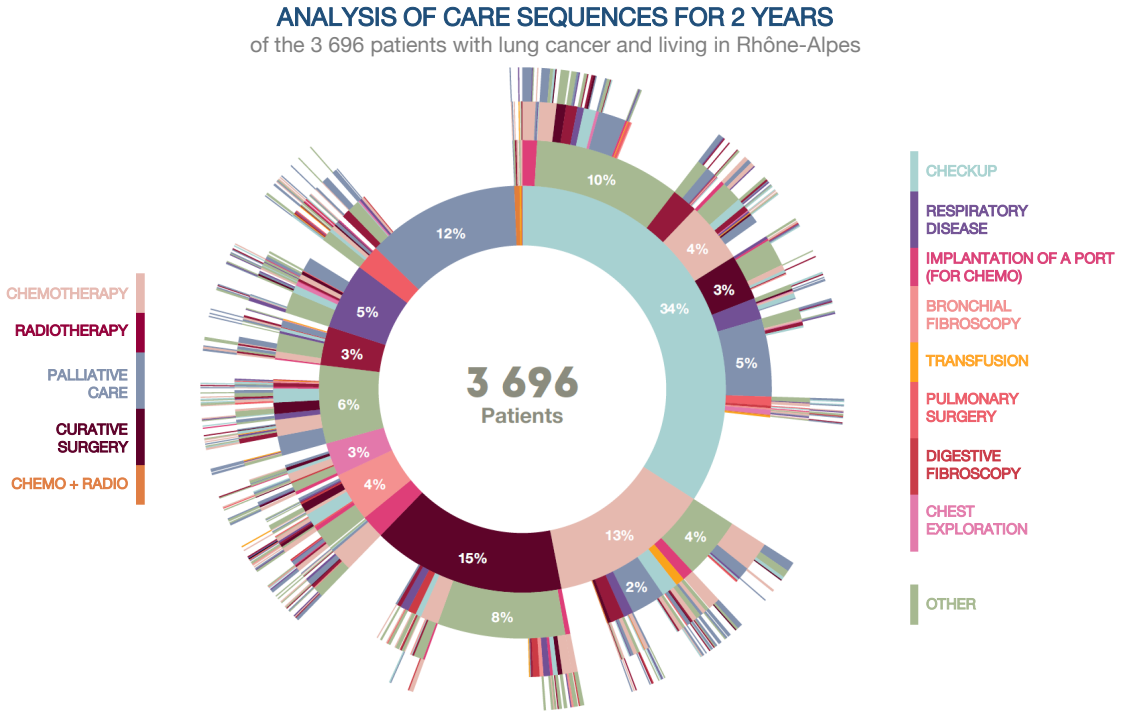


Figure 6.1: Analysis results of care sequences in lung cancer

An advantage of the approach is to be simple to implement (the heuristic) and graphically powerful. It provides new insights about the care management of patients during a two-year follow-up. In addition, it allows for comparison between two regions or two groups of patients. For instance, Figure 6.2 shows a comparison between non-metastatic patients and metastatic patients in Rhône-Alpes. These two sunburst graphs can be seen as a split of the Figure 6.1 graph in two graphs. The comparison shows that a large proportion of metastatic patients immediately goes in palliative care (23%) because no treatment could work at that stage, whereas non-metastatic patients can have curative surgery (29%) as a first treatment. We see that the therapeutic strategies are extremely different.

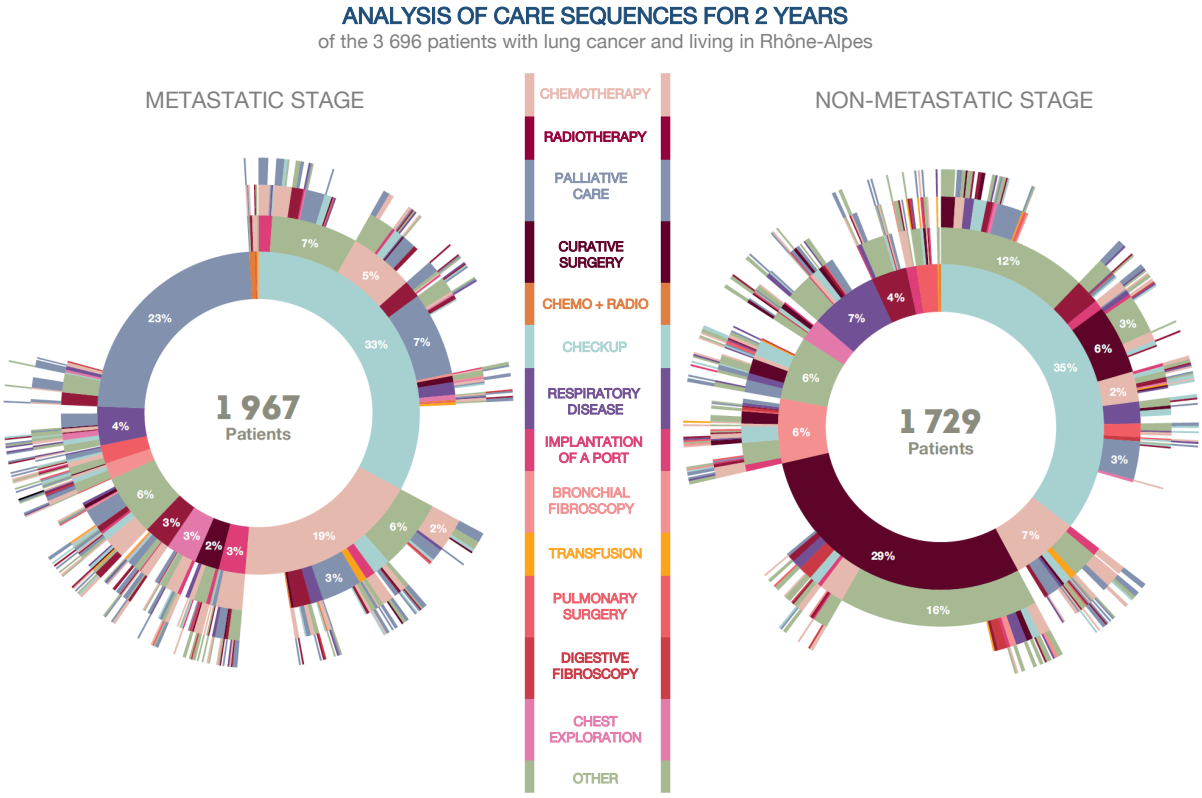


Figure 6.2: Comparison of metastatic status care sequences in lung cancer

The lung cancer study is a preliminary work on clinical pathways. It illustrates the capacity of the PMSI to provide strong medical insights about disease management. A basic heuristic, combined with an innovative data visualization, can help decision makers in their understanding of the care management of lung cancer in France.

A machine learning project: classification of patient profiles in HIV

In Chapter 4, we presented a **health-care analytics toolbox** with 3 components (*comparison of two sequences*, *predictive models* and an *automated analysis process*). Here, we present the results of using the approach described in the predictive models. The focus is not on the prediction of the next step of a clinical pathway, but on the classification of patients into cost profiles. The therapeutic area is the **Human Immunodeficiency Virus** infection, which causes the Acquired ImmunoDeficiency Syndrome. The entire study is presented in Appendix E.

6.2 Cardiovascular diseases, arrhythmia and implantable cardioverter defibrillators

This section presents the medical case that we study. We first present the context of cardiovascular diseases, then we explain how we can identify these diseases in the PMSI database and we finally explain our objectives.

6.2.1 General context

Cardiovascular diseases

Cardiovascular diseases are one of the major health problems today. It was ranked the first leading cause of death in the world in 2012 by the WHO, accounting for almost 17 million deaths in the world in 2005 (Mehra, 2007). It represents 30% of all annual deaths. More specifically, cardiac arrhythmia is a group of conditions in which the heartbeat is abnormal. A cardiac arrhythmia occurs when the heart beats irregularly or if it beats fewer than 60 pulses (too slow) or more than 100 pulses per minute (too fast), without being justified (e.g. physical effort). There are several types of cardiac arrhythmia which can be grouped in 4 main types: extra-beats, supra-ventricular tachycardia, ventricular arrhythmia, and brady-arrhythmia. Here, we are only interested in **ventricular arrhythmia** because they are the most important cause of sudden cardiac arrest (80% of the cases). Sudden cardiac arrests (SCA) represent half of the deaths due to cardiovascular diseases, thus accounting for 15% of global deaths every year (Sasson et al., 2010). Sudden cardiac arrests affect about 40,000 people per year in France (0.7 per million population) and 300,300 in the USA (1.0 per million population). Defibrillation by a defibrillator (implantable or external) is the only treatment to restore normal heart rhythm in case of ventricular arrhythmia (ventricular fibrillation and ventricular tachycardia).

Implantable Cardioverter Defibrillators

An Implantable Cardioverter Defibrillator (ICD) is a small device which is placed in the chest. Physicians implant the device to treat irregular heartbeats such as arrhythmia. An ICD uses electrical pulses or shocks to help control life-threatening arrhythmia, especially those that can cause SCA. SCA is a condition in which the heart suddenly stops beating. If the heart stops beating, blood stops flowing to the brain and other vital organs. SCA usually causes death if not treated within minutes. This is why ICDs are indicated in prevention of an episode of cardiac arrest due to a ventricular tachycardia. It is also recommended after the occurrence of an episode, but the survival rate is extremely low (about 3-8% for cardiac arrests outside of a hospital (Bougouin et al., 2014; Rea et al., 2004)). A defibrillator is similar to a pacemaker but it does not perform the same tasks. Bradycardia, an excessive slowing of the heart, is treated by a pacemaker which restarts cardiac activity by producing an electrical impulse. Tachycardia is treated by defibrillators. The latter will make it possible to re-synchronize a ventricle which begins to get carried away and which contracts in an anarchic manner. In practice, most defibrillators can also perform a pacemaker activity.

Defibrillators are battery-powered device placed under the skin. They are made of two parts: a generator and electrode wires (or leads). Thin wires connect the ICD generator to the heart. The electric pulses are delivered at the extremity of leads that passed through a vein to the right chamber. Figure 6.3 shows the location of an implanted defibrillator. The generator is usually nearby the heart and under the clavicle[3]. Because the electrical conduction system of the heart can present disorders at several levels, several types of ICDs have been developed in order to correct the various disorders. There are 3 types of ICDs, depending on the number of leads.

1. **Single chamber ICD** have a single lead in the heart. It is located in the right ventricle.
2. **Dual chamber ICD** have two leads, one in the right atrium and one in the right ventricle.
3. **Triple chamber ICD**, or bi-ventricular ICDs, have three leads. Locations are the right atrium, the right ventricle and the outer wall of the left ventricle.

[3]Blausen.com staff. "Blausen gallery 2014". Wikiversity Journal of Medicine. DOI:10.15347/wjm/2014.010. ISSN 20018762.

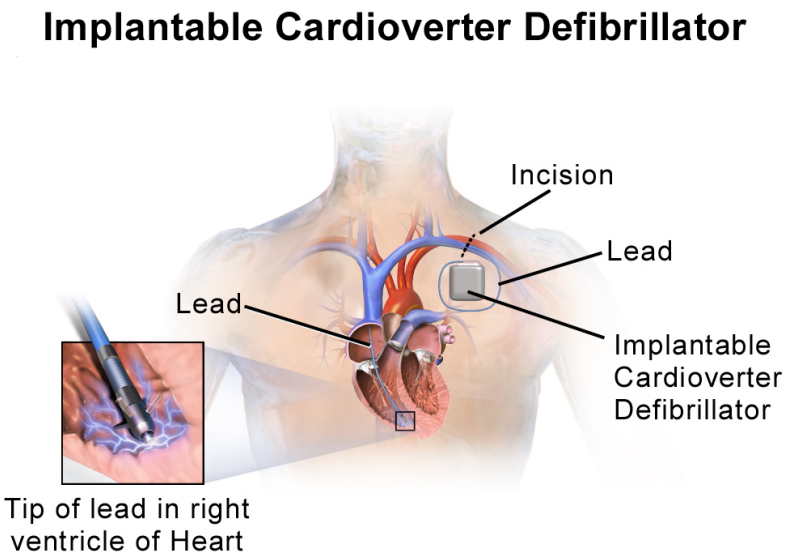


Figure 6.3: Illustration of Implantable Cardioverter Defibrillator (ICD)[3]

Compared to the single chamber ICD, the double chamber ICD makes it possible to restore the function of the atrium. It works either by stimulating the atrium before the ventricles, or by detecting its activity from the natural stimulator, the sinus node, and then to stimulate the ventricle in the event of failure. Triple chamber ICDs can also perform cardiac resynchronisation, in addition to the defibrillation. Cardiac resynchronization therapy is an effective therapy in patients with heart failure. Triple chamber ICDs are also called "CRT-D" (Cardiac Re-synchronization Therapy Defibrillators). In the following case study, we extracted all the patients who were implanted from the database, whatever the type of ICD. A specific focus will be made for the patients with CRT-D devices.

The number of implanted ICDs in France has been continuously increasing for two decades. Figure 6.4 shows the number of implantation procedures between 1991 and 2015 in France. The first publication concerning the implantation of an ICD in human dates from 1980 (Mirowski et al., 1980). The nineties saw the coming up of the technology, while the 2000-2010 decade witnessed its booming. The increase between 2000 and 2015 is almost linear. In 2015, the number of implantation procedures reached 10,904. It includes both first implantation cases and replacements. Replacements represented 14% of implantation in 2006 and 25% in 2015. Regarding first implantation, it raised from 5,300 new patients implanted in 2006 to 9,200 in 2013. Figure 6.4 also presents two projections of the number of implantation procedures for the upcoming years. The first projection is a linear regression based on the historical data from 2005 to 2015 (only the most recent decade), whereas the second projection is a sigmoid function that takes into account older data point from the nineties. The sigmoid function is particularly adapted to model the life-cycle of new technologies, from their very beginning to a market saturation. The linear projection is more suitable for the case of an unlimited market (or when the saturation effect is still far away) or when the technology is continuously improving (the interest for the product is renewed). The number of future implanted ICD is also mechanically influenced by the number of already implanted patients and the lifetime of the device. The more patients are implanted, the more ICD will be replaced.

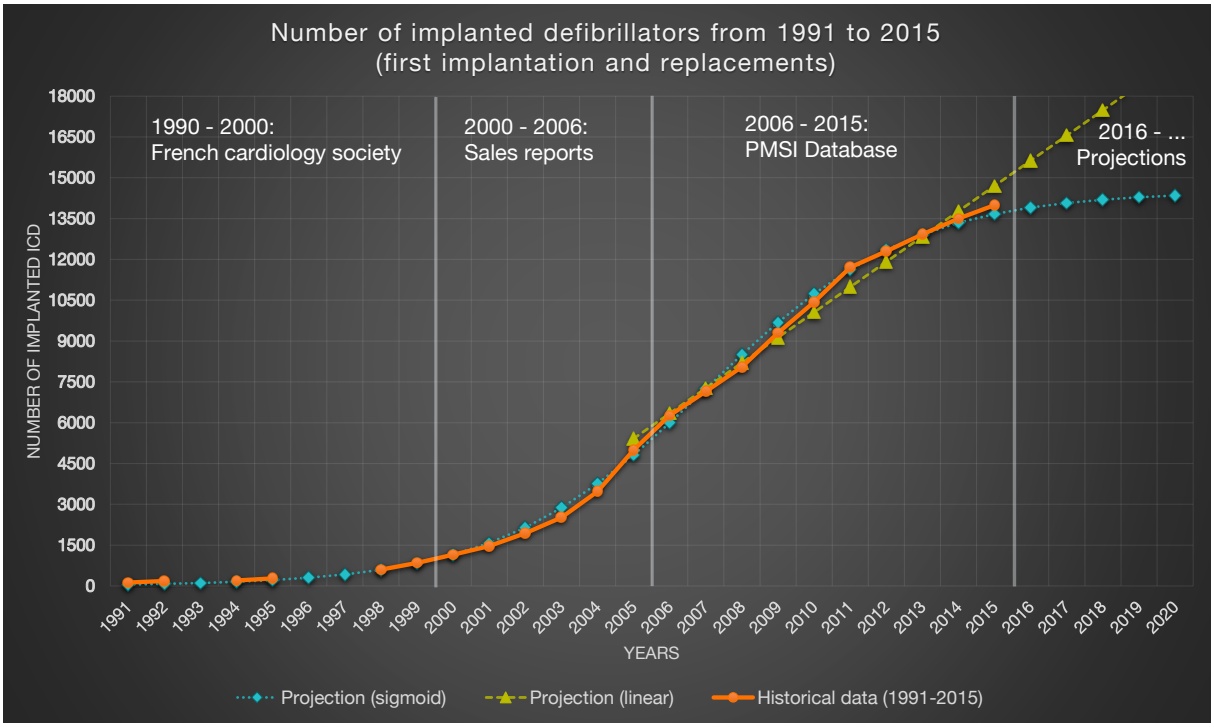


Figure 6.4: Number of implanted defibrillators in France from 1991 to 2015 and projections for upcoming years

6.2.2 Objectives

Based on the general context of cardiac arrhythmia and implantable defibrillators, our general objective is to use the PMSI database to discover the clinical pathway of patients suffering from such disease. Heart diseases have a significant impact on the health of patients. As a result, they are mainly taken care of at hospital, especially for acute episodes. It makes the PMSI database very suitable for their studies. Moreover, ICD implantation is a surgical act that is systematically performed at hospital. Regarding this specific case study, our objectives are the followings.

Objectives:

- Obj_1: Discover the sequence of events prior to a defibrillator implantation to search early signs. It would help shortening the delay before implantation.
- Obj_2: Build a model of clinical pathways to compare what happened (delays, acute events, follow-up) with medical expert's opinion. Deviations will be investigated.
- Obj_3: Assess the efficiency of ICD after implantation. It can be measured with the absence of cardiac relapses. A survival analysis will also be performed.
- Obj_4: Compare the actual delay between the implantation and the replacement of an ICD versus the theoretical lifetime (device maker).
- Obj_5: Evaluate new scenarios (an aging population, widening the indications for implantation, a new device on the market).

6.2.3 Data extraction

The PMSI database is a structured database which uses consensual classifications for the medical variables. The data extraction step is dedicated to the selection of the right patients from the database. It requires the selection of codes related to our case study. There are two groups of codes: (i) medical diagnoses related to arrhythmia, (ii) medical procedures related to an implantation surgery. Then, we need to identify any hospital stay related to the included patients.

Patient inclusion

Our objective is to study the clinical pathway of patients who were implanted an ICD. We are interested in their medical history before being implanted and in the post-surgery follow-up. To fulfill that objective, and based on the fact that we can use the PMSI data from 2006 to 2013[4], we decided to initially include all the patients that were implanted during the year 2008. It gives us a 2-year backward follow-up and a 5-year afterward follow-up.

First cohort: all the patients implanted in 2008. Patients were identified in the database thanks to the codes for the medical procedure: "Implantation of an automatic cardiac defibrillator". Experts from the field identified 8 codes corresponding to that definition is the French classification of medical procedures (Classification commune des actes médicaux [CCAM]). They are presented in Table 6.2. Each code corresponds to a precise procedure, with the surgical access and the number and the location of the probes. The tariff (of the procedure, not the device), as chosen by regulatory authorities, is also indicated. Finally, the type of implanted ICD is given. We used these CCAM codes to identify the patients that were implanted in 2008. The resulting cohort was made of **8,053 patients**. The distribution of the 3 types of ICD (simple, double of triple chamber) among these patients is presented in Figure 6.5. Simple chamber ICDs represent 1/3 of the implanted ICDs (2,684). Triple chamber devices are the most frequent with 39% of the cases (3,141), and double chamber ICDs account for 27% (2,174). The remaining 0.7% (54) of ICDs could not be assigned to any type of ICD, so they were removed from the study. Medical experts were also interested in getting a focus on the patients who received a triple chamber defibrillator (CRT-D). These patients are a subgroup of the 8,053 patients included. The challenge was to identify these patients with the highest possible certainty that they indeed received a triple chamber ICD. We cross-validated our patients selection by the presence of the right surgical procedure code (DELF014 or DELF020) and the presence of a bill for a triple chamber device. This cross-validation narrowed down the number the inclusion to **1,602 patients**. Still, this was preferable compared to the risk of including a patient with another type of ICD.

Second cohort: all the patients suffering from heart failure in 2008. The approach differs from the first cohort because we identified all the patients who suffered from at least one heart failure that led to a hospitalization. The identification of these patients with heart failure was made thanks to the "main diagnosis" and the "secondary diagnosis" fields of all the 2008 hospitalizations. In the International Classification of Diseases, "Heart failure" is referred as one of the 4 following codes: I50, I500, I501 and I509. Figure 6.6 shows the legend and the hierarchical structure of these 4 codes. I50 encompasses the 3 others. Using these codes in the PMSI database, we found **152,393 patients** for the year 2008.

[4]Due to changes in the regulation of data access, we could not include the years 2014 and 2015 in this specific analysis.

Table 6.2: Codes of medical procedures related to ICD implantation in the French classification

CCAM codes	Description of the medical procedure	ICD Type
DELF013	Implantation of an automatic cardiac defibrillator, with the insertion of a right intraventricular probe through a transcutaneous venous access	Simple
DELA004	Implantation of a cardiac defibrillator with epicardial electrode placement, by direct approach	Simple
DELF016	Implantation of an automatic cardiac defibrillator without atrial defibrillation function, with the insertion of an intra-transatrial probe and a right intraventricular probe through a transcutaneous venous access	Double
DELF014	Implantation of an automatic cardiac defibrillator with the insertion of an intra-atrial probe and a right intra-ventricular probe and a probe into a left cardiac vein through a transcutaneous vein	Triple
DELF020	Implantation of an automatic cardiac defibrillator with the insertion of a right intraventricular probe and a probe in a left cardiac vein through a transcutaneous venous access	Triple
DELA007	Implantation of a cardiac defibrillator	Unknown
DELF900	Implantation of an automatic cardiac defibrillator with atrial defibrillation function, with insertion of an intra-atrial probe and a right intra-ventricular probe through a transcutaneous venous access	Unknown
DEGA003	Removal of an implantable cardiac pacemaker or cardiac defibrillator	Unknown

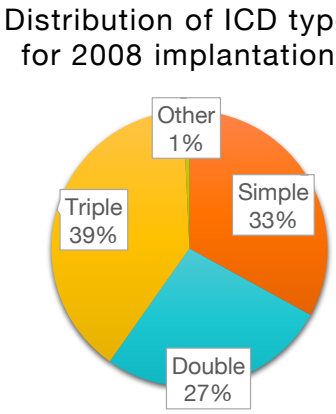


Figure 6.5: Distribution of the 3 types of implantable cardioverter defibrillators

Backward and afterward follow-ups

The two cohorts of patients were defined through an inclusion stay which happened in 2008. Then, our objective is to follow these patients across time. For that, we used the unique ID of each patient to find and extract all his/her hospital stays, whatever the hospitalization motive.

The first cohort, made of 8,053 patients, generated a total of **69,947 hospital stays** for a follow-up from

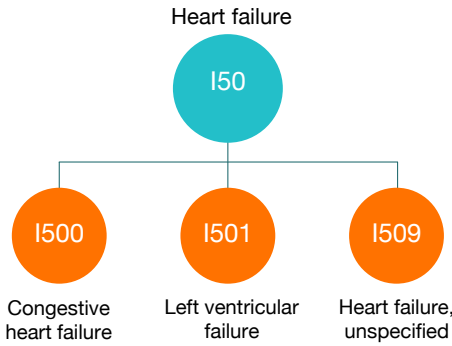


Figure 6.6: Hierarchical structure of heart failure codes in the ICD-10th)

2006 to 2013 included. Patients were followed 2 years (2006-2008) before the defibrillator implantation and 5 years after (2008-2013). The objective is to discover the clinical pathway of patients who were implanted, but also the pathways that led them to the implantation. The subgroup of 1,602 patients with a triple chamber ICD represents 21,170 hospital stays.

The second cohort, made of 152,393 patients, generated **997,648 hospital stays** during the period 2008-2014. We only extracted hospital stays which occurred after the first episode of heart failure in 2008. We did not look backward.

Labeling of events with medical diagnoses

The step that follows the extraction of all the patients' stays (= events) is their labeling. In process mining, each event must have at least 3 attributes: a case ID, a time-stamp and an event class (= a name). In our case study, the case ID is the patient ID and the time-stamp is the date of the stay. The **labeling of hospital stays** is the process of assigning an event class (= a character string) which best describes what happened. The assignment is made based on the event's attributes (duration, diagnoses, tariff, location, etc.). The same labeling must be applied to all the stays.

The labeling process

We identified two ways to perform the labeling: the first one is to use the already existing DRG classification, whereas the second one is to use the medical diagnoses. More classes could be generated by combining the diagnosis with the length of stay, the patient's age or the hospital location, but it would drastically increase the variability by making each stay unique. The advantage of the DRG classification is to rely on a consensual nomenclature which was produced and improved over several decades. It includes the work of many financial and medical experts. The classification algorithm takes into account all the hospital stay's attributes to produce a DRG. However, our objective is to study the different steps of each patient's pathway. In particular, we are interested in finding correlations between the different diseases that a patient may have, or finding a link between the elapsed time before a treatment is started and its efficiency. We came to the conclusion that DRGs were not the most suitable labels to that regard. Instead, we considered the medical diagnoses (main and secondary). They better express the patient's condition at a given moment (= the reason why he came to the hospital). In addition, we include the impact of the medical response (=medical procedure) to a patient's condition into the concept of clinical pathway state chart. It allows us to model the patients' condition and the response care separately.

The label of a hospital stay was chosen as the **main medical diagnosis**. A hospital stay can now

be summed up with its 3 minimal attributes. For example, *stay 1 refers to the hospitalization of patient 1 on the 1st of January 2008 and for left ventricular heart failure*. For each stay, the main diagnosis is a code selected from the International Classification of Disease. An advantage of this classification is to be hierarchically structured, so that we can choose different levels of aggregation. Once the label of each stay is obtained, we can study the individual pathway of each patient. A graphical illustration of 3 patients' sequences is presented in Figure 6.7. Each patient can have a different number of hospital stays in his sequence. We can see that the 3 patients had an ICD implantation, but not at the same moment. In this example, 7 types of stays are considered: ICD Implantation, I501 (left ventricular failure), I48 (atrial fibrillation), I200 (unstable angina), I42[0-2] (cardiomyopathy), Z450 (adjustment of cardiac devices), I251 (atherosclerotic heart disease), and two final events (death or end of record).

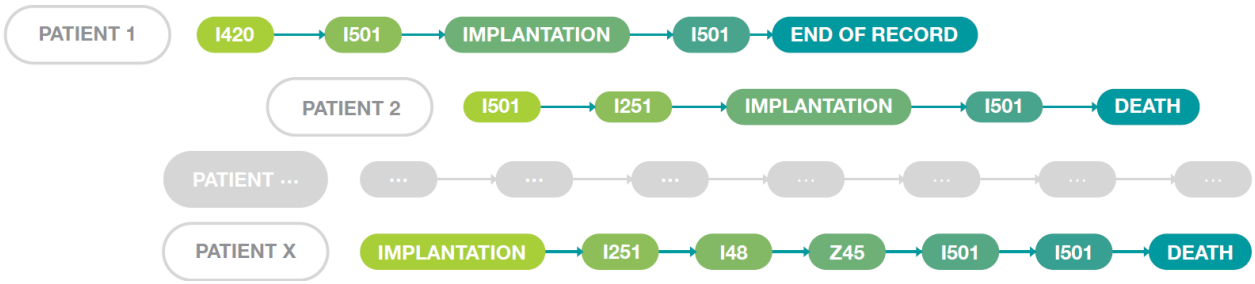


Figure 6.7: Graphical representation of individual clinical pathways after the labeling of stays

Event classes and labels

The labeling process is the most critical aspect of the data extraction. Based on the labeling method, the number of different event classes can drastically change. For instance, for the second cohort and its 997,648 stays, using the main diagnosis leads to 6912 classes. Note that existing works in the process mining literature rarely apply on case studies with more than a hundred classes. Regarding our first cohort with its 69,947 stays, the labeling with diagnoses generates 1,058 event classes. The labeling of the CRT-D subgroup of cohort 1 and its 21,170 stays generates 689 classes. The number of classes in the event log is the number one factor for the complexity of process discovery and of the resulting model.

Clusters of classes

Besides a complexity threshold on the model's size, we allow for some classes to be gathered in super-classes, called **clusters**, to create a compact clinical pathway model. Any class cannot be clustered with another. A domain-specific rule determines the feasible clusters. Here, classes are labeled based on the diagnosis, so we choose to allow the clustering of classes from the same medical specialty (digestive, circulatory, respiratory, nervous system, ...), as defined in the ICD-10th. It is done by comparing the digits of diagnosis codes (e.g. I501 and I505 can be clustered in I50). It narrows down the clustering possibilities (ex: 20 classes would generate more than 1 million possible clusters with no rule, whereas only 100 are feasible with our rule). A cluster is a pool of one or several event classes. The list of feasible clusters is known ahead of the optimization part (process discovery). The optimization balances between reducing the number of elements by merging classes in clusters, and losing precision. It means that all the clusters are not systematically used in the model.

Data preparation

The final part of data extraction is data preparation. The objective is to transform the data set of hospital stays that we just extracted into a ready-to-be-mined database. Basic data transformations were the following:

- Given the volume of hospital stays at stake, we remove any stay with a **missing value** for the diagnoses or the time-stamp ($< 1\%$ of data).
- We deal with **incoherence** in the data, such as overlapping stays (the first stay is not finished when the second starts). It may happen in practice when a patient is temporarily transferred into another facility before he returns to the original facility. In that case, the first stay encompasses the second one. We remove the second stay from the data set ($< 1\%$ of data).
- When a patient spends less than a day at the hospital, the length of stay is recorded as "0 days" (28% of the stays). We arbitrarily transform their length of stay to "half a day (0.5)", it is the minimal possible duration.
- When the length of stay is missing or negative (2.0% of the events), we arbitrary set the value as the minimum value of half a day.

Event classes filtering. The labeling of raw data induced a large number of event classes (ex: 6,912 classes for cohort 2). This is due to the inclusion of all the hospital stays of the patients, whatever the motive of hospitalization. It means that a stay for a broken arm will be included and will most certainly have no link with the heart condition. They are the **noisy stays**. For instance, in the CRT-D subgroup of cohort 1, among the 689 event classes, 233 of them (34%) only occurred once. It means that 1% of hospital stays (233 out of 21,170) account for 34% of the variability. The complete graph of the number of events represented depending on a filter on the number of classes is presented in Figure 6.8. It shows how only a few number of classes (24) are needed to represent two thirds of the events (66%). On this other hand, the gap in the number of classes needed to go from 90% to 100% of the events is tremendous (545 classes). This example illustrates a trend that was observed on all the data sets that we extracted from the PMSI. There is an important variety of hospital stays, even for patients suffering from the same disease. It emphasizes the need to apply an **event class filter** before starting the process discovery step. For the present case study, we used the following filters:

- Cohort 1 (entire): we keep the 197 most frequent classes (18.6%), which accounts for 90.5% of the stays (63,302 stays).
- Cohort 1' (CRT-D subgroup): we keep the 169 most frequent classes (24.5%), which accounts for 91.8% of the stays (19,431 stays).
- Cohort 2: we keep the 149 most frequent classes (2.2%), which accounts for 74% of the stays (734,155). For this cohort, the motivation for a strict filter is to balance the broad inclusion criterion ("heart failure") which gathers very heterogeneous patients.

Patients' attributes are derived from their medical history. In addition to the *age when implanted* and the *gender*, for each stay of a patient during the follow-up period, we search for the record of medical characteristics. We are interested in finding **comorbidities**, which are the presence of one or more additional diseases/disorders co-occurring with a primary disease. These comorbidities can be found thanks to the **list**

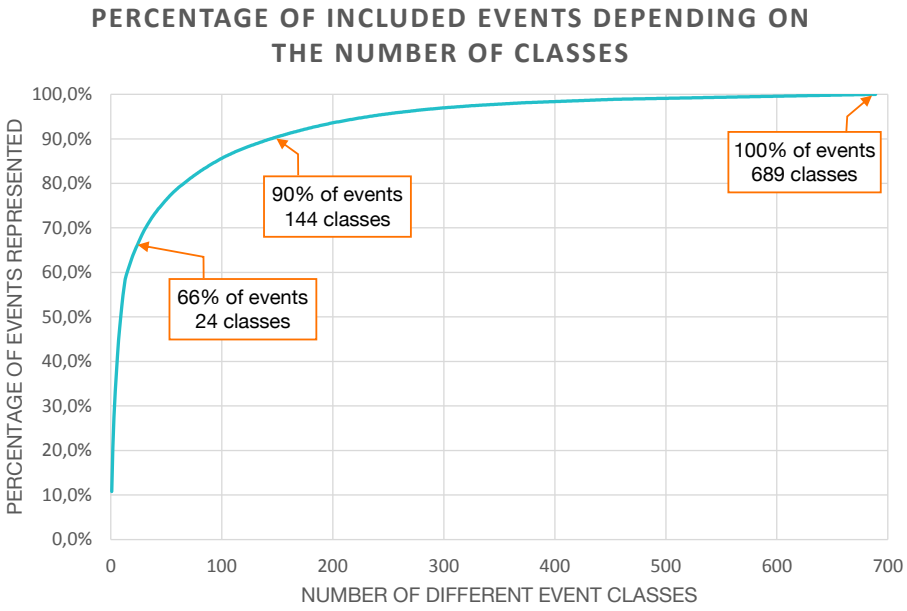


Figure 6.8: Most of the variability (= number of classes) is due to a small number of stays. The 10% of the less frequent events are labeled by 79% of the classes (545).

of other related diagnoses field. In (Quan et al., 2005), a complete list of codes for the identification of the comorbidities used to evaluate the Charlson[5] index is proposed. We propose a simplified version with 5 general comorbidities which can be used for any case study.

1. Hypertension
2. Diabetes
3. Obesity
4. Renal failure
5. Presence of a cancer

These 5 comorbidities can be chosen with more or less detail (e.g. presence of a metastatic solid tumor and diabetes with end-organ damage), and other comorbidities can be defined for each specific study (AIDS, liver disease, pulmonary disease, leukemia, etc.). The resulting data set, with patients' attributes, cleaned and labeled events, is our final event log.

Summary

The results of the data extraction are summarized in Table 6.3.

[5]The Charlson comorbidity index is a measure of the mortality for patients who have a range of comorbid conditions.

Table 6.3: Data summary

	Cohort 1	Cohort 1'	Cohort 2
Inclusion criteria	ICD implantation (all types)	CRT-D implantation, subgroup of cohort 1	Heart failure
Inclusion year	2008	2008	2008
Number of patients	8,053	1,602	152,393
Follow-up window	2006-2014	2006-2013	2008-2014
Raw extraction			
Number of stays	69,947	21,170	997,648
Number of classes	1,058	689	6,912
After filtering			
Number of stays	63,302 (90.5%)	19,431 (91.8%)	313,007 (31.4%)
Number of classes	197 (18.6%)	169 (24.5%)	8 (0.1%)

6.3 Process discovery

Process discovery is the starting point of the automatic modeling framework presented in this thesis. For the sake of clarity and space, we do not present the totality of the discovered process models for the 3 cohorts described above. The models related to the cohort 1 are not presented here, but are visually very similar to those presented for cohort 1'. In the following, we mainly focus on the results of cohort 1' (patients implanted with a triple chamber ICD) because they bring new striking information. We also present the result of cohort 2 as a preparation for the conversion into a simulation model.

Before we start mining the log of cohort 1', we asked their opinion to **medical specialists**. Indeed, by experience, cardiologists already know the main steps of ICD implanted patient's care pathway: (i) a severe heart failure, (ii) a device implantation, (iii) postoperative complications, (iv) device replacement, (v) another heart failure, and (vi) death (high mortality is observed). These steps are expected to be found in the future discovered process model.

6.3.1 Process discovery with our tabu search

The model discovered by our tabu search is illustrated on Figure 6.9. First observation is that the figure is graphically readable: the optimization process only kept most important paths to respect the size constraint (which fulfills objective 1 out of 5 of Section 6.2.2). Note that the top node, named "Start" is a virtual node that was added for graphical readability. It is not related to a hospital stay. This model also validates cardiologists' knowledge about the main steps (which fulfills objective 2) and provides more precision:

1. First, patients usually undergo a heart failure, and/or other cardiac issues (73% of patients had at least 1 stay in the last 6 months before implantation).
2. There is an average delay of 3 months and 1 week between this first event and the implantation surgery.
3. Patients are implanted of a CRT-D during an 8-day hospital stay.
4. A patient may be readmitted (49.7% of risks) on average 8 months and 2 weeks later for another heart failure (which fulfills objective 3).

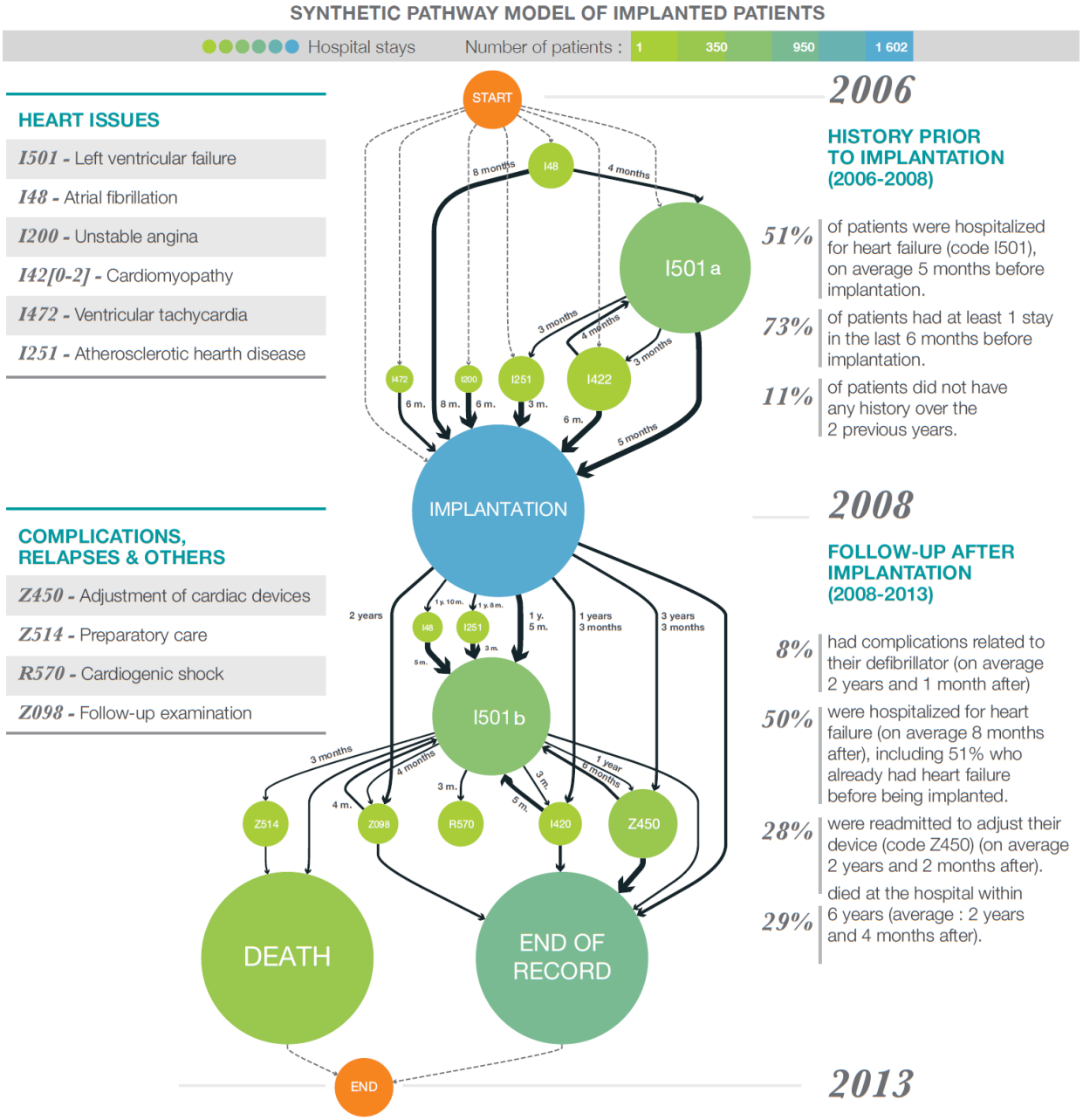


Figure 6.9: Heart failure process model for a size threshold of 50 (arcs+nodes)

5. 28.4% of patients are readmitted 2 years and 2 months after implantation for an adjustment of their cardiac device.
6. As most patients are between 65 and 75 years old, cataract surgery is often observed (8.5%) during the follow-up.
7. Death rate (at hospital) after 5 years is 28.7%. Death rate at home is unknown.

Additional outcomes about cardiomyopathy, tachycardia and ischemic disease were also discovered in the model (see Figure 6.9). In addition to the general benefit of the overall model, the most surprising discoveries about the clinical pathways were 2 folds for practitioners. First, it is the **important proportion**

of patients who were hospitalized at least once for heart failures, and often several months before implantation. This rises the question of improving the detection of eligible patients to reduce the delay before implantation, and thus mortality.

The second element is the seemingly **too short delay** between the implantation stay and the "adjustment of a cardiac device" (Z450). This type of stay is highly associated with a replacement of the defibrillator. First cause of replacement is the battery which is running out. Device makers suggest that the lifetime of defibrillator is between 5 and 8 years (based on the frequency of electric pulse). However, our model shows that the average delay between the implantation and the replacement is 3 years and 3 months (+/- 1 year and 10 months), median is 4 years and 5 months (which fulfills objective 4). It means that half of the patients who had an ICD replacement (N = 418, 26% of the cohort) do so before 4 years and 5 months after implantation, thus being under the lower bound of the supposedly lifetime. These median and mean values are underestimations of the actual values because we only followed the patients until December 31st 2013, which means that we could not observe later replacement. Thus, patients were followed from 5 to 6 years, depending on the date of implantation in 2008. Still, the proportion of patients whose device was replaced before 5 years is astonishing. The cost of a defibrillator fluctuates between 15,000€ and 20,000€. The question of its lifetime and of its replacement rate is important from an health insurance perspective.

6.3.2 Process discovery and replayability formulas

Using this case study on cardiovascular diseases and implantable defibrillators, it is also possible to discuss the added value of the replayability formulas proposed in Chapter 3 (optimal process discovery). In the following, we propose a qualitative comparison of the resulting models depending on the replayability formula. Figure 6.10 presents the models discovered using a fictitious data-set obtained by the log generator. Figure 6.11 presents the models generated using a real data-set from our defibrillator case study.

In both figures, the models generated using R^1, R^2, R^3 and R^4 scores are useless, since the optimal solution is reached with one cluster containing all classes and one arc. The model in Figure 6.11 contains two additional clusters since the predefined structure of the data forces all patients to have an implantation; all other classes are gathered into two clusters before and after implantation. Such result is not surprising (as demonstrated in Theorem 1 of Chapter 3).

The comparison between models using R^5 and R^6 scores is rather interesting. In Figure 6.10-R^6, the model contains more information, especially because of the representation of several sub-sequences (such as *C0-C38-C16-C36-C1-C29-C13-C59* or *C29-C3-C41*). This behavior is important to enlighten clinical pathways instead of isolated events, as presented in the model generated using the R^5 score. In the latter, the model is too "horizontal", with many very short sub sequences. Such phenomenon is denoted "Daisy flower" effect (van der Aalst, 2011) and should be avoided for the sake of clarity. In Figure 6.11, we can discuss the medical validity of both graphs taking into account the special medical features of our case study. In Figure 6.11-R^5, our algorithm is unable to determine the most representative pre-implantation care pathway, whereas it appears clearly in Figure 6.10-R^6: *Unstable angina-Heart failure-Cardiomyopathy* and *Unstable angina-Heart failure-Chronic ischemic cardiomyopathy* are known to be two coherent sub sequences of care leading to implantation when considering patients with cardiac diseases. The same remark holds for post-implantation care sub-processes, where *Atrial fibrillation-Heart failure-Chronic ischemic cardiomyopathy* clearly appears whereas similar clusters are not always connected with the R^5 score, which is considered as irrelevant by health-care practitioners.

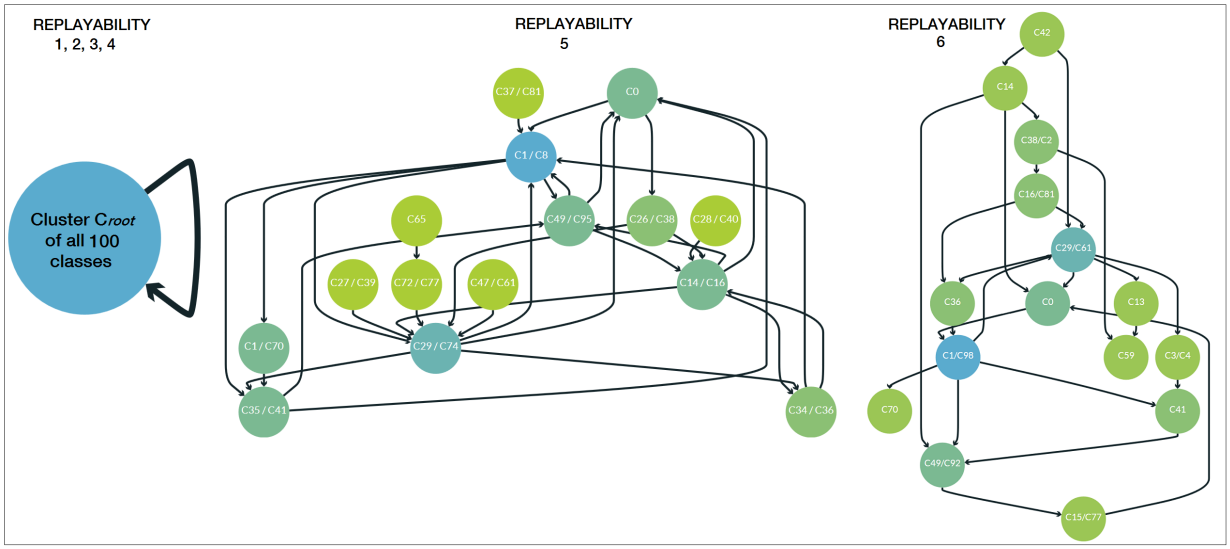


Figure 6.10: Process modes vs replayability criterion on a fictitious log with 100 classes, 10 000 patients and 150 000 events

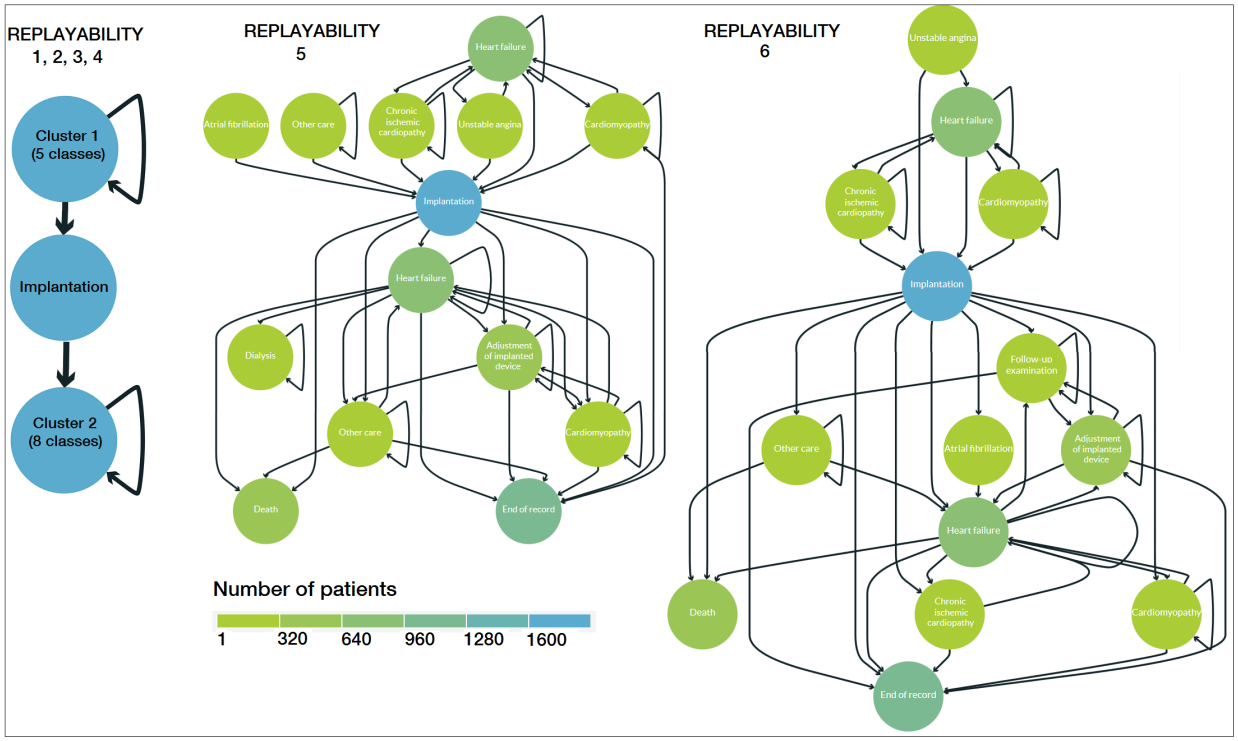


Figure 6.11: Process models vs replayability criterion on a real data set with 169 classes, 1,602 patients and 19,431 events

6.4 Process model enrichment

The second part of the automatic modeling framework presented in this thesis is **model enrichment**. For that, we used the health-care analytics toolbox presented in Chapter 4. We continue with the 1,602 patients of cohort 1' and the process model of Figure 6.9.

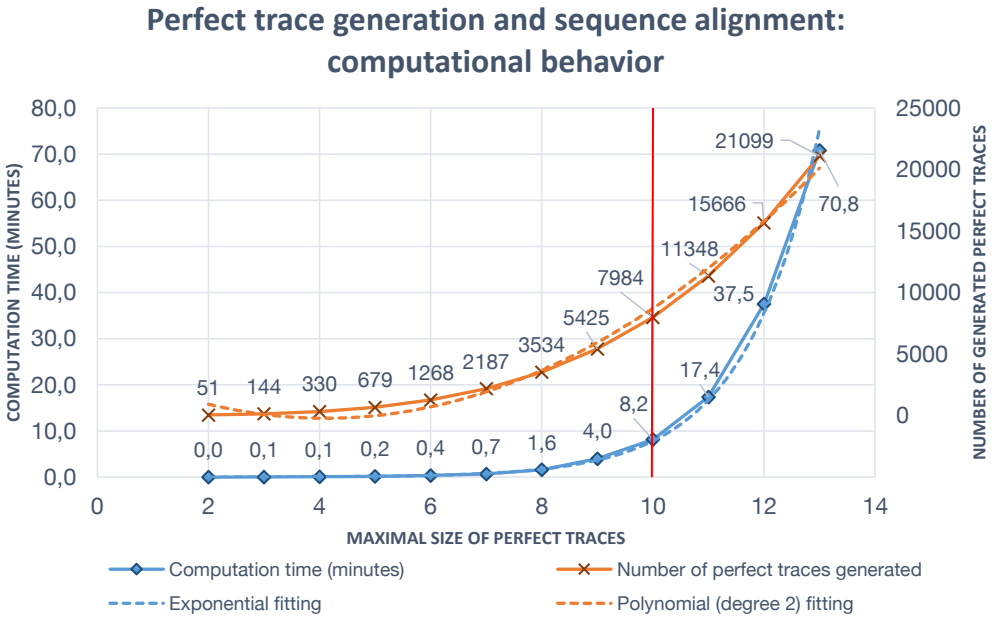


Figure 6.12: Computational behavior of the trace enhancement process (perfect trace generation and sequence alignment) on the event log of cohort 1'

6.4.1 Sequence alignment

Based on the discovered model presented in Figure 6.9, the generation of all the possible perfect traces was performed. **7,984 perfectly replayed traces** were obtained (maximal length of perfect trace was set to 10). The similarity matrix elements were calculated using the hierarchical structure of the International Classification of Disease. The class/class distance (or cluster/cluster or cluster/class) was defined as the inverse of the distance between the two elements in the classification (See chapter 4, section 4.4.2, Figure 4.3 for more details).

Then, each of the 1,602 original traces has been aligned with the perfect traces to find the closest perfect sequence. The computational time for this operation is linear in the number of traces in the log and exponential in the size of the model (i.e. the number of generated perfect traces). The overall process (perfect trace generation, then sequence alignment) required 8 minutes and 12 seconds on a 4Go RAM virtual machine running on Linux (Intel core i7 processor). Figure 6.12 shows the computational behavior of this process based on the threshold on the maximal length of perfect traces' sequence. The number of generated perfect traces is a **degree-2 polynomial function** of n, the maximal length ($O(n^2)$). The computation time for the trace generation and the sequence alignment is **exponential**. This is due to the simplicity of the alignment procedure: each original trace is aligned with every perfect trace to find the closest sequence. More advanced procedures could be developed to reduce the computation time. For instance, typical strategies like branch and bounds could save tremendous time (many perfect traces are extremely similar). Here, as this step needs to be performed only once for each case study, we could afford computation times between few minutes to few hours without being critical. Finally, we obtain a new enhanced log which contains 1,602 perfect traces.

6.4.2 Analysis of the routing choices

Algorithm selection

The analysis of the routing choices was made using two well-known machine learning algorithms: **Decision Trees** and **Random Forest**. For comparison purpose, we also tested the result of using a **Dummy** classifier, that-is-to-say a simple prediction method for our decision point problem. We implemented this algorithm in Python, using scikit-learn libraries.

- **Decision Tree** ("DecisionTreeClassifier"): Decision tree learning is a predictive model. In the tree, each internal (non-leaf) node is labeled with an input feature. The arcs represent filters on values of these input values. Each leaf of the tree is labeled with a prediction or a probability distribution over the possible predictions. Parameters of the method are: *the maximal depth of the tree*,*the minimal number of patients in a leaf* and *the splitting criterion*. More advanced parameters exist in most recent versions of the algorithms.
- **Random Forest** ("RandomForestClassifier"): A random forest is a meta estimator. It is composed of several decision trees, each of them being learned on a different sub-samples of the data set. Then, the random forest averages the decision trees' predictions to improve the predictive accuracy and to control over-fitting. Parameters are: *the number of trees*, *the aggregation method* (e.g. bootstrap) and the same parameters as the decision tree classifier.
- **Dummy** ("DummyClassifier"): it is a classifier that makes predictions using simple rules. Examples of rules are *stratified* (random predictions by respecting the class distribution of the data), random (totally random predictions) and *most frequent* (predicts the most frequent label of the data).

Other machine learning algorithms could have been used for this classification task, but we did not investigate this point further. In the remaining of this chapter, the following parameters were used. The dummy classifier follows a **stratified** rule. The decision tree uses a **Gini impurity** as a splitting criterion, **the maximal depth** is chosen to be equal to the number of input variables (5 in this case study), the minimum number of patients in a leaf is set to **5% of the number of observations**. The random forest is made of **100 trees**, built with a **bootstrap** strategy and each tree has the same parameters as the decision tree classifier. Our choices were motivated by our need to have generic choices that can be automatically reused without being manually defined. However, each machine learning task is an optimization process where the objective is to maximize the predictive capacity of the learned classifier, and the input parameters are the adjustment variables. Finding the optimal configuration of a classifier is not trivial and was not investigated here. We choose a set of rather standard values, without being too strict on the learning criteria. It enables the classifiers to find certain rules in the data.

Performance evaluation

Based on the clinical pathway model of Figure 6.9, 8 decision points were identified. They correspond to the nodes with at least two output arcs. For each decision point, the enhanced log has been transformed to obtain a set of observations (1 line = 1 trace) with the input variables (patients' features) and the output target (the next state). Patients' features (process-related and medical condition) are used as input variables for the prediction of the next state of their clinical pathway. The 3 classifiers (decision tree, dummy and random forest) were evaluated on 3 performance measures:

- The precision $= \frac{\text{true positive}}{\text{true positive + false positive}} \in [0-1]$

- The recall $= \frac{\text{true positive}}{\text{true positive + false negative}} \in [0-1]$
- The f1-score $= 2 \times \frac{(\text{precision} \times \text{recall})}{(\text{precision} + \text{recall})} \in [0-1]$

The objective is to maximize the 3 measures simultaneously. Precision represents positive predictive value and recall represents the probability of detection. The f1-score can be interpreted as a weighted average of the precision and the recall. Here, we use a "weighted" version of the f1-score, where f1 is computed for each class label, then their average is computed, weighted by the support (number of instance of each label). It can result in a f1-score that is not between precision and recall (as presented in the formula above).

Table 6.4 presents the results of the 3 classifiers on the 8 decision points of the clinical pathway model of Figure 6.9. For each decision point (= horizontal line), the highest value of each performance measure is highlighted. Random forest strictly dominates decision tree and dummy on 4 cases out of 8 (implantation, I501a, I501b and Z098) for the 3 measures. Decision tree strictly dominates the two others for the 3 measures on 1 case out of 8 (Z450). For the remaining 3 cases (I420, I48, I422), no classifier is dominant for the 3 measures. Random forest has the best values for 7 measures out of 9, decision tree has 4 (random forest and decision tree perform equally three time) and dummy has 1. We can conclude that **Random forest** outperforms decision tree on their predictive capability.

Table 6.4: Performance results of 3 classifiers for 8 decision points of a clinical pathway

Decision point (nb observations)	**Decision Tree**			**Dummy (historical probabilities)**			**Random Forest (100 trees)**		
	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score
Implantation (1537)	0.28	0.34	0.29	0.16	0.17	0.16	0.48	0.45	0.41
I420 (178)	0.72	0.72	0.72	0.58	0.58	0.58	0.75	0.71	0.62
I501.a (2688)	0.62	0.65	0.59	0.53	0.53	0.53	0.66	0.67	0.61
Z450 (360)	0.77	0.78	0.71	0.64	0.65	0.65	0.74	0.77	0.67
I501.b (3274)	0.57	0.64	0.50	0.45	0.46	0.46	0.61	0.65	0.54
I48 (144)	0.79	0.89	0.84	0.82	0.83	0.82	0.79	0.89	0.84
I422 (575)	0.62	0.61	0.51	0.52	0.52	0.52	0.62	0.63	0.60
Z098 (1327)	0.65	0.66	0.55	0.54	0.54	0.54	0.67	0.68	0.61

Still, it is worth noticing that the gap between random forest and decision tree is usually small (8% on average of all the measures), and that the absolute value of all the classifiers do not reach high and desirable levels (no classifier gets a precision or a recall higher than 89%). Further works could be dedicated to improve these results. Another interesting point is that decision tree performs much better than a dummy classifier. This is valuable because, in addition to its predictive capability, the quality of a classifier also relies in its **interpretability**. The prediction made by a random forest can hardly be interpreted as it is based on the averaging of multiple decision trees that were learned in a specific manner. On the contrary, a decision tree provides a set of explicit rules that entirely explain how each prediction is made. The choice of a decision tree classifier is relevant when looking for a good predictive level and for explicit rules, whereas random forests are indicated when trying to maximize predictive performances.

Figure 6.13 illustrates the advantage of using a decision tree classifier to discover the underlying rules of the predictions. It represents the decision point of state *implantation*. The question to solve is "what will be the very first hospital event of a patient after his/her implantation?". At the top of the tree, the root node represents all the historical observations for the studied decision point (1,537 patients out of 1,602 because 65 had no further stay after implantation). The splitting criterion is the Gini index and each split is binary. An example of a rule that can be derived from the tree is:

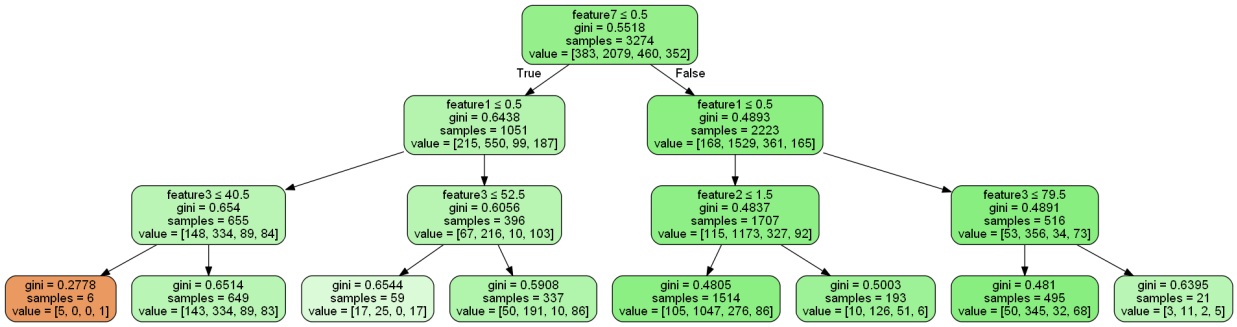


Figure 6.13: Knowledge discovery with the explicit rules of decision trees

"A patient with $feature5 = 0$ (i.e. the patient does not have diabetes) and $feature1 = 0$ (no hypertension) and $feature3 > 40.5$ (age greater than 40.5) can have one of 4 next states with the probabilities: 51.5% for state *heart failure* (334/649), 22.0% for state *end of record* (143/69), 13.7% for state *adjustment of cardiac devices* (89/649) and 12.8% for state *cataract surgeries* (83/649). This is the second most bottom left leaf of the tree."

The knowledge of the rules used in the predictive model is valuable. It helps finding unknown correlation, or on the contrary it helps validating preconceived ideas about the clinical pathway.

6.4.3 Time perspective

The time perspective was added to the model using classic distribution fitting method on the historical data. The theoretical distributions that were obtained for the lengths of stay are presented in Table 6.5. The most frequently used distribution is the "log-normal" distribution. The same fitting procedure was applied to obtain the duration of all the arcs of the model.

Table 6.5: Distribution fitting for the length of stay of each state of Figure 6.9 process model

State	**Theoretical distribution**
I48	-0.5 + lognormal(4.55, 5.3)
I42[0-2]	normal(7.3, 6.21)
I501a	-0.5 + weibull(8.23, 1.34)
I251	-0.5 + lognormal(5.35, 7.09)
I200	-0.5 + lognormal(6.08, 7.49)
I472	-0.5 + lognormal(7.34, 8.47)
Implantation	0.5 + lognormal(7.44, 7.05)
I501b	-0.001 + exponential(8.78)
Z450	-0.001 + exponential(2.69)
Z514	-0.5 + lognormal(1.15, 1.17)
Z098	-0.5 + lognormal(1.21, 1.19)
R570	-0.001 + gamma(20.2, 0.606)

6.5 Simulation of clinical pathways

We perform the conversion of the previously discovered process model into a simulation model. The model is validated with 5 Key Performance Indicators. Finally, the model is used to perform a sensitivy analysis on patient features and to evaluate new scenarios of implantation strategies.

6.5.1 Model creation

The clinical pathway model of Figure 6.9 is a process model in the form of a causal net. We uses the conversion procedure presented in Chapter 5 to obtain a Clinical Pathway State Chart $CPSC = (S, V, \zeta, \tau, p, q)$. S and V are directly derived from the nodes and arcs of the causal net, ζ is made of the decision trees described above (Figure 6.13) and τ was obtained with distribution fitting (the wait states part of τ is described in table 6.5). The two last elements of the CPSC are p and q. They were obtained from the historical data and they are presented in Table 6.6. In the current case study, the p and q function have special values because all the patients have at least the implantation state. Then, all the prior care states have a probability of zero to be the end of the pathway and all the following care states have a probability of zero to be the starting point of the pathway. In addition, as we added a care state "death" and a care state "end of record", they are the only two care states with a stopping probability strictly greater than 0 (exactly equal to 1). These two care states have a null length of stay. 8 care states (I48 after ICD, I251 after ICD, I501b, Z514, Z098, R570, I420, Z450) are not displayed in Table 6.6 because their starting and stopping probabilities are both null.

Table 6.6: Starting and stopping probabilities of the Clinical Pathway State Chart

Wait States	**Starting probability**	**Stopping probability**
I48 (before ICD)	6.5%	0
I472	3.9%	0
I200	4.7%	0
I251 (before ICD)	11.1%	0
I422	11.0%	0
I501a	49.2%	0
Implantation	13.6%	0
Death	0	1
End of record	0	1

6.5.2 Model validation

The model was validated using the 5 Key Performance Indicators presented in Chapter 5.

- **KPI #1:** The average time spent in care-states by a patient
- **KPI #2:** The average time spent in wait-states by a patient
- **KPI #3:** The average number of visited care-states by a patient
- **KPI #4:** The average number of times that state s_i was visited by a patient, $\forall i \in S$
- **KPI #5:** The average number of different entities that visited state s_i at least once, $\forall i \in S$

The results for all the KPIs are presented in Table 6.7, based on the simulation of 100,000 patients. Regarding **KPI-1 and KPI-2** (time related measures), the validation was challenging because of the large variability of these measures in the original data: historical mean for KPI-1 is 65 days, standard deviation is 88 days. Historical mean for KPI-2 is 4 years and 1 month, standard deviation is 2 years and 1 month. The simulation model seems to underestimate the time spent by patients in care states (KPI-1) and in wait states (KPI-2) when using the mean and the standard deviation. However, the simulation results show a significant decrease in the variability (standard deviation) compared to historical data. The high variability of the data is explained by the presence of some outsiders (e.g. a patient spent 4 years at hospital). We think that a variability reduction is an asset for the simulation model. Based on these results, we think that KPI-1 and KPI-2 would require a different validation method, such as a more advanced distribution comparison. It would provide more reliable conclusions that the current mean and standard deviation.

Table 6.7: Validation results for 5 measures (100,000 simulated patients)

KPI	**Historical data Mean (+/- STD)**	**Simulation model Mean (+/- STD)**	**Simulation model 95% CI**
KPI #1	65.80 days (+/- 88.10)	45.07 (+/- 29.18)	+/- 0.15
KPI #2	4 years 1 month (+/- 2 years 1 month)	3 years 8 months (+/- 9 months)	+/- 1.55 days
KPI #3	13.2 care states (+/-18.8)	11.7 (+/- 4.8)	+/- 0.025
KPI #4	Figure 6.14	68.5%	-
KPI #5	Figure 6.15	75.0%	-

The validation of the CPSC is more straightforward for the remaining KPIs. Regarding **KPI-3**, we obtained a close value of the number of care states in a trace sequence (11.7 versus 13.2). **KPI-4 and KPI-5** are presented in detail in Figure 6.14 and Figure 6.15 respectively, but only for care states (not wait states). For each care state, the histogram shows the historical data (orange), the simulation result (blue) and the 95% confidence interval (red line). Based on a binary validation approach, the simulation model gets a validation score of 68.5% for KPI-4, and 75.0% for KPI-5, which is above regular thresholds (50% or 66% for binary validation). Even if the model passes over the regular thresholds, it presents the same behavior of a slight underestimation of the number of patients in each state (blue lines are lower than orange lines). We found a possible explanation for this shift. After a thorough examination of the decision points rules (decision trees), it appears that the routing probabilities toward "death" and "end of record" are slightly overestimated, making patients' sequences ending too early. A solution would be to study the sensibility of the results regarding such variables and adjust the values using a simulation-optimization approach. This conclusion matches our previous discussion about the need for an optimized tuning of the data mining algorithm (not provided here). It would result in a model adjustment leading to higher validation scores.

6.5.3 Sensitivity analysis

A sensitivity analysis of input parameters was then performed for the simulation model described above, as described in Chapter 5. The input parameters are the patient features available in the case study data. It includes the 5 comorbidities described in Section 6.2.3, 2 non-medical patient characteristics and 1 variable related to defibrillators:

1. **Patient has hypertension**
2. **Patient has diabetes**

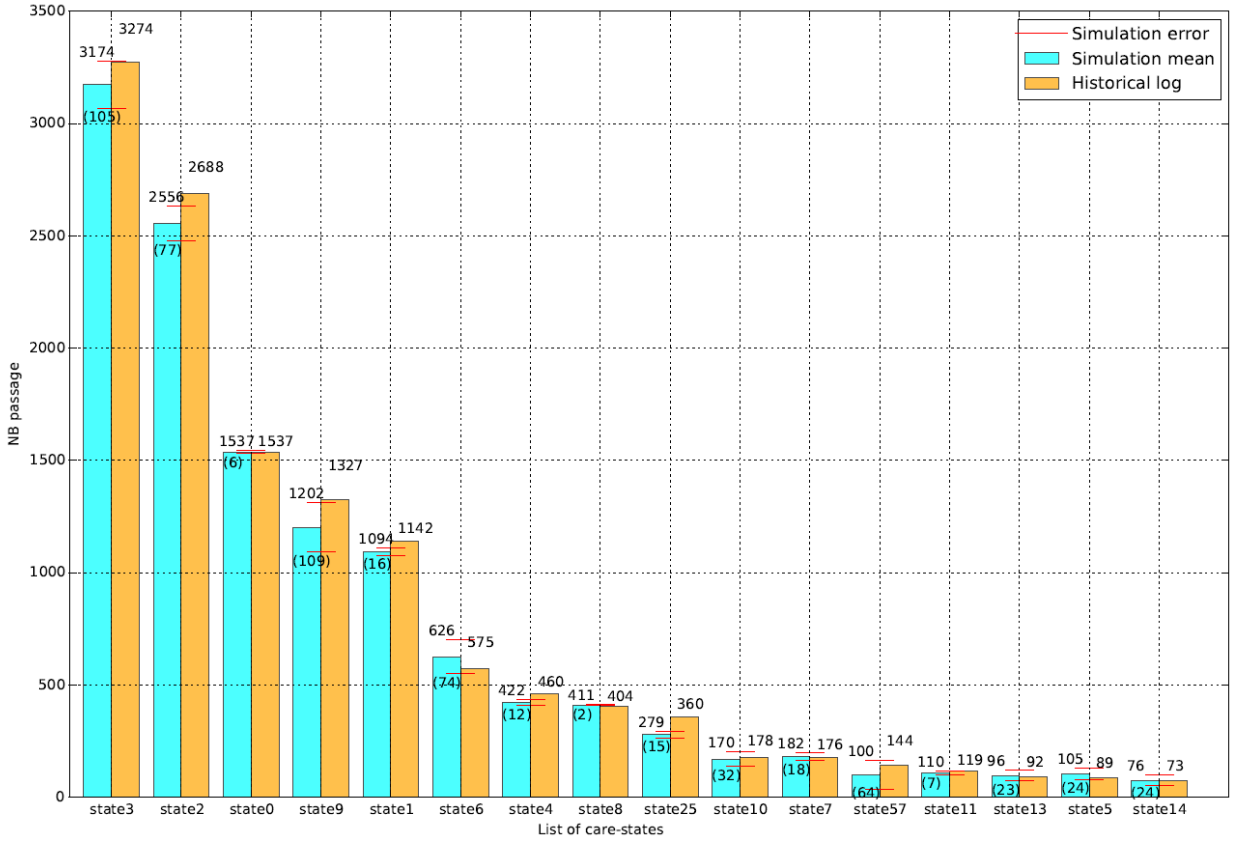


Figure 6.14: Validation of the CPSC on KPI#4. States legend: 0 (implantation), 1 (end of record), 2 (I501a), 3 (I501b), 4 (death), 5 (I200), 6 (Z450), 7 (I420), 8 (Z098), 9 (I422), 10 (I251), 11 (I48-before), 13 (I472), 14 (I48-after), 25 (Z514), 57 (R570)

3. **Patient is obese**
4. **Patient has kidney failure**
5. **Patient has a cancer**
6. **Age of the patient at the first ICD implantation**
7. **patient gender**
8. **Replacement rate**

The replacement rate represents the chances that a patient's ICD is replaced during post-implantation follow-up. It corresponds to the routing probabilities of having the state Z450 (from any other state connected with a transition).

The sensibility analysis is performed once for each KPI. A total of 103 graphs (KPI-1, KPI-2 and KPI-3 require one graph each, whereas KPI-4 and KPI-5 require 50 graphs each, one for each care state and one for each wait state). Most of the 50 graphs related to KPI-4 are similar, and so are those of KPI-5. We present 4 of the most remarkable results, which means a striking impact of the input variables on the output KPI or its total absence, in Figures 6.16, 6.17, 6.18 and 6.19. For each parameter setting, 10,000 patients were simulated (confidence intervals are shown on the graphs).

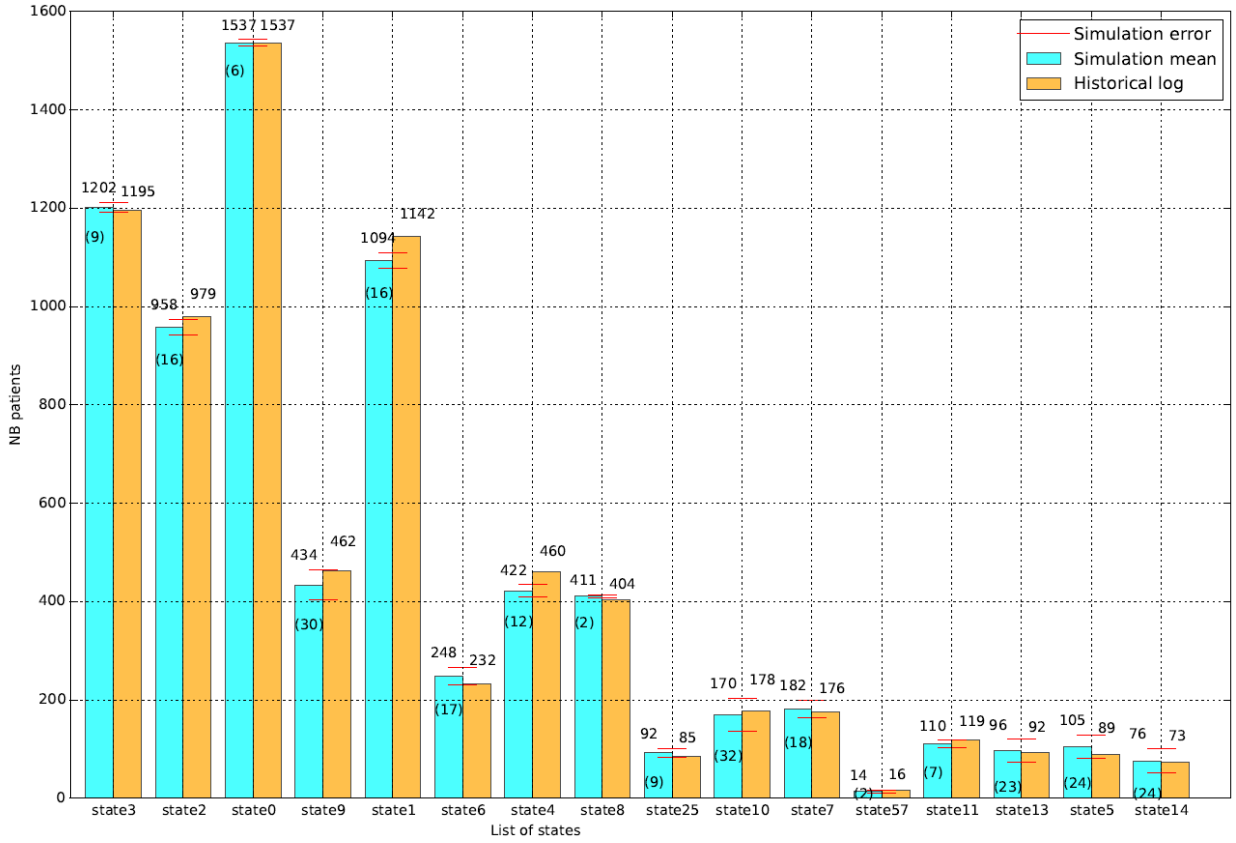


Figure 6.15: Validation of the CPSC on KPI#5. States legend: 0 (implantation), 1 (end of record), 2 (I501a), 3 (I501b), 4 (death), 5 (I200), 6 (Z450), 7 (I420), 8 (Z098), 9 (I422), 10 (I251), 11 (I48-before), 13 (I472), 14 (I48-after), 25 (Z514), 57 (R570)

Results (1/4). Figure 6.16 shows the result of the sensitivity analysis on KPI-1, the total time spent by patients in care-states. The impact of the 8 input variables is displayed on the same graph (8 curves), even if each variable varies independently (anything else equal). The y-axis represents the possible values of KPI-1 and the x-axis represents variations on the input variables. In order to plot and to easily compare the 8 curves, we normalized the possible values of each variable. The baseline point is when the modification coefficient of all variables is 1 (Green Arrow).

Among the 8 inputs, only two influence the time spent by patients in care-states: the age at implantation (red line) and the presence of kidney (grey line) failure. First, the impact of kidney failure is linear. The fewer patients have kidney failure (caution, a high coefficient of this input variables actually means fewer patients have it), the shorter the total time spent in care-states (i.e. at hospital) will be. It can be explained by the necessity of having very regular dialyses sessions (half a day) when having kidney failure. Regarding the age of implantation, the shape of the curve appears more atypical at first sight. Starting from the left, there is a fast increase in KPI-1 when the implantation age increases, then it stagnates, and it finally slowly decreases. This shape illustrates the fact that an increase in age is totally correlated with the need for more cares (the initial increase). However, after a certain threshold (mean age at implantation is 75), the need for care on a 4-year term decreases because patients die faster.

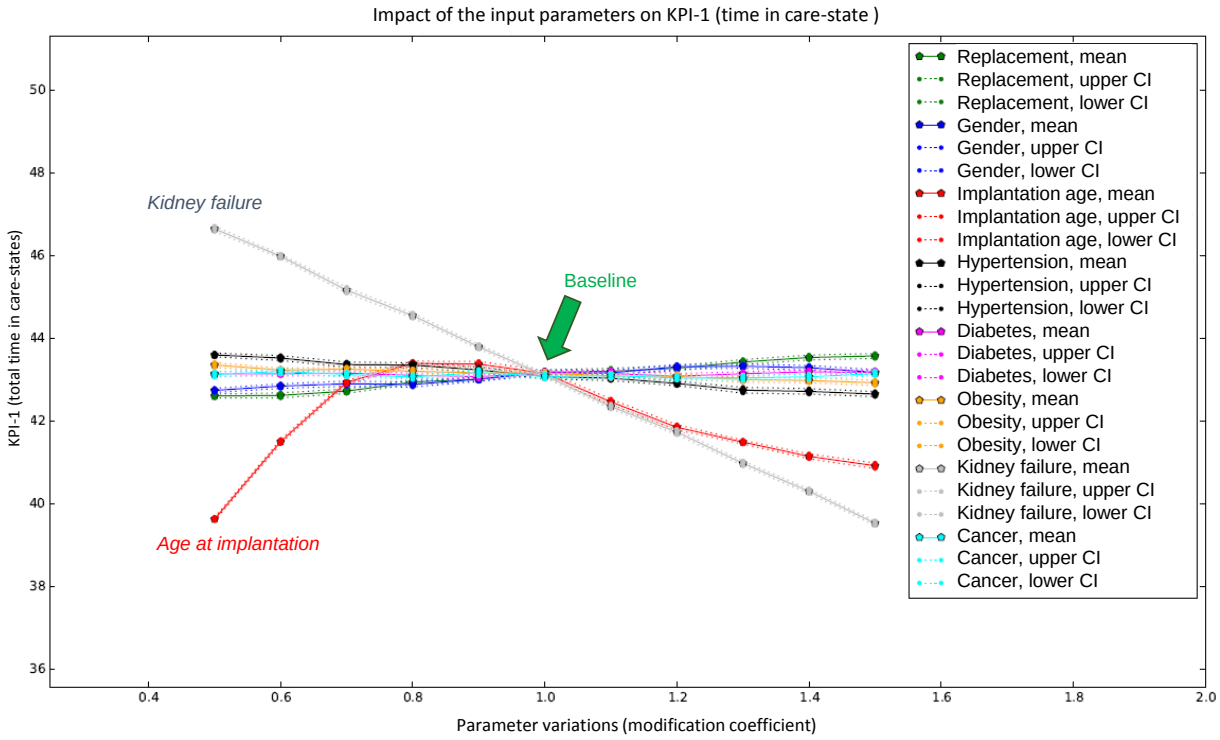


Figure 6.16: Sensitivity analysis result - impact of 8 input variables on KPI-1

Results (2/4). Figure 6.17 shows the result of the sensitivity analysis on KPI-4, the number of times that state *cardiomyopathy before implantation* was visited by a patient. The outcome values are standardized for 1,602 patients (even if 10,000 patients were simulated). For this KPI, it is interesting to notice that no input variable significantly impacts the output values. It means that such cardiac issues are not dependent on factors that we incorporated in the model. A more in-depth backward analysis of patient history might turn out more relevant (more than 2 years before implantation).

Results (3/4). Similarly to the previous graph, Figure 6.18 shows the result of the sensitivity analysis on KPI-4, the number of times that state *cardiomyopathy after implantation* was visited by a patient. This time, two input variables show a direct impact on the output values: the age at implantation and the replacement rate. An increase in the age of patients when being implanted induces a substantial decrease in the number of times they have a cardiomyopathy (red line). This is probably explained by an edge effect of the long-term follow up of patients (4-5 years). Older patients with severe heart conditions have "less time" to develop other issues as the 2-year death rate is extremely high for patients over 75 years old.

Regarding the replacement rate, an increase (i.e. more patients have a defibrillator replacement after few years) induces a linear decrease in the risk of having a cardiomyopathy (green line). It shows the importance of a close follow-up of patients and of anticipating the device malfunctioning.

Results (4/4). Figure 6.19 shows the result of the sensitivity analysis on KPI-4, for a care state which is not directly related to heart issues: the *cataract surgery* (eye troubles). As a side effect of the cohort characteristics (mean age at implantation is 66 years old), lots of patients require cataract surgery. It is known as being very predominant in elderly people. We find the same results: the age at implantation

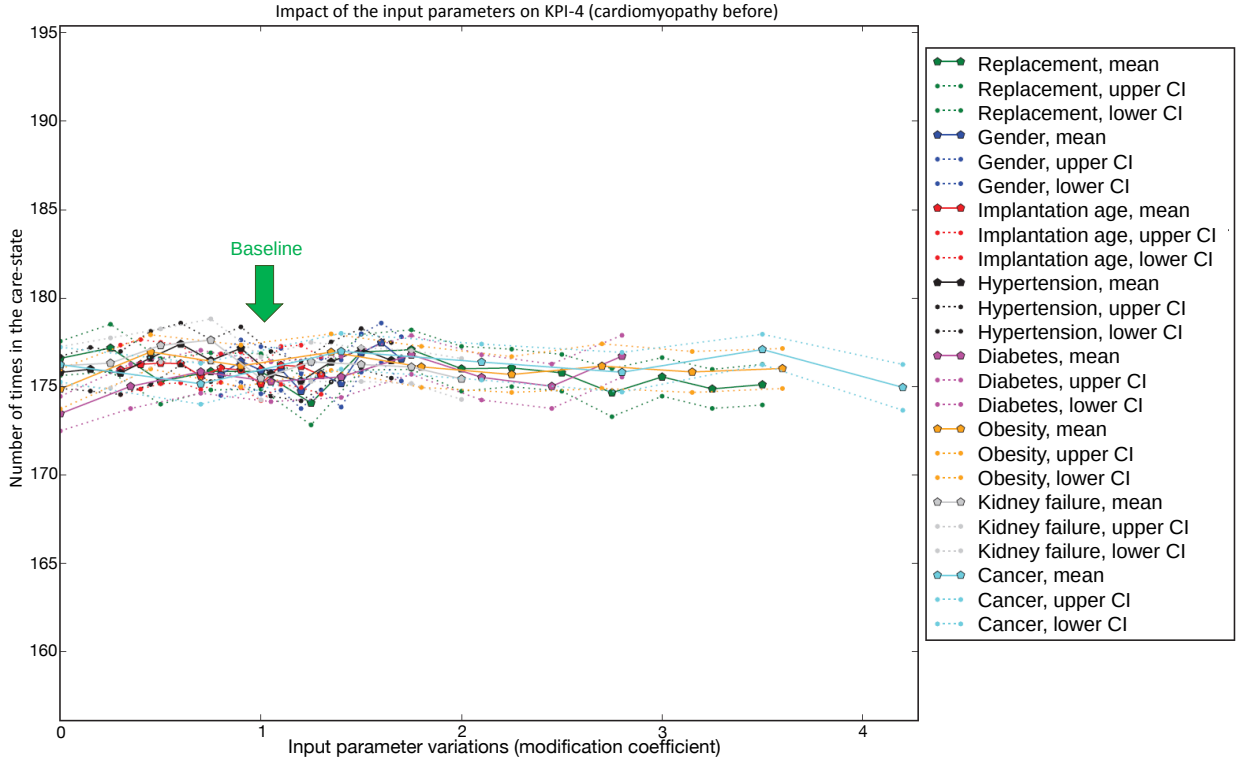


Figure 6.17: Sensitivity analysis result - impact of 8 input variables on KPI-4 (a)

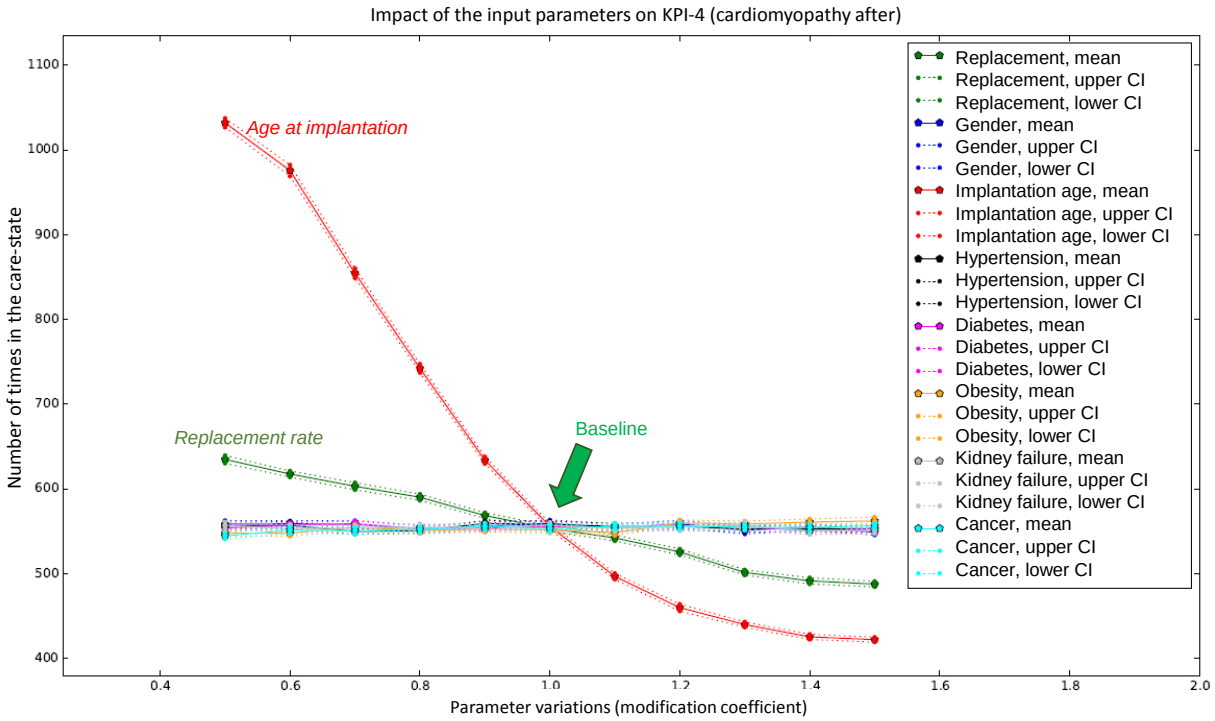


Figure 6.18: Sensitivity analysis result - impact of 8 input variables on KPI-4 (b)

is highly correlated with the need for more surgeries (red line). The relation between kidney failure and

cataract is unclear to us, even if a positive linear relation is observed (grey line). An interesting point is the relation between the replacement rate and the number of surgeries (green line). The more patients have an ICD replacement, the more they have a cataract surgery. This can be explained by the lengthening of the life expectancy of patients with a replaced device.

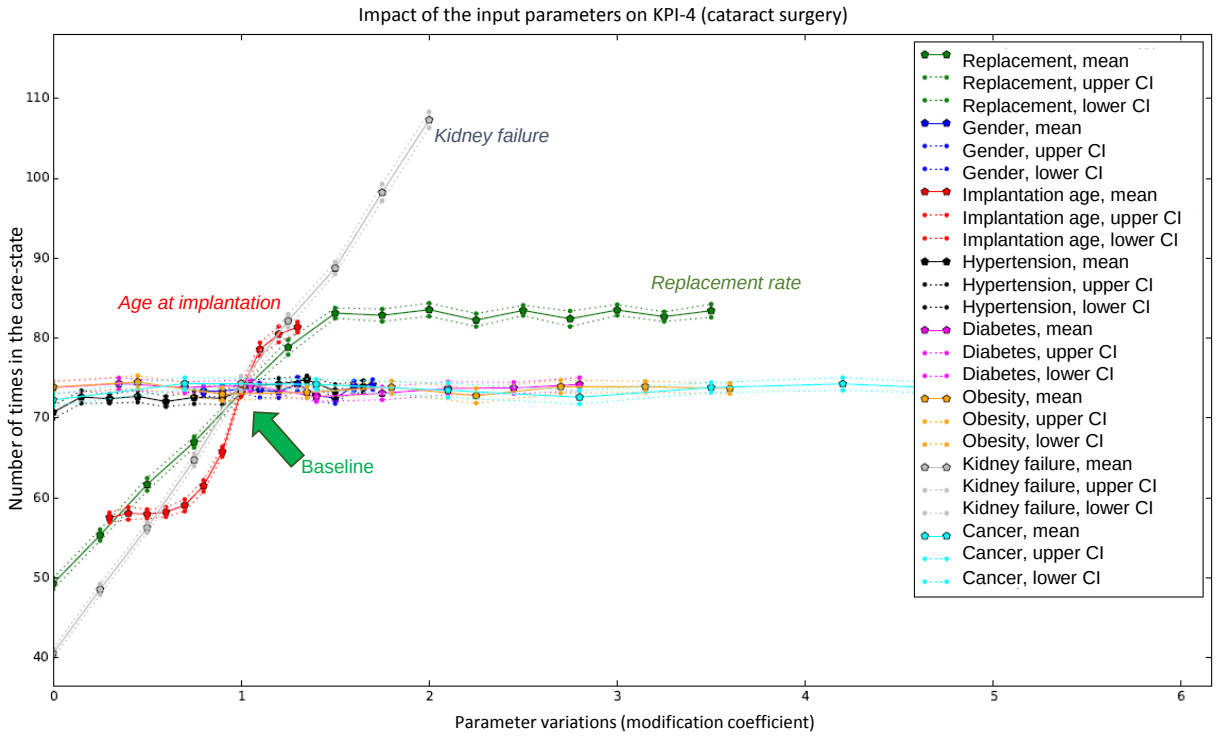


Figure 6.19: Sensitivity analysis result - impact of 8 input variables on KPI4 (c)

6.5.4 Scenarios evaluation - new implantation strategies

The ultimate use of the simulation model is to evaluate new scenarios. Based on our case study on defibrillators, a question that arose from experts of the medical field concerns the implantation strategies. They asked about the potential impact of opening the criterion for patients to be eligible for implantation. For that purpose, we use the cohort 2 of the 152,393 patients who had a heart failure. Then, we compare the performances of using different implantation strategies. A strategy is composed of two elements: a ratio of patients that are eligible for 1st implantation and a ratio of patients eligible for replacement.

Clinical Pathway State Chart Creation After the data extraction (Section 6.2.3), we created a Causal Net using a Process Mining approach. For this specific part only, the resulting clinical pathway state chart, obtained after using our conversion algorithm was implemented in Anylogic 7.2.0 software. It is shown on Figure 6.20. Figure 6.20-a shows the CPSC. Its 8 care-states are depicted by yellow boxes (first yellow box is excluded as it is the common entry point): (1) I501a for the *first left ventricular failure*, (2) I472 for *ventricular tachycardia*, (3) I500 for *congestive heart failure*, (4) I509 for *unspecified heart failure*, (5) I422 for *other hypertrophic cardiomyopathy*, (6) I501b for a *relapse of left ventricular failure*, (7) *deceased* and (8) *end of follow-up*. A financial cost is assigned to each care-state. It includes human, material and facility costs to take care of the patient during his/her hospitalization. The 20 wait-states are depicted by arrows

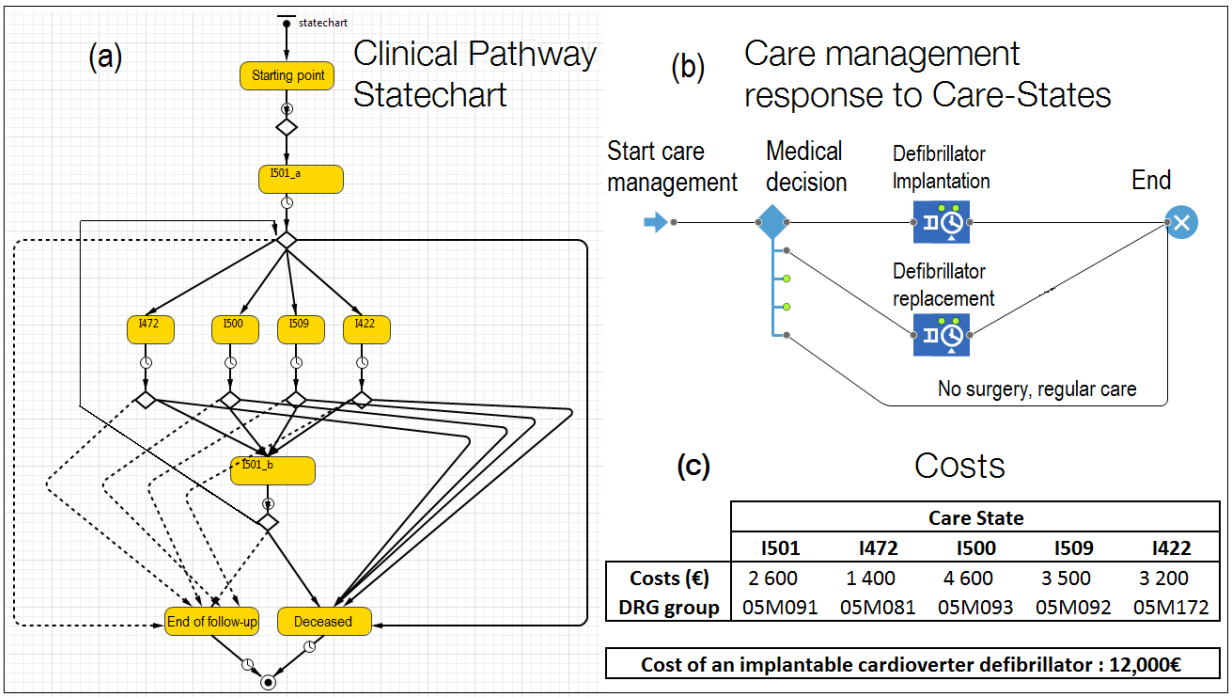


Figure 6.20: (a) Clinical pathway state chart used in the heart failure case study (Anylogic software screenshots), (b) Care process triggered by certain care-state, (c) Table of the costs (based on http://www.aideaucodage.fr/ghm)

(solid and dotted lines). On Figure 6.20-b, we show a simple model of the medical decision triggered when a patient is in one of 4 specific care-states (1,2,3,4). A physician decides to implant or not a cardioverter defibrillator to the patient to prevent a cardiac arrest. If the patient was previously implanted, the physician may decide to replace the device and to implant a new one (lifetime of a defibrillator is 5 to 8 years). The third possible decision is to not implant the patient but to hospitalize him/her for regular care (nursing, monitoring and drugs). Decisions of implantation and replacement are based on probabilities observed from data history. These 2 probabilities will be studied as variable inputs in the simulation experiments.

The last components of CPSC to define are the functions ζ and τ. Transition probabilities represent the risk for a patient to switch from his current state to another state. It is the risk of being readmitted at hospital later for another issue. In terms of patient's health condition, lower probabilities are always better. Only the transition probability toward care-state number (8)-*end of follow-up* shall be high for a better outcome: the patient was cured and will have no more adverse event. Implanted Patients do not have significantly different lengths of stay compared to never implanted patients, so τ is assumed the same for all patients. However, major differences in transition probabilities were observed between the 3 groups of patients: those never implanted, those implanted once and those implanted and replaced. Data history shows that implanted patients have lower risk of readmission compared to not implanted patients. It tends to show that implantable defibrillators have positive effects on the patients' health condition. Similarly, replaced patients have a slightly lower risk of readmission than implanted patients. The underlying reasons might be correlated to factors out of the scope of the current study (age of first implantation, device lifetime or type of technology). Different transition probabilities are used for the 3 groups. Distributions used for τ were found using a best fit tool based on the mean squared error. Distributions for each of the 8 care-states are (unit is days): (1) $371 \times Beta(1.2, 46.4)$, (2) $213 \times Beta(0.688, 24.1)$, (3) $Weibull(11.6, 1.18)$, (4)

$-0.5 + Weibull(12.4, 1.28)$, (5) $-0.5 + LogN(5.86, 8.22)$, (6) $-0.5 + Erlang(4.77, 2)$, care-states (7) and (8) have no distribution by definition.

Experimentation and Results The previously defined Clinical Pathway State Chart models the evolution of patients' health condition in heart failure. It also models the medical decision to implant or not a patient with a defibrillator. Such decision impacts on the probability of adverse events. An implantable cardioverter defibrillator costs between 10,000€ and 16,000€, whereas hospitalizations for a heart issue cost between 1,000€ and 4,000€ euros depending on the severity (See Figure 6.20-c). Now, we show how we used our simulation model to study several care management scenarios that balance costs and care quality. The performance of scenarios is assessed by evaluating 3 key performance indicators (KPIs): (1) total cost incurred by all patients, (2) death rate and (3) proportion of patients who had a heart failure relapse. All three KPIs are measured after a fixed simulation time of 5 years.

We specifically studied the variation of two parameters: **first implantation probability** (i.e. the medical decision to implant the device when in care-States (2)(3)(4)(5)) and **replacement probability** (i.e. the medical decision to replace the device in the same care-states). Both parameters varied between 0 and 0.5 with a 0.1 step. In order to ensure the statistical validity of our results, we performed several replications of each parameter setting. The number of replications was chosen large enough to ensure a 95% confidence interval on the 3 KPIs. It is set to 40,000 patients. Results of the simulation runs are shown in Table 6.8 and Figure 6.21.

Table 6.8: Simulation results for 3 KPI (cost, death rate and heart failure relapse) for different values of implantation and replacement probabilities. Each simulation was done with 40,000 patients.

Scenario	Implantation Probability	Replacement probability	Total cost (million €)	Death rate (%)	Heart failure relapse
1	0,30	0,05	335,2	33,41%	23 890
2	0,40	0,05	365,3	33,41%	23 888
3	0,50	0,05	374,6	33,41%	23 888
4	0,10	0,1	412,7	33,40%	23 886
5	0,20	0,1	356,3	33,40%	23 876
6	0,40	0,1	367,2	33,40%	23 870
7	0,50	0,1	383,5	33,41%	23 888
8	0,10	0,15	346,0	33,40%	23 880
9	0,20	0,15	394,8	33,39%	23 879
10	0,30	0,15	385,4	33,39%	23 876
11	0,40	0,15	406,1	33,39%	23 865
12	0,50	0,15	435,3	33,38%	23 862
13	0,10	0,2	387,2	33,39%	23 862
14	0,15	0,2	365,8	33,38%	23 886
15	0,20	0,2	307,1	33,38%	23 898
16	0,30	0,2	356,3	33,39%	23 884
17	0,35	0,2	409,7	33,40%	23 837
18	0,40	0,2	429,2	33,41%	23 828
19	0,45	0,2	438,5	33,40%	23 842
20	0,50	0,2	398,3	33,40%	23 860

Numerical results validate our modeling approach and the balance mechanism between costs and care quality. When the implantation rate increases, the total cost follows because of the device's cost. It also slightly decreases the death rate (significant difference only between extreme scenarios). No significant reduction of heart failure relapse was observed. The decreasing trend in death rate is slow compared to the increase in cost. It shows that the current model reaches its limits and is not rich and complex enough to

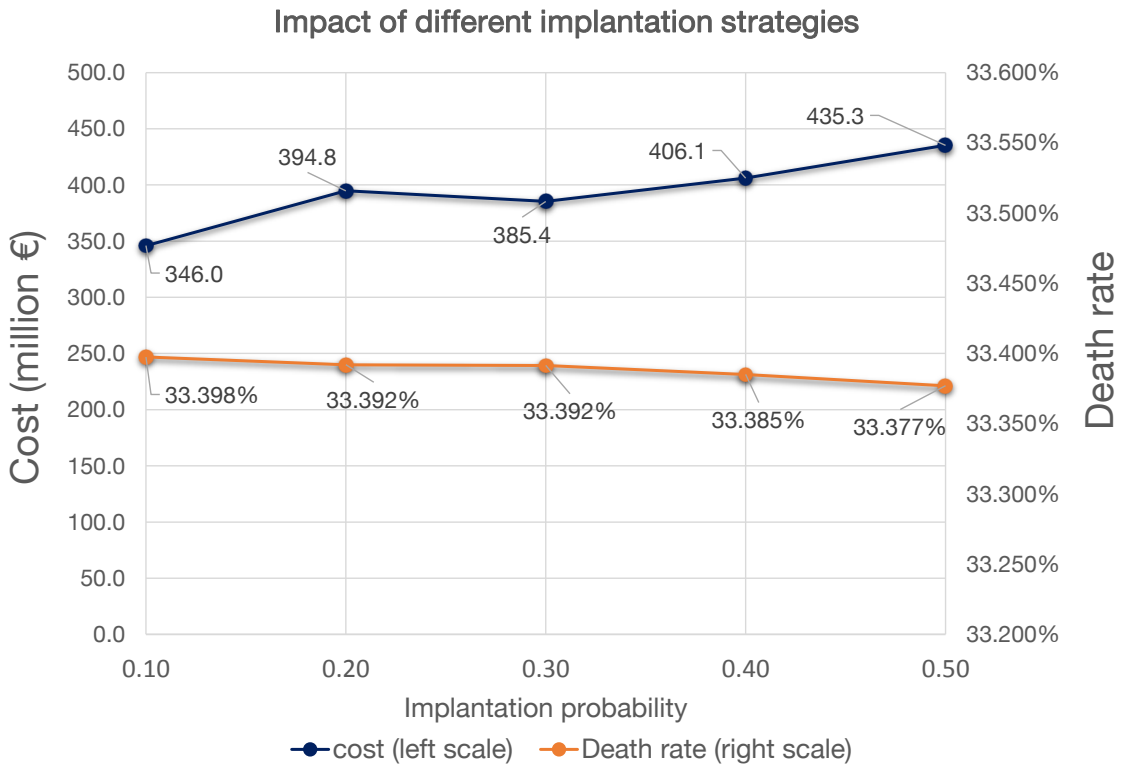


Figure 6.21: Simulation results: measure of 2 KPI (cost and death rate) for different values of implantation probability. Each simulation was done with 40,000 patients.

capture all the mechanisms at work. These results are preliminaries for deeper experimentation. Our goal here was to validate the concept of an innovative way of converting Process Mining results into a simulation model. This objective was reached and this work is a good starting point for further investigations.

6.6 Conclusion

This chapter presented a comprehensive case study to illustrate the practical use of the approaches introduced in this thesis. The French national database of the hospital claims from 2006 to 2015 is used as an event log. The case of patients suffering from cardiac arrhythmia and who need the implantation of cardioverter defibrillators is addressed. Numerous results were provided and show the benefit of our approach (knowledge discovery, process discovery, car sequence comparison, decision tree rules, time perspective, model validation, sensitivity analysis and scenarios evaluation).

The modeling methodology can be re-used on new case studies (other medical areas) or on new data having the same structure. This is the most important strength of the approach. Models are not handmade at each step, they are automatically derived from well prepared data. Regarding the simulation of new clinical pathway scenarios, our approach can be seen as a proof of concept that could be extended to simulate larger models and more complex scenarios, whatever the size of the data.

Conclusion

Summary

Industrial engineering, among other scientific disciplines, promotes the adaptation of existing methods, or the development of new ones, to help improve the performance of health-care systems. In health-care organizations, a major trend for the improvement of care quality while reducing costs is the design and implementation of clinical pathways. Clinical pathway modeling is a popular topic which has been addressed in various ways, depending on the techniques and the description level. However, due to the inherent variability of care processes, to the stochasticity of patient management and the lack of evidence-based decisions, several challenges remain to propose sufficiently flexible and realistic models.

In this thesis, we proposed a complete and innovative methodology to automatically discover, analyze and simulate clinical pathways using health data. This methodology splits up in several steps, each of whom is dedicated to a scientific challenge. It makes the contributions of this work multi-fold.

First, a new approach to compute optimal process models from complex event logs is proposed. It includes the mathematical definition of new replayability functions, the first attempt to propose a quantitative criterion to evaluate mined process models. A solution method based on a tabu search is also proposed. Then, a health-care analytic toolbox with 3 *instruments* is introduced: a method to quantitatively compare two sequences of medical events, a predictive model of the next step in a clinical pathway and a complete methodology to automatically integrate the two previous points in the context of clinical pathway modeling. A formal procedure for the automatic conversion of a static process model, in the form of a causal net, into a simulation model was presented. A new class of state chart, the Clinical Pathway State Chart, enables the integration of many concepts related to care processes. The resulting simulation model is used to perform sensitivity analysis and scenarios evaluations on a personalized and automatic basis. The entire methodology has been formalized in a generic framework, so that it can be implemented and used in an automatic manner.

Our modeling methodology was applied on a several millions-hospital events database, the French national database of all hospital activities from 2006 to 2015 (11 million patients per year). A comprehensive case study on cardiovascular diseases was made to illustrate the practical use of the approaches and their benefit for medical decision aid.

Future works

As a final point, we would like to gather our scattered discussions about some of the possible extensions of the present work that we could identified. Through this thesis, we came through a variety of scientific challenges. We addressed each challenge with a dedicated technique, or a combination of dedicated techniques. Some of our choices could have been different, and some of our models can be improved.

For future works regarding the process discovery problem, one could improve process model computation by taking into account domain specific parameters and by adding weights on clusters, classes and/or arcs. The use of experts' knowledge could be integrated in an ontology map, so that a discovery algorithm can use it to converge quickly to realistic process model. A more in-depth study of the relationship between a process model quality measures and the information contained in the data is also needed.

Our analytic toolbox could also benefit from some improvements, especially in the choice of a machine learning algorithm to solve the classification problems. We only proposed two supervised learning algorithms (decision trees and random forest), which is insufficient to guarantee the best possible predictions. Based on the classification problem that we defined, a performance benchmark of machine learning algorithms on health data sets would benefit to our methodology. The emergent field of discriminant analysis using mixed-integer programming also seems a promising way to perform classifications tasks (Lee et al., 2012, 2016).

A major extension of our models relies in the addition of resources and medical decision modeling. It would open a large set of possibilities. The individual modeling of each hospital stay as a small process itself would bring a new perspective to the entire approach. In our current model, care-states modify patients' features in a deterministic way. An interesting extension would be to add a resource perspective and to model medical decisions into the clinical pathway state chart. Our definition of a clinical pathway state chart enables the integration resources and medical decisions without having to redefine every concept. This new integration of resources implies a dynamic management of their seizing and release by the patients. It means that patients cannot be considered independently anymore, they interact with each other through resource-sharing mechanisms. The Monte-Carlo simulation would not apply anymore. Thus, we could extend the modeling of care-states to the next level by proposing that each care-state is a dedicated discrete-event simulation model.

We strongly believe that the work of this thesis can be used as the ground foundations to create a bridge between traditional modeling-and-simulation of hospital services (such as Discrete Event Simulation), more original modeling methods (Multi-Agent Systems) and process mining techniques. The combination of the 3 approaches would result in a single, and probably rather complex, but extremely complete model of long-term clinical pathways. It would integrate at the same time the objectivity of data-driven process discovery at a national scale, and the precision of detailed hospital service models and optimized organizations. In addition, it would also capture complex interactions between patients, care providers and disease natural evolution.

Appendix A

Overview of the 25 most used machine learning in practice[1]

Algorithm	Accuracy	Training time	Linearity	Parameters
Two-class classification				
logistic regression		●	●	5
decision forest	●	○		6
decision jungle	●	○		6
boosted decision tree	●	○		6
neural network	●			9
averaged perceptron	○	○	●	4
support vector machine		○	●	5
locally deep support vector machine	○			8
Bayes' point machine		○	●	3
Multi-class classification				
logistic regression		●	●	5
decision forest	●	○		6
decision jungle	●	○		6
neural network	●			9
one-v-all	-	-	-	-
Regression				
linear		●	●	4
Bayesian linear		○	●	2
decision forest	●	○		6
boosted decision tree	●	○		5
fast forest quantile	●	○		9
neural network	●			9
Poisson			●	5
ordinal				0
Anomaly detection				
support vector machine	○	○		2
PCA-based anomaly detection		○	●	3
K-means		○	●	4

Algorithm properties:
● - shows excellent accuracy, fast training times, and the use of linearity
○ - shows good accuracy and moderate training times

[1]From Microsoft Azure Machine Learning https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-algorithm-choice

Appendix B

Student t-distribution table

Table B.1: Student's t-distribution for k degrees of freedom and quantiles of order $1 - \alpha$

α	25 %	20 %	15 %	10 %	5 %	2,5 %	1 %	0,5 %	0,25 %	0,1 %	0,05 %
$1 - \alpha$	75 %	80 %	85 %	90 %	95 %	97,5 %	99 %	99,5 %	99,75 %	99,9 %	99,95 %
k											
1	1	1,376	1,963	3,078	6,314	12,71	31,82	63,66	127,3	318,3	636,6
2	0,816	1,061	1,386	1,886	2,92	4,303	6,965	9,925	14,09	22,33	31,6
3	0,765	0,978	1,25	1,638	2,353	3,182	4,541	5,841	7,453	10,21	12,92
4	0,741	0,941	1,19	1,533	2,132	2,776	3,747	4,604	5,598	7,173	8,61
5	0,727	0,92	1,156	1,476	2,015	2,571	3,365	4,032	4,773	5,893	6,869
6	0,718	0,906	1,134	1,44	1,943	2,447	3,143	3,707	4,317	5,208	5,959
7	0,711	0,896	1,119	1,415	1,895	2,365	2,998	3,499	4,029	4,785	5,408
8	0,706	0,889	1,108	1,397	1,86	2,306	2,896	3,355	3,833	4,501	5,041
9	0,703	0,883	1,1	1,383	1,833	2,262	2,821	3,25	3,69	4,297	4,781
10	0,7	0,879	1,093	1,372	1,812	2,228	2,764	3,169	3,581	4,144	4,587
11	0,697	0,876	1,088	1,363	1,796	2,201	2,718	3,106	3,497	4,025	4,437
12	0,695	0,873	1,083	1,356	1,782	2,179	2,681	3,055	3,428	3,93	4,318
13	0,694	0,87	1,079	1,35	1,771	2,16	2,65	3,012	3,372	3,852	4,221
14	0,692	0,868	1,076	1,345	1,761	2,145	2,624	2,977	3,326	3,787	4,14
15	0,691	0,866	1,074	1,341	1,753	2,131	2,602	2,947	3,286	3,733	4,073
16	0,69	0,865	1,071	1,337	1,746	2,12	2,583	2,921	3,252	3,686	4,015
17	0,689	0,863	1,069	1,333	1,74	2,11	2,567	2,898	3,222	3,646	3,965
18	0,688	0,862	1,067	1,33	1,734	2,101	2,552	2,878	3,197	3,61	3,922
19	0,688	0,861	1,066	1,328	1,729	2,093	2,539	2,861	3,174	3,579	3,883
20	0,687	0,86	1,064	1,325	1,725	2,086	2,528	2,845	3,153	3,552	3,85
21	0,686	0,859	1,063	1,323	1,721	2,08	2,518	2,831	3,135	3,527	3,819
22	0,686	0,858	1,061	1,321	1,717	2,074	2,508	2,819	3,119	3,505	3,792
23	0,685	0,858	1,06	1,319	1,714	2,069	2,5	2,807	3,104	3,485	3,767
24	0,685	0,857	1,059	1,318	1,711	2,064	2,492	2,797	3,091	3,467	3,745
25	0,684	0,856	1,058	1,316	1,708	2,06	2,485	2,787	3,078	3,45	3,725
26	0,684	0,856	1,058	1,315	1,706	2,056	2,479	2,779	3,067	3,435	3,707
27	0,684	0,855	1,057	1,314	1,703	2,052	2,473	2,771	3,057	3,421	3,69
28	0,683	0,855	1,056	1,313	1,701	2,048	2,467	2,763	3,047	3,408	3,674
29	0,683	0,854	1,055	1,311	1,699	2,045	2,462	2,756	3,038	3,396	3,659
30	0,683	0,854	1,055	1,31	1,697	2,042	2,457	2,75	3,03	3,385	3,646
40	0,681	0,851	1,05	1,303	1,684	2,021	2,423	2,704	2,971	3,307	3,551
50	0,679	0,849	1,047	1,299	1,676	2,009	2,403	2,678	2,937	3,261	3,496
60	0,679	0,848	1,045	1,296	1,671	2	2,39	2,66	2,915	3,232	3,46
80	0,678	0,846	1,043	1,292	1,664	1,99	2,374	2,639	2,887	3,195	3,416
100	0,677	0,845	1,042	1,29	1,66	1,984	2,364	2,626	2,871	3,174	3,39
120	0,677	0,845	1,041	1,289	1,658	1,98	2,358	2,617	2,86	3,16	3,373
∞	0,674	0,842	1,036	1,282	1,645	1,96	2,326	2,576	2,807	3,09	3,291

Appendix C

List of 15 validation techniques for simulation models (Sargent, 2011)

1. *Animation:* The model's operational behavior is displayed graphically as the model moves through time. For instance, the movements of parts through a factory during a simulation run are shown graphically.
2. *Comparison to other models:* Various results of the simulation model being validated are compared to results of other (valid) models. For example, simple cases of a simulation model are compared to known results of analytic models.
3. *Degenerate Tests:* The degeneracy of the model's behavior is tested by appropriate selection of values of the input and internal parameters. For example, does the average number in the queue of a single server continue to increase over time when the arrival rate is larger than the service rate?
4. *Event Validity*: The events of occurrences of the simulation model are compared to those of the real system to determine if they are similar. For example, compare the number of fires in a fire department simulation to the actual number of fires.
5. *Extreme Condition Tests*: The model structure and outputs should be plausible for any extreme and unlikely combination of levels of factors in the system. For example, if in-process inventories are zero, production output should usually be zero.
6. *Face Validity*: Individuals knowledgeable about the system are asked whether the model and its behavior are reasonable.
7. *Historical Data Validation*: If historical data exist, part of the data is used to build the model and the remaining data are used to test whether the model behaves as the system does.
8. *Historical Methods*: The three historical methods of validation are rationalism, empiricism, and positive economics. Rationalism requires that the assumptions underlying a model be clearly stated and that they are readily accepted. Logic deductions are used from these assumptions to develop the correct (valid) model. Empiricism requires every assumption and outcome to be empirically validated. Positive economics requires only that the model's outcome(s) be correct and is not concerned with a model's assumptions or structure (causal relationships or mechanisms).
9. *Internal Validity*: Several replications of a stochastic model are made to determine the amount of (internal) stochastic variability in the model. A large amount of variability may cause the model's

results to be questionable.

10. *Multistage Validation*: Naylor and Finger (1967) proposed combining the three historical methods of rationalism, empiricism, and positive economics into a multistage process of validation. It consists of (1) developing the model's assumptions on theory, observations, and general knowledge, (2) validating the model's assumptions where possible by empirically testing them, and (3) comparing the input-output relationships of the model to the real system.
11. *Operational Graphics*: Values of various performance measures are shown graphically as the model runs through time. The dynamical behaviors of performance indicators are visually displayed as the simulation model runs through time to ensure they behave correctly.
12. *Parameter Variability - Sensitivity Analysis*: This technique consists of changing the values of the input and internal parameters of a model to determine the effect upon the model's behavior or output. The same relationships should occur in the model as in the real system.
13. *Predictive Validation*: The model is used to predict the system's behavior. Then, comparisons are made between the system's behavior and the model's forecast to determine if they are the same. The system data may come from an operational system or be obtained by conducting experiments on the system.
14. *Traces*: The behaviors of different types of specific entities in the model are traced (followed) through the model to determine if the model's logic is correct and if the necessary accuracy is obtained.
15. *Turing Tests*: Individuals who are knowledgeable about the operations of the system being modeled are asked if they can discriminate between system and model outputs.

Appendix D

The relevant PMSI fields for the study of clinical pathways

Table D.1: List of the 28 most useful fields of the PMSI database.

Field name	**Brief description**
Administrative fields	
Facility ID	Name, location and status of the facility
Hospital stay ID	Unique identifier of the stay in the database
Version of the DRG classification	A new version of the DRG codes is edited each year
Tariff of the stay	Cost of the stay from a health insurance perspective
Length of stay	Duration (in days)
Year (at discharge)	2013, 2014, 2015, ...
Month (at discharge)	From 1 to 12
Number of RUM	Number of aggregated RUM producing the stay summary
Patient fields	
Patient ID	Unique identifier of a patient
Age of the patient (year)	Age at entrance (in days if younger than 1 year)
Gender	Male / Female
Elapse time since the last stay	Duration in days
Entry mode	From home / internal transfer / external transfer
Exit mode	To home / internal transfer / external transfer / death
Home ZIP code	Location area of the patient's home
Medical fields	
Main diagnosis	Medical reason of the hospitalization (code from the ICD-10^{th})[1]
Secondary diagnosis	Associated reason of the hospitalization (code from the ICD-10^{th})[1]
List of other related diagnoses	Useful for multi-pathology patients
List of medical procedures	Procedures are coded according to the French CCAM[2]
List of expensive drugs	The list only includes drugs that induce extra charges
Diagnose Related Group (GHM)	Aggregation of all medical and financial information in one code
Number of chemo/radio-therapy sessions	Several sessions can be invoiced in a single stay
Use of dialysis	Number of medical procedures, induces extra charge
Use of hemodialysis	Number of medical procedures, induces extra charge
Use of hyperbaric chamber	Number of medical procedures, induces extra charge
Need for reanimation	Number of days in reanimation, induces extra charge
Need for intensive care	Number of days in intensive care, induces extra charge
Pregnancy duration	If relevant, duration in weeks

Appendix E

Application of machine learning to find cost profiles for HIV patients

In Chapter 4, we presented a **health-care analytics toolbox** with 3 components (*comparison of two sequences*, *predictive models* and an *automated analysis process*). Here, we present the results of specifically using the approach described in the predictive models on a case study. The focus is not on the prediction of the next step of a clinical pathway, but on the classification of patient profiles based on their features. The therapeutic area is the **Human Immunodeficiency Virus (HIV)** infection, which causes the Acquired ImmunoDeficiency Syndrome (AIDS). The entire study is illustrated in Appendix E.

The objective was to assess the capability of a classic data mining technique to be applied on a health-care database in order to better understand drivers of health-care expenditures and the management of diseases.

Method. We selected hospital stays with an HIV code, HIV being the principal cause of hospitalization or not (codes B20*, B21*, B22*, B23*, B24*). Patients hospitalized with an HIV code in 2013 were extracted and followed up for one year (365 days). 10 groups of comorbidities and 5 types of opportunistic infections (OI) linked to HIV were also identified, and their presence was tracked among these patients. Data were analyzed with a Decision Tree algorithm (CART algorithm with smart pruning option), in order to explain **HIV hospitalization costs** depending on non-linear combinations of age, gender and the presence of comorbidities or OIs.

Results. 30,294 patients with 70,180 hospital stays were included, for a total cost of 180 million euros. The Decision Tree algorithm could determine 165 different patient profiles, created automatically to maximize the gathering of patients with similar features. The most discriminating variables for the cost of hospitalization were infections not associated to HIV, bacterial OI, cancer, fungal infections and endocrino-metabolic complications, whereas age, psychiatric and hepatic comorbidities were not discriminating. The average annual cost of patient profiles ranged from 1,680€ to 42,650€. These results are shown in Figure E.1. We used a sunburst graph to visualize how the entire cohort (30,294) is iteratively split in two subgroups at each layer (from the inner circle to the outside circle). Each split is based on the binary response to a specific question (= a patient's feature). In the end, each strip of the sunburst represents a patient profile. The bottom of Figure E.1 presents 4 stringent profiles, from a

frequent and low cost profile (profile 2) to a rare and high cost profile (profile 4).

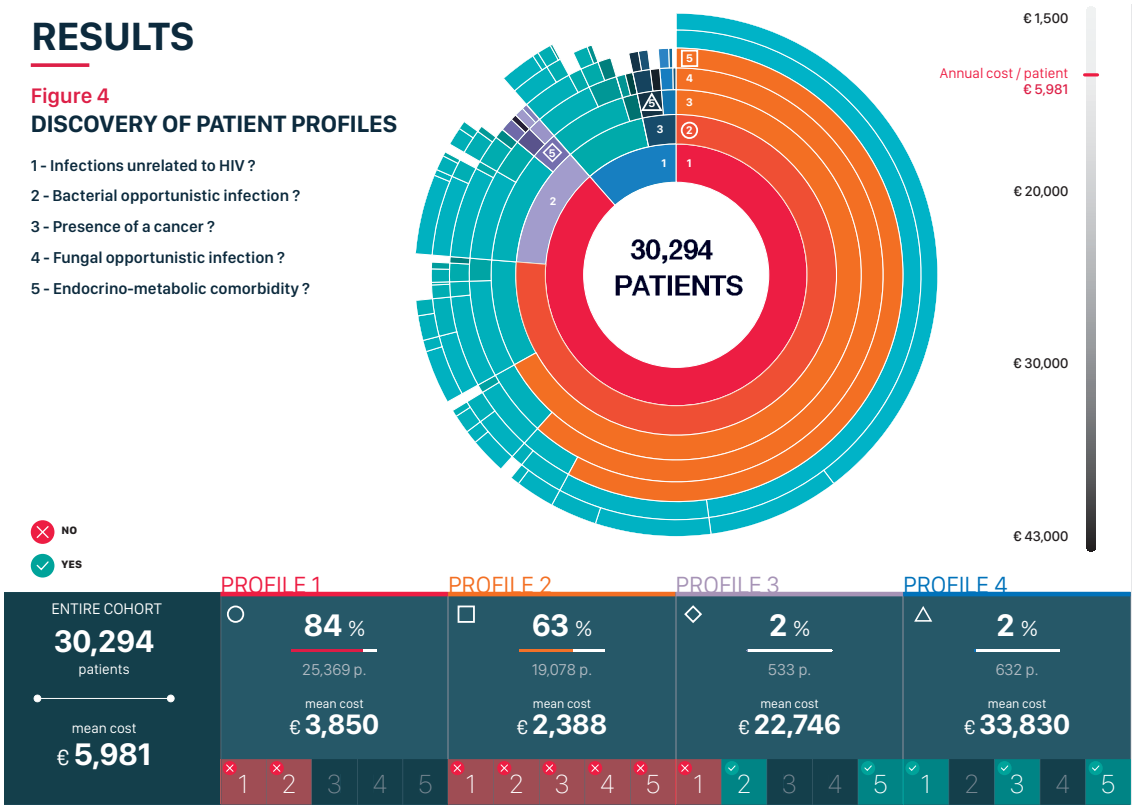


Figure E.1: Cost profiles for patients with HIV

Conclusion. This exploratory study shows that traditional data mining techniques, such as Decision Tree Algorithms, are relevant to identify patient profiles from big databases and may have predictive capabilities. It could help identifying leverages to prevent hospitalizations costs. Further research should be done, adding therapeutic and biological parameters.

In this section, we have introduced a health-care database that can be used for several purposes. The next section presents a comprehensive medical case study on which we propose to apply the modeling methodology presented in this thesis.

Martin PRODEL[1,3], Camille AMAZ[1], Alexandre VAINCHTOCK[1], Laurent FINKIELSZTEJN[2], Céline AUBIN[2]
[1]HEVA, Lyon, France ; [2]ViiV Healthcare, Marly-le-Roi, France ; [3]Mines, Saint-Étienne, France

ISPOR 19th Annual European Congress
October 29 - November 2, 2016
Vienna, Austria

PRM71

ADVANCED DATA MINING APPROACH

Prediction of costs associated with the hospital management of HIV patients in France

INTRODUCTION

Data mining is not new and has been efficiently employed in others domains *(Bank, Cybernetics, Marketing, Energy, etc.)*[1]. The amount of data collected in **medical information systems** is tremendous, but much data remains unused because of their complexity[2]. This still increasing volume of data requires new analytical approaches that are efficient, sensitive and better than classical statistics to handle Big Data[1].

Figure 1
METHODOLOGICAL DIVERGENCES BETWEEN DATA MINING AND STATISTICS

DATA MINING	CRITERIA	STATISTICS
Explanatory approach, discover the unknown with no preconception	PARADIGM	(Un)-validate predefined hypothesis, risk of experimenter bias
Unrivalled quality of predictions and explanations, using cross-validations. Use extrapolation for missing data	MAIN FEATURE	Precise measurement of uncertainty, comparison of 2 populations, confidence assessment of measures
High adaptability and reusability in other domains	SECOND FEATURE	Mathematical definition of statistical tests ensures significant results
Decision Tree, Neural Network, Bayesian, Genetic Algorithm, Support Vector Machine	EXAMPLES OF TECHNIQUES	Principal Component Analysis, p-value testing, multivariate & univariate regression
Capable of dealing with millions of data ("Big data")	VOLUME	Suitable for middle size datasets (< 1 million observations)
Data mining selects and tells you the non-linear combinations of many variables that best explain the value of the target variable.	TYPICAL RESULTS	Logistic regression tells you that X% of the cost is explained by having this or this comorbidity, PCA tells you if variables are redundant.

OBJECTIVE

The main objective of this study is to assess the capability of a state-of-the-art data mining technique to be applied on a healthcare database in order to **better understand drivers of healthcare expenditure and the management of diseases.**
We utilized the approach in order to explain the patient features that are the main drivers of cost associated with HIV patients' hospital management. A secondary objective is to assess how reproducible such an approach is on other medical databases with potentially different patient features.

METHODS

KEY FIGURES

Study period: January 2013 - December 2014
Number of HIV patients: 30,294
Number of stays: 70,180
Total annual cost: 180 million €
Mean annual cost per patient : 5,981 € (+/- 10,661)
Nb of patient profiles found using Data Mining: 165

DATA EXTRACTION

The PMSI-MSO *(French Medical Information System - Medicine, Surgery, Obstetric units)* database was used to extract all hospital stays in 2013 with at least one of the following HIV ICD-10 *(International Classification of Diseases, 10th revision)* codes as principal diagnosis, related diagnosis or significantly associated diagnosis: **B20*, B21*, B22*, B23*, B24*.**

70,180 stays

M€ **180** total annual cost

1-YEAR FOLLOW-UP OF PATIENTS & CLASSIFICATIONS OF STAYS

For each patient hospitalized once in 2013 with an HIV code, a 12 months follow-up was performed to capture any further hospital stay. The same algorithm and medical review as for the inclusion stay were conducted on these additional stays, in order to exclude stays not related to HIV. All stays were classified by HIV clinician experts into **10 groups of comorbidities** and **5 groups of opportunistic infections (OI)** using ICD-10 codes *(figure 2)*.

HIV HOSPITALIZATION COST

The cost of stays were evaluated from a **NHS perspective**. The PMSI database provides many details about stays' cost as it was specifically designed for economic evaluation of hospital activities.

OUR DATA MINING ALGORITHM

Data were analyzed with a supervised-learning data mining technique, an Enhanced Decision Tree algorithm based on Breiman' CART model[3] *(figure 3)*.

Figure 2
INPUT VARIABLES FOR THE ANALYSIS OF PATIENT PROFILES WITH DATA MINING

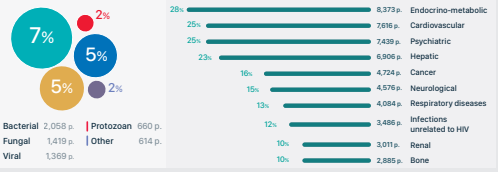
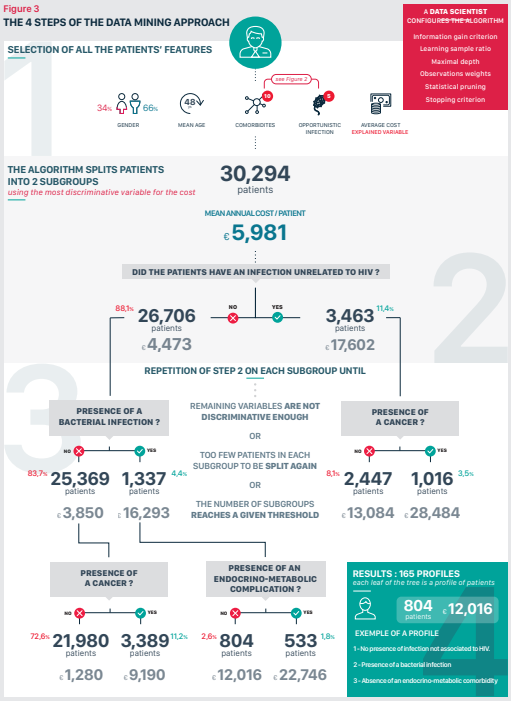
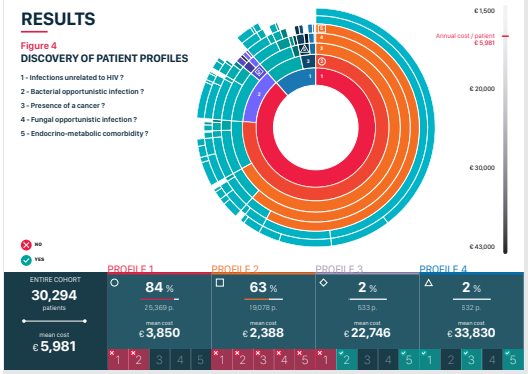


Figure 3
THE 4 STEPS OF THE DATA MINING APPROACH



RESULTS

Figure 4
DISCOVERY OF PATIENT PROFILES



CONCLUSION

The present study is a pilot that successfully demonstrates how a data mining technique, from the field of Artificial Intelligence, can help us better understand Hospital costs for HIV patients: our data mining algorithm identified **specific patient profiles which explain the differentiating cost drivers in HIV inpatient care**. Classical statistical approaches would struggle to provide such detailed profiles with numerous combinations of variables.

1 - This approach could work with other data sources, especially with more clinical and laboratory data *(e.g.: viral load)*.
It can also explain drivers for any available target variables *(cost, death, patients virologically suppressed or patients dropping out of care)*.

2 - Such collaborative efforts between health care professionals and engineers can lead to enhanced uses of health data that can provide new answers and thus improve disease management
Example: Preventing a bacterial infection in newly diagnosed HIV patients with Endocrino-Metabolic comorbidities, is much more cost reducing than preventing a fungal infection.

3 - Based on results of Data Mining analyses, clinicians will know the medical interventions to focus on and prioritize.
Example: Results show that newly diagnosed HIV patients are more likely to get bacterial infections if they already have neurological and renal comorbidities (=predictive capabilities).

REFERENCES : [1]Adem Karahoca, Dilek Karahoca and Mert Şanver (2012). Survey of Data Mining and Applications (Review from 1996 to Now), Data Mining Applications in Engineering and Medicine, Associate Prof. Adem Karahoca (Ed.), InTech, DOI: 10.5772/48803.
[2]Herland, Matthew, Taghi M Khoshgoftaar, et Randall Wald. «A review of data mining using big data in health informatics.» Journal Of Big Data, 2014: 1–35.
[3]Breiman, L., Friedman, J.H., Olshen, R.A., and Stone, C.J. (1984) "Classification and Regression Trees", Wadsworth, Belmont, CA. Republished by CRC Press

Bibliography

Armand Abergel, Michel Rotily, Sébastien Branchoux, Raoudha Akremi, Lucie de Léotoing, Alexandre Vainchtock, and Anne-Franoise Gaudin. Chronic hepatitis c: Burden of disease and cost associated with hospitalisations in france in 2012 (the hepc-lone study). *Clinics and Research in Hepatology and Gastroenterology*, 40(3):340 – 348, 2016. ISSN 2210-7401.

Shola Adeyemi, Eren Demir, and Thierry Chaussalet. The analyses of individual patient pathways : Investigating regional variation in copd readmissions. In *Applied Stochastic Models and Data Analysis (ASMDA 2009 proceedings)*, volume 6, pages 316–319, July 2009.

Shola Adeyemi, Eren Demir, and Thierry Chaussalet. Towards an evidence-based decision making healthcare system management: Modelling patient pathways to improve clinical outcomes. *Decision Support Systems*, 55(1):117 – 125, 2013. ISSN 0167-9236.

Reza Akhavian and Amir H. Behzadan. Automated knowledge discovery and data-driven simulation model generation of construction operations. In *2013 Winter Simulations Conference (WSC)*, pages 3030–3041, Dec 2013.

François-André Allaert, Eric Benzenine, and Catherine Quantin. Hospital incidence and annual rates of hospitalization for venous thromboembolic disease in france and the usa. *Phlebology*, page 0268355516653005, Oct 2016. ISSN 0268-3555.

Ahmet K. Arslan, Cemil Colak, and Mehmet E. Sarihan. Different medical data mining approaches based prediction of ischemic stroke. *Comput. Methods Prog. Biomed.*, 130(C):87–92, July 2016. ISSN 0169-2607.

Vincent Augusto and Xiaolan Xie. Modélisation et analyse de flux par la simulation en milieu hospitalier : état de l'art. *Proceedings of the GISEH conference*, 2006.

Vincent Augusto and Xiaolan Xie. Redesigning pharmacy delivery processes of a health care complex. *Health Care Management Science*, 12(2):166–178, 2009a. ISSN 1572-9389.

Vincent Augusto and Xiaolan Xie. Redesigning the neurovascular unit of a health care complex using simulation. In *Proceedings of the 2009 IEEE Conference on Industrial Engineering and System Management*, 2009b.

Vincent Augusto and Xiaolan Xie. A modeling and simulation framework for health care systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 44(1):30–46, Jan 2014. ISSN 2168-2216.

Vincent Augusto, Olfa Rejeb, Xiaolan Xie, Saber Aloui, Lionel Perrier, Pierre Biron, and Thierry Durand. Performance evaluation of health information systems using aris modeling and discrete-event simulation. In *2015 Winter Simulation Conference (WSC)*, pages 1503–1514, Dec 2015.

L. Benjamin, F.-E. Cotté, F. Mercier, A. Vainchtock, G. Vidal-Trécan, and I. Durand-Zaleski. Burden of breast cancer with brain metastasis: a french national hospital database analysis. *Journal of Medical Economics*, 15(3):493–499, 2012.

Tobias Blum, Nicolas Padoy, Hubertus Feußner, and Nassir Navab. Workflow mining for visualization and analysis of surgeries. *International Journal of Computer Assisted Radiology and Surgery*, 3(5):379–386, 2008. ISSN 1861-6429.

R.P. Jagadeesh Chandra Bose and Wil M.P. van der Aalst. *Abstractions in Process Mining: A Taxonomy of Patterns*, pages 159–175. Springer Berlin Heidelberg, 2009. ISBN 978-3-642-03848-8.

R.P. Jagadeesh Chandra Bose, Eric H.M.W. Verbeek, and Wil M.P. van der Aalst. *Discovering Hierarchical Process Models Using ProM*, pages 33–48. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-29749-6.

R.P. Jagadeesh Chandra Bose, Ronny S. Mans, and Wil M.P. van der Aalst. Wanna improve process mining results? In *2013 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, pages 127–134, April 2013.

Loubna Bouarfa and Jenny Dankelman. Workflow mining and outlier detection from clinical activity logs. *Journal of Biomedical Informatics*, 45(6):1185–1190, 2012. ISSN 1532-0464.

Loubna Bouarfa, Pieter P. Jonker, and Jenny Dankelman. Discovery of high-level tasks in the operating room. *Journal of Biomedical Informatics*, 44(3):455 – 462, 2011. ISSN 1532-0464.

Wulfran Bougouin, Lionel Lamhaut, Eloi Marijon, Daniel Jost, Florence Dumas, Nicolas Deye, Frankie Beganton, Jean-Philippe Empana, Emilie Chazelle, Alain Cariou, and Xavier Jouven. Characteristics and prognosis of sudden cardiac death in greater paris. *Intensive Care Medicine*, 40(6):846–854, 2014. ISSN 1432-1238.

Adrain W. Bowman and Adelchi Azzalini. *Applied smoothing techniques for data analysis : the kernel approach with S-Plus illustrations*. Oxford : Clarendon Press ; New York : Oxford University Press, 1997. ISBN 0198523963.

Richard Braun, Martin Burwitz, Hannes Schlieter, and Martin Benedict. Clinical processes from various angles - amplifying bpmn for integrated hospital management. In *2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 837–845, Nov 2015.

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. ISSN 1573-0565.

Michel Camilleri and Filippo Neri. Parameter optimization in decision tree learning by using simple genetic algorithms. *Wseas Transactions on Computers*, 13:582–591, 2014. ISSN 2224-2678.

Michel Camilleri, Filippo Neri, and Michail Papoutsidakis. An algorithmic approach to parameter selection in machine learning using meta-optimization techniques. *WSEAS Transactions on Systems*, 13:202–213, 2014. ISSN 2224-2678.

Emilio Carrizosa and Dolores Romero Morales. Supervised classification and mathematical optimization. *Comput. Oper. Res.*, 40(1):150–165, January 2013. ISSN 0305-0548.

Yolanda Carson and Anu Maria. Simulation optimization: Methods and applications. In *Proceedings of the 29th Conference on Winter Simulation*, WSC '97, pages 118–126, Washington, DC, USA, 1997. IEEE Computer Society. ISBN 0-7803-4278-X.

Rich Caruana and Alexandru Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 161–168, New York, NY, USA, 2006. ACM. ISBN 1-59593-383-2.

Rich Caruana, Nikos Karampatziakis, and Ainur Yessenalina. An empirical evaluation of supervised learning in high dimensions. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 96–103, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-205-4.

M. Emre Celebi, Y. Alp Aslandogan, and Paul R. Bergstresser. Mining biomedical images with density-based clustering. In *Proceedings of the International Conference on Information Technology - Volume 01*, pages 163–168. IEEE Computer Society, 2005. ISBN 0-7695-2315-3.

Martha Centeno, Marsha A. Lee, Elizabeth Lopez, Helida R. Fernandez, Manuel Carrillo, and Tom Ogazon. A simulation study of the labor and delivery rooms at jmh. In *Proceeding of the 2001 Winter Simulation Conference*, volume 2, pages 1392–1400, december 2001.

Christophe Chaignot, Alain Weill, Philippe Ricordeau, and Francois Alla. Utilisation en france du baclofène dans l'alcoolodépendance de 2007 à 2013 : étude à partir du {SNIIRAM} et du {PMSI}. *Thérapie*, 70(5):443 – 453, 2015. ISSN 0040-5957.

Pete Chapman, Julian Clinton, Randy Kerber, Thomas Khabaza, Thomas Reinartz, Colin Shearer, and Rudiger Wirth. Crisp-dm 1.0 step-by-step data mining guide. Technical report, The CRISP-DM consortium, August 2000.

Moez Charfeddine and Benoit Montreuil. Integrated agent-oriented modeling and simulation of population and healthcare delivery network: Application to copd chronic disease in a canadian region. In *Proceedings of the 2010 Winter Simulation Conference*, pages 2327–2339, Dec 2010. doi: 10.1109/WSC.2010.5678930.

Courtney D. Corley, Diane J. Cook, Armin R. Mikler, and Karan P. Singh. Text and structural data mining of influenza mentions in web and social media. *Int J Environ Res Public Health*, 7(2):596–615, Feb 2010. ISSN 1661-7827.

David Corne, Clarisse Dhaenens, and Laetitia Jourdan. Synergies between operations research and data mining: The emerging use of multi-objective approaches. *European Journal of Operational Research*, 221(3):469 – 479, 2012. ISSN 0377-2217.

Adriana M. Coroiu. Tuning model parameters through a genetic algorithm approach. In *2016 IEEE 12th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 135–140, Sept 2016.

Murray J. Cote and William E. Stein. A stochastic model for a visit to the doctors office. *Mathematical and Computer Modelling*, 45(34):309 – 323, 2007. ISSN 0895-7177.

Aron Culotta. Towards detecting influenza epidemics by analyzing twitter messages. In *Proceedings of the First Workshop on Social Media Analytics*, SOMA '10, pages 115–122. ACM, 2010. ISBN 978-1-4503-0217-3.

Arianna Dagliati, Lucia Sacchi, Carlo Cerra, Paola Leporati, Pasquale De Cata, Luca Chiovato, John H. Holmes, and Riccardo Bellazzi. Temporal data mining and process mining techniques to identify cardiovascular risk-associated clinical pathways in type 2 diabetes patients. In *IEEE-EMBS International Conference on Biomedical and Health Informatics*, pages 240–243, June 2014.

Frederica Darema. *Dynamic Data Driven Applications Systems: A New Paradigm for Application Simulations and Measurements*, pages 662–669. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004. ISBN 978-3-540-24688-6.

Amit K. Das, Aman Kedia, Lisha Sinha, Saptarsi Goswami, Tamal Chakrabarti, and Amlan Chakrabarti. Data mining techniques in indian healthcare: A short review. In *2015 International Conference on Man and Machine Interfacing (MAMI)*, pages 1–7, Dec 2015.

Massimiliano de Leoni, Wil M.P. van der Aalst, and Marcus Dees. A general process mining framework for correlating, predicting and clustering dynamic behavior based on event logs. *Information Systems*, 56:235 – 257, 2016. ISSN 0306-4379.

L. de Léotoing, J. Fernandes, C. Tournier, B. Jouaneton, and A. Vainchtock. An assessment of annual costs of patients hospitalized for spinal tumors in france: Analysis using the pmsi database. *Value in Health*, 18(7):A443, November 2015. ISSN 1098-3015.

Craig C. Douglas and Yalchin R. Efendiev. A dynamic data-driven application simulation framework for contaminant transport problems. *Computers & Mathematics with Applications*, 51(11):1633–1646, 2006. ISSN 0898-1221.

Christine Duguay and Fatah Chetouane. Modeling and improving emergency department systems using discrete event simulation. *Simulation*, 83(4):311320, 2007.

Chathura C. Ekanayake, Marlon Dumas, Luciano GarcIa-Banuelos, and Marcello La Rosa. Slice, mine and dice: Complexity-aware automated discovery of business process models. In *Business Process Management*, volume 8094, pages 49–64. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-40175-6.

Elia El-Darzi, Christos Vasilakis, Thierry Chaussalet, and Petter Millard. A simulation modelling approach to evaluating length of stay, occupancy, emptiness and bed blocking in a hospital geriatric department. *Health Care Management Science*, 1(2):143, 1998. ISSN 1572-9389.

Haytham Elghazel, Veronique Deslandres, Kassem Kallel, and Alain Dussauchoy. Clinical pathway analysis using graph-based approach and markov models. In *2007 2nd International Conference on Digital Information Management*, volume 1, pages 279–284, Oct 2007.

Laurent Fauchier, Adeline Samson, Gwendoline Chaize, Anne-Françoise Gaudin, Alexandre Vainchtock, Cécile Bailly, and Francois-Emery Cotté. Cause of death in patients with atrial fibrillation admitted to french hospitals in 2012: a nationwide database study. *Open Heart*, 2(1), 2015.

Laurent Fauchier, Gwendoline Chaize, Anne-Francoise Gaudin, Alexandre Vainchtock, Sophie K. Rushton-Smith, and Francois-Emery Cotté. Predictive ability of has-bled, hemorr2hages, and {ATRIA} bleeding risk scores in patients with atrial fibrillation. a french nationwide cross-sectional study. *International Journal of Cardiology*, 217:85 – 91, 2016. ISSN 0167-5273.

Herve Fernandez, Chabbert-Buffet Nathalie, M. Koskas, and A. Nazac. Epidémiologie du fibrome utérin en france en 20102012 dans les établissements de santé analyse des données du programme médicalisé des systèmes d'information (pmsi). *Journal de Gynécologie Obstétrique et Biologie de la Reproduction*, 43(8):616 – 628, 2014. ISSN 0368-2315.

David Ferrin, Martin J. Miller, Sherry Wininger, and Michael S. Neuendorf. Analyzing incentives and scheduling in a major metropolitan hospital operating room through simulation. In *Proceedings of the 2004 Winter Simulation Conference*, volume 2, pages 1975–1980 vol.2, Dec 2004.

David Fone, Sandra Hollinghurst, Mark Temple, Alison Round, Nathan Lester, Alison Weightman, Katherine Roberts, Edward Coyle, Gwyn Bevan, and Stephen R. Palmer. Systematic review of the use and value of computer simulation modelling in population health and health care delivery. *J Public Hlth Med*, 25 (4):325335, 2003.

Thomas Franck, Vincent Augusto, Xiaolan Xie, Regis Gonthier, and Emilie Achour. Performance evaluation of an integrated care for geriatric departments using discrete-event simulation. In *Proceedings of the 2015 Winter Simulation Conference (WSC)*, WSC'15, pages 1331–1342. IEEE Press, Dec 2015. ISBN 978-1-4673-9741-4.

Gilles Freyer, Florian Scotte, Isabelle Borget, Amandine Bruyas, Alexandre Vainchtock, and Christos Chouaid. Hospitalisations pour neutropénie fébrile chimio-induite en france en 20102011 : impact clinique et caractéristiques des patients à partir des données de la base {PMSI}. *Bulletin du Cancer*, 103 (6):552 – 560, 2016. ISSN 0007-4551.

D. Girard, D. Antoine, and D. Che. Epidemiology of pulmonary tuberculosis in france. can the hospital discharge database be a reliable source of information? *Médecine et Maladies Infectieuses*, 44(1112): 509 – 514, 2014. ISSN 0399-077X.

Besma Glaa, Slim Hammadi, and Christian Tahon. Modeling the emergency path handling and emergency department simulation. In *2006 IEEE International Conference on Systems, Man and Cybernetics*, volume 6, pages 4585–4590, Oct 2006.

Fred Glover. Future paths for integer programming and links to artificial intelligence. *Comput. Oper. Res.*, 13(5):533–549, May 1986.

Carlos Gomes, Bernardo Almada-Lobo, Jose Borges, and Carlos Soares. *Integrating Data Mining and Optimization Techniques on Surgery Scheduling*, pages 589–602. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-35526-4.

Daniela Grigori, Fabio Casati, Malu Castellanos, Umeshwar Dayal, Mehmet Sayal, and Ming-Chien Shan. Business process intelligence. *Comput. Ind.*, 53(3):321–343, April 2004. ISSN 0166-3615.

Murat M. Günal and Mike Pidd. Discrete event simulation for performance modelling in health care: a review of the literature. *Journal of Simulation*, 4(1):42–51, 2010. ISSN 1747-7786.

Christian W. Gunther. *Process Mining in Flexible Environments*. PhD thesis, Eindhoven University of Technology, 2009.

Christian W. Gunther and Wil M.P. van der Aalst. Fuzzy mining adaptive process simplification based on multi-perspective metrics. In *Business Process Management*, volume 4714 of *Lecture Notes in Computer Science*, pages 328–343. Springer Berlin Heidelberg, 2007. ISBN 978-3-540-75182-3.

Christian W. Gunther, Anne Rozinat, Wil M.P. van der Aalst, and Kenny van Uden. Monitoring deployed application usage with process mining. *BPM Center Report*, November 2008.

Christian W. Gunther, Anne Rozinat, and Wil M.P. van der Aalst. Activity mining by global trace segmentation. In *Business Process Management Workshops*, volume 43, pages 128–139. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-12185-2.

Sabri Hamana, Vincent Augusto, and Xiaolan Xie. Modelling interactions between health institutions in the context of patient care pathway. In *16th Working Conference on Virtual Enterprises (PROVE)*, volume AICT-463 of *Risks and Resilience of Collaborative Networks*, pages 448–455, Albi, France, October 2015. Springer. doi: 10.1007/978-3-319-24141-8_41.

Matthew Herland, Taghi M. Khoshgoftaar, and Randall Wald. A review of data mining using big data in health informatics. *Journal Of Big Data*, 1(1):2, 2014. ISSN 2196-1115.

Lisa M. Hess, Frederick B. Stehman, Michael W. Method, Tess D. Weathers, Paridha Gupta, and Jeanne M. Schilder. Identification of the optimal pathway to reach an accurate diagnosis in the absence of an early detection strategy for ovarian cancer. *Gynecologic Oncology*, 127(3):564 – 568, 2012. ISSN 0090-8258.

Feixiang Huang, Shengyong Wang, and Chien-Chung Chan. Predicting disease by using data mining based on healthcare information system. In *2012 IEEE International Conference on Granular Computing*, pages 191–194, Aug 2012.

Zan Huang and Akhil Kumar. A study of quality and accuracy trade-offs in process mining. *INFORMS J. on Computing*, 24(2):311–327, April 2012. ISSN 1526-5528.

Zhengxing Huang, Xudong Lu, Huilong Duan, and Wu Fan. Summarizing clinical pathways from event logs. *Journal of Biomedical Informatics*, 46(1):111 – 127, 2013. ISSN 1532-0464.

Haruko Iwata, Shusaku Tsumoto, and Shoji Hirano. Data mining based clinical care plan construction. In *2013 International Joint Conference on Awareness Science and Technology Ubi-Media Computing (iCAST 2013 UMEDIA 2013)*, pages 286–292, Nov 2013.

Anders B. Jensen, Pope L. Moseley, Tudor I. Oprea, Sabrina G. Ellesoe, Robert Eriksson, Henriette Schmock, Peter B. Jensen, Lars J. Jensen, and Soren Brunak. Temporal disease trajectories condensed from population-wide registry data covering 6.2 million patients. *Nature Communications*, 5, 2014. ISSN 2041-1723.

Neesha Jothi, Nur'Aini A. Rashid, and Wahidah Husain. Data mining in healthcare a review. *Procedia Computer Science*, 72:306 – 313, 2015. ISSN 1877-0509.

Brian J. Jun, Sheldon H. Jacobson, and James R. Swisher. Application of discrete-event simulation in health care clinics: A survey. *Journal of the Operational Research Society*, 50(2):109123, 1999.

Mohammed Khalilia, Sounak Chakraborty, and Mihail Popescu. Predicting disease risks from highly imbalanced data using random forest. *BMC Medical Informatics and Decision Making*, 11(1):51, 2011. ISSN 1472-6947.

Andrey Khudyakov, Camille Jean, Marija Jankovic, Julie Stal-Le Cardinal, and Jean-Claude Bocquet. *Simulation Methods in the Healthcare Systems*, pages 141–149. Springer International Publishing, Cham, 2014. ISBN 978-3-319-02812-5.

Kathrin Kirchner, Nico Herzberg, Andreas Rogge-Solti, and Mathias Weske. *Embedding Conformance Checking in a Process Intelligence System in Hospital Environments*, pages 126–139. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-36438-9. doi: 10.1007/978-3-642-36438-9_9.

Angelina P. Kurniati, Owen Johnson, David Hogg, and Geoff Hall. Process mining in oncology: A literature review. In *2016 6th International Conference on Information Communication and Management (ICICM)*, pages 291–297, Oct 2016. doi: 10.1109/INFOCOMAN.2016.7784260.

Andrew Kusiak, Bradley Diwon, and Shital Shah. Predicting survival time for kidney dialysis patients: a data mining approach. *Computers in Biology and Medicine*, 35(4):311 – 327, 2005.

Martin Lang, Thomas Burkle, Susanne Laumann, and Hans-Ulrich Prokosch. Process mining for clinical workflows: challenges and current limitations. *Stud Health Technol Inform.*, 136:229, 2008.

Averill M. Law and W. David Kelton. *Simulation Modeling and Analysis*. McGraw-Hill Higher Education, 3rd edition, 2000. ISBN 0070592926.

Eva K. Lee, Fan Yuan, Daniel A. Hirsh, Michael D. Mallory, and Harold K. Simon. A clinical decision tool for predicting patient care characteristics: Patients returning within 72 hours in the emergency department. *AMIA Annu Symp Proc*, 2012:495–504, Nov 2012. ISSN 1942-597X.

Eva K. Lee, Hany Y. Atallah, Michael D. Wright, Calvin Thomas, Eleanor T. Post, Daniel T. Wu, and Leon L. Haley Jr. *Systems Analytics : Modeling and Optimizing Clinic Workflow and Patient Care*, pages 261–302. John Wiley and Sons, Inc., 2016. ISBN 9781118919408.

Ting-Ting Lee, Chieh-Yu Liu, Ya-Hui Kuo, Mary E. Mills, Jian-Guo Fong, and Cheyu Hung. Application of data mining to the identification of critical factors in patient falls using a web-based reporting system. *International Journal of Medical Informatics*, 80(2):141–150, 2011. ISSN 1386-5056.

Fu-ren Lin, Shien-chao Chou, Shung-mei Pan, and Yao-mei Chen. Mining time dependency patterns in clinical pathways. *International Journal of Medical Informatics*, 62(1):11–25, 2001. ISSN 1386-5056.

Fu-ren Lin, Lu-shih Hsieh, and Shung-mei Pan. Learning clinical pathway patterns by hidden markov model. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, pages 142a–142a, Jan 2005.

Stefano Lucidi, Massimo Maurici, Luca Paulon, Francesco Rinaldi, and Massimo Roma. A simulation-based multiobjective optimization approach for health care service management. *IEEE Transactions on Automation Science and Engineering*, 13(4):1480–1491, Oct 2016. ISSN 1545-5955.

Linh Thao Ly, Stefanie Rinderle, Peter Dadam, and Manfred Reichert. *Mining Staff Assignment Rules from Event-Based Data*, pages 177–190. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. ISBN 978-3-540-32596-3.

Dan Maljovec, Bei Wang, Paul Rosen, Andrea Alfonsi, Giovanni Pastore, Cristian Rabiti, and Valerio Pascucci. Rethinking sensitivity analysis of nuclear simulations with topology. In *2016 IEEE Pacific Visualization Symposium (PacificVis)*, pages 64–71, April 2016.

Ronny Mans, Helen Schonenberg, Giorgio Leonardi, Silvia Panzarasa, Anna Cavallini, Silvana Quaglini, and Wil M.P. van der Aalst. Process mining techniques: an application to stroke care. In *Proceedings of XXIst International Congress of the European Federation for Medical Informatics*, 2008.

Ronny Mans, Hellen Schonenberg, Min Song, Wil M.P. van der Aalst, and Piet J.M. Bakker. *Application of Process Mining in Healthcare – A Case Study in a Dutch Hospital*, pages 425–438. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. ISBN 978-3-540-92219-3.

Ronny Mans, Hajo Reijers, Michiel van Genuchten, and Daniel Wismeijer. Mining processes in dentistry. In *Proceedings of the 2Nd ACM SIGHIT International Health Informatics Symposium*, IHI '12, pages 379–388. ACM, 2012. ISBN 978-1-4503-0781-9. doi: 10.1145/2110363.2110407.

Ronny Mans, Wil M.P. van der Aalst, and Rob J.B. Vanwersch. *Process Mining in Healthcare: Evaluating and Exploiting Operational Healthcare Processes*. Springer International Publishing, 1st edition, 2015. ISBN 978-3-319-16070-2.

Ronny S. Mans, Wil M.P. van der Aalst, Rob J.B. Vanwersch, and Arnold J. Moleman. *Process Mining in Healthcare: Data Challenges When Answering Frequently Posed Questions*, pages 140–153. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-36438-9. doi: 10.1007/978-3-642-36438-9_10.

N. Martin, B. Depaire, and A. Caris. The use of process mining in a business process simulation context: Overview and challenges. In *Computational Intelligence and Data Mining (CIDM), 2014 IEEE Symposium on*, pages 381–388, Dec 2014a.

Niels Martin, Benoit Depaire, and An Caris. Event log knowledge as a complementary simulation model construction input. In *Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH), 2014 International Conference on*, pages 456–462, Aug 2014b.

Thomas H. Marwick, Paul A. Scuffham, and M.G. Myriam Hunink. Selection for early surgery in asymptomatic mitral regurgitation: A markov model. *International Journal of Cardiology*, 165(2):266 – 272, 2013. ISSN 0167-5273.

Rahul Mehra. Global public health problem of sudden cardiac death. *Journal of Electrocardiology*, 40(6, Supplement 1):S118 – S122, 2007. ISSN 0022-0736. {ISCE} 32nd Annual Conference.

Jan Mendling, Hajo A. Reijers, and Jorge Cardoso. What makes process models understandable? In *Business Process Management*, volume 4714, pages 48–63. Springer Berlin Heidelberg, 2007. ISBN 978-3-540-75182-3.

Rym M'Hallah and A.H. Al-Roomi. The planning and scheduling of operating rooms. *Comput. Ind. Eng.*, 78(C):235–248, December 2014. ISSN 0360-8352.

Martin J. Miller, David M. Ferrin, and Jill M. Szymanski. Simulating six sigma improvement ideas for a hospital emergency department. In *Proceedings of the 2003 Winter Simulation Conference, 2003.*, volume 2, pages 1926–1929 vol.2, Dec 2003.

M. Mirowski, Philip R. Reid, Morton M. Mower, Levi Watkins, Vincent L. Gott, James F. Schauble, Alois Langer, M. S. Heilman, Steve A. Kolenik, Robert E. Fischell, and Myron L. Weisfeldt. Termination of malignant ventricular arrhythmias with an implanted automatic defibrillator in human beings. *New England Journal of Medicine*, 303(6):322–324, 1980.

Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997. ISBN 0070428077, 9780070428072.

Stefania Montani, Giorgio Leonardi, Silvana Quaglini, Anna Cavallini, and Giuseppe Micieli. Improving structural medical process comparison by exploiting domain knowledge and mined information. *Artificial Intelligence in Medicine*, 62(1):33–45, 2014.

Lorenzo Moreno, Rosa M. Aguilar, C.A. Martin, Jose D. Pineiro, Jose I. Estevez, Jose L. Sanchez, Jose F. Sigut, and V.I. Jimenez. Patient-centered computer simulation in hospital management. *Journal of Network and Computer Applications*, 21(4):287 – 310, 1998. ISSN 1084-8045.

Ralph Mueller, Christos Alexopoulos, and Leon F. McGinnis. Automatic generation of simulation models for semiconductor manufacturing. In *2007 Winter Simulation Conference*, pages 648–657, Dec 2007.

Thomas H. Naylor and Joseph M. Finger. Verification of computer simulation models. *Management Science*, 14(2):B92–B101, 1967.

Moulaye A. A. Ndiaye, Jean-Francois Petin, Jacques Camerini, and Jean Philippe Georges. Performance assessment of industrial control system during pre-sales uncertain context using automatic colored petri nets model generation. In *2016 International Conference on Control, Decision and Information Technologies (CoDIT)*, pages 671–676, April 2016.

Saul B. Needleman and Christian D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443 – 453, 1970. ISSN 0022-2836.

Olegas Niaksu, Jolita Skinulyte, and Hermine Grubinger Duhaze. *A Systematic Literature Review of Data Mining Applications in Healthcare*, pages 313–324. Springer Berlin Heidelberg, 2014. ISBN 978-3-642-54370-8.

IEEE Task Force on Process Mining (80 authors). Process mining manifesto. In *Business Process Management Workshops: BPM 2011 International Workshops, Clermont-Ferrand, France, August 29, 2011, Revised Selected Papers, Part I*, pages 169–194, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-28108-2.

Marie-Noelle Osmont, Marc Cuggia, Elisabeth Polard, Christine Riou, Frédéric Balusson, and Emmanuel Oger. Utilisation du {PMSI} pour la détection d'effets indésirables médicamenteux. *Thérapie*, 68(4): 285 – 295, 2013. ISSN 0040-5957. {XXVIIIes} Rencontres Nationales de Pharmacologie et Recherche Clinique, Innovation et Evaluation des Technologies de Santé, Tables rondes {GIENS} 7 au 9 octobre 2012.

Nicolas Padoy, Tobias Blum, Hubertus Feussner, Marie-Odile Berger, and Nassir Navab. On-line recognition of surgical activity for monitoring in the operating room. In *Proceedings of the 20th National Conf. on Innovative Applications of Artificial Intelligence - Volume 3*, pages 1718–1724, 2008. ISBN 978-1-57735-368-3.

P.-B. Pages, J. Cottenet, A.-S. Mariet, A. Bernard, and C. Quantin. Mesure de la qualité des soins à partir de la base de données nationale du pmsi. étude de la mortalité hospitalière après résections pulmonaires pour cancer. *Revue d'Epidémiologie et de Santé Publique*, 64, Supplement 1:S13 –, 2016. ISSN 0398-7620.

Canan Pehlivan. *Design and flow control of stochastic health care networks without waiting rooms: A perinatal application*. PhD thesis, Ecole Nationale Supérieure des Mines de Saint-Etienne, France, 2014.

Canan Pehlivan, Vincent Augusto, and Xiaolan Xie. Admission control in a pure loss healthcare network: Mdp and des approach. In *Proceedings of the 2013 Winter Simulation Conference*, WSC '13, pages 54–65, Piscataway, NJ, USA, 2013a. IEEE Press. ISBN 978-1-4799-2077-8.

Canan Pehlivan, Vincent Augusto, and Xiaolan Xie. Admission control in a pure loss healthcare network: Mdp and des approach. In *2013 Winter Simulations Conference (WSC)*, pages 54–65, Dec 2013b.

Viviana Perdomo, Vincent Augusto, and Xiaolan Xie. Operating theatre scheduling using lagrangian relaxation. In *2006 International Conference on Service Systems and Service Management*, volume 2, pages 1234–1239, Oct 2006.

L. Petit, E. Laurent, Z. Maakaroun-Vermesse, T. Odent, L. Bernard, and L. Grammatico-Guillon. Facteurs de risque d'hospitalisation prolongée pour infection ostéo-articulaire pédiatrique en france à partir du pmsi 2013. *Revue d'Épidémiologie et de Santé Publique*, 64, Supplement 1:S23 –, 2016. ISSN 0398-7620.

Gergely Popovics, Csaba Kardos, Andras Pfeiffer, Botond Kadar, Zoltan Ven, and Laszlo Monostori. Automatic simulation model generation supported by data stored in low level controllers. *IFAC Proceedings Volumes*, 45(6):242 – 247, 2012. ISSN 1474-6670.

Martin Prodel, Vincent Augusto, and Xiaolan Xie. Hospitalization admission control of emergency patients using markovian decision processes and discrete event simulation. In *Proceedings of the 2014 Winter Simulation Conference*, WSC '14, pages 1433–1444, Piscataway, NJ, USA, 2014. IEEE Press.

Martin Prodel, Vincent Augusto, Xie Xiaolan, Baptiste Jouaneton, and Ludovic Lamarsalle. Discovery of patient pathways from a national hospital database using process mining and integer linear programming. In *Automation Science and Engineering (CASE), 2015 IEEE Int. Conf. on*, pages 1409–1414, 2015.

Jean-Marie Proth and XIaolan Xie. *Petri Nets: A Tool for Design and Management of Manufacturing Systems*. Wiley Subscription Services, Inc., A Wiley Company, 1st edition, 1996. ISBN 047196770X.

Hude Quan, Vijaya Sundararajan, Patricia Halfon, Andrew Fong, Bernard Burnand, Jean-Christophe Luthi, L. Duncan Saunders, Cynthia A. Beck, Thomas E. Feasby, and William A. Ghali. Coding algorithms for defining comorbidities in icd-9-cm and icd-10 administrative data. *Medical Care*, 43(11), 2005. ISSN 0025-7079.

J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993. ISBN 1-55860-238-0.

Abdur Rais and Ana Viana. Operations research in healthcare: a survey. *International Transactions in Operational Research*, 18(1):1–31, 2011. ISSN 1475-3995.

Uzma Raja, Tara Mitchell, Timothy Day, and James M. Hardin. Text mining in healthcare. applications and opportunities. *Journal of healthcare information management JHIM*, 22(3):52–6, 2008.

Periyasamy Rajendran and Muthusamy Madheswaran. Hybrid medical image classification using association rule mining with decision tree algorithm. *Journal of Computing*, 2, 2010.

Francisco J. Ramis, Jorge L. Palma, and Felipe F. Baesler. The use of simulation for process improvement at an ambulatory surgery center. In *Proceeding of the 2001 Winter Simulation Conference*, volume 2, pages 1401–1404 vol.2, 2001.

Samik Raychaudhuri. Introduction to monte carlo simulation. In *Proceedings of the 40th Conference on Winter Simulation*, WSC '08, pages 91–100. Winter Simulation Conference, 2008. ISBN 978-1-4244-2708-6.

Thomas D. Rea, Mickey S. Eisenberg, Greg Sinibaldi, and Roger D. White. Incidence of ems-treated out-of-hospital cardiac arrest in the united states. *Resuscitation*, 63(1):17–24, march 2004. ISSN 0300-9572.

Alvaro Rebuge and Diogo R. Ferreira. Business process analysis in healthcare environments: A methodology based on process mining. *Inf. Syst.*, 37(2):99–116, April 2012. ISSN 0306-4379.

Eric Rojas, Jorge Munoz-Gama, Marcos Sepúlveda, and Daniel Capurro. Process mining in healthcare. *J. of Biomedical Informatics*, 61(C):224–236, June 2016. ISSN 1532-0464.

Thomas Rotter, Leigh Kinsman, Erica L. James, Andreas Machotta, Holger Gothe, Jon Willis, Pamela Snow, and Joachim Kugler. Clinical pathways: effects on professional practice, patient outcomes, length of stay and hospital costs. *Cochrane Database of Systematic Reviews*, 2010. ISSN 1465-1858. doi: 10.1002/14651858.CD006632.pub2. EPOC.

Anne Rozinat and Wil M.P. van der Aalst. Decision mining in business processes. *BETA Working Paper Series*, page WP 164, 2006a.

Anne Rozinat and Wil M.P. van der Aalst. Decision mining in business processes. *BPM Center Report*, BPM-06-10, 2006b.

Anne Rozinat and Wil M.P. van der Aalst. *Decision Mining in ProM*, pages 420–425. BPM'06. Springer-Verlag, Berlin, Heidelberg, 2006c. ISBN 3-540-38901-6, 978-3-540-38901-9.

Anne Rozinat and Wil M.P. van der Aalst. Conformance checking of processes based on monitoring real behavior. *Information Systems*, 33(1):64 – 95, 2008. ISSN 0306-4379.

Anne Rozinat, Ana K.A. de Medeiros, Christian W. Gunther, Ton A.J.M.M. Weijters, and Wil M.P. van der Aalst. The need for a process mining evaluation framework in research and practice. In *Business Process Management Workshops*, volume 4928, pages 84–89. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-78237-7.

Anne Rozinat, Ronny S. Mans, Min Song, and Wil M.P. van der Aalst. Discovering simulation models. *Inf. Syst.*, 34(3):305–327, May 2009. ISSN 0306-4379.

Andrea Saltelli, Marco Ratto, Terry Andres, Francesca Campolongo, Jessica Cariboni, Debora Gatelli, Michaela Saisana, and Stefano Tarantola. *Introduction to Sensitivity Analysis*, pages 1–51. John Wiley & Sons, Ltd, 2008. ISBN 9780470725184.

Robert G. Sargent. Verification and validation of simulation models. In *Proceedings of the Winter Simulation Conference*, WSC '11, pages 183–198. Winter Simulation Conference, 2011.

Comilla Sasson, Mary A.M. Rogers, Jason Dahl, and Arthur L. Kellermann. Predictors of survival from out-of-hospital cardiac arrest. *Circulation: Cardiovascular Quality and Outcomes*, 3(1):63–81, 2010. ISSN 1941-7705.

Stewart Schlesinger. Terminology for model credibility. *Simulation*, 32(3):103–104, 1979.

Florian Scotte, Nicolas Martelli, Alexandre Vainchtock, and Isabelle Borget. The cost of thromboembolic events in hospitalized patients with breast or prostate cancer in france. *Advances in Therapy*, 32(2): 138–147, 2015. ISSN 1865-8652.

Ahmad Shahin, Walid Moudani, Fadi Chakik, and Mohamad Khalil. Data mining in healthcare information systems: Case studies in northern lebanon. In *The Third International Conference on e-Technologies and Networks for Development (ICeND2014)*, pages 151–155, April 2014.

Maria Shitkova, Victor Taratukhin, and Jorg Becker. Towards a methodology and a tool for modeling clinical pathways. *Procedia Computer Science*, 63:205 – 212, 2015. ISSN 1877-0509.

David Sinreich and Yariv N. Marmor. A simple and intuitive simulation tool for analyzing emergency department operations. In *Proceedings of the 2004 Winter Simulation Conference, 2004.*, volume 2, pages 1994–2002 vol.2, Dec 2004.

Suriadi Suriadi, Chun Ouyang, Wil M. P. van der Aalst, and Arthur H. M. ter Hofstede. *Root Cause Analysis with Enriched Process Logs*, pages 174–186. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. ISBN 978-3-642-36285-9.

Soemon Takakuwa and Daisuke Katagiri. Modeling of patient flows in a large-scale outpatient hospital ward by making use of electronic medical records. In *2007 Winter Simulation Conference*, pages 1523–1531, Dec 2007.

Soemon Takakuwa and Hiroko Shiozaki. Functional analysis for operating emergency department of a general hospital. In *Proceedings of the 36th Conference on Winter Simulation*, WSC '04, pages 2003–2011. Winter Simulation Conference, 2004. ISBN 0-7803-8786-4.

Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining*. Addison-Wesley, 2005. ISBN 0321321367.

Mohamed Touati, Ludovic Lamarsalle, Stéphane Moreau, Françoise Vergnenègre, Sophie Lefort, Catherine Brillat, Laetitia Jeannet, Aline Lagarde, Annick Daulange, Arnaud Jaccard, Alain Vergnenègre, and Dominique Bordessoule. Cost savings of home bortezomib injection in patients with multiple myeloma

treated by a combination care in outpatient hospital and hospital care at home. *Supportive Care in Cancer*, 24(12):5007–5014, 2016. ISSN 1433-7339.

Shusaku Tsumoto, Haruko Iwata, Shoji Hirano, and Yuko Tsumoto. Similarity-based behavior and process mining of medical practices. *Future Generation Computer Systems*, 33:21 – 31, 2014. ISSN 0167-739X. doi: http://dx.doi.org/10.1016/j.future.2013.10.014. Special Section on Applications of Intelligent Data and Knowledge Processing Technologies; Guest Editor: Dominik lzak.

M. Uhart, C. Blein, M. L'Azou, L. Thomas, and L. Durand. Costs of dengue in three french territories of the americas: an analysis of the hospital medical information system (pmsi) database. *The European Journal of Health Economics*, 17(4):497–503, 2016. ISSN 1618-7601.

Wil M.P. van der Aalst. Workflow mining: Discovering process models from event logs. *Computers in industry*, 16:1128–1142, 2004.

Wil M.P. van der Aalst. *Business Process Simulation Revisited*, pages 1–14. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. ISBN 978-3-642-15723-3.

Wil M.P. van der Aalst. *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Springer Publishing Company, Incorporated, 1st edition, 2011. ISBN 3642193447, 9783642193446.

Wil M.P. van der Aalst. A decade of business process management conferences: Personal reflections on a developing discipline. In *Business Process Management*, volume 7481 of *Lecture Notes in Computer Science*, pages 1–16. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-32884-8.

Wil M.P. van der Aalst and Ton A. J. M. M. Weijters. Process mining: A research agenda. *Comput. Ind.*, 53(3):231–244, April 2004. ISSN 0166-3615.

Wil M.P. van der Aalst, Arthur Hofstede, and Mathias Weske. Business process management: A survey. In *Proc. of the 2003 Int. Conf. on BPM*, pages 1–12. Springer-Verlag, 2003. ISBN 3-540-40318-3.

Wil M.P. van der Aalst, Ana K.A. de Medeiros, and Ton A.J.M.M. Weijters. Genetic process mining. In *Applications and Theory of Petri Nets 2005*, volume 3536, pages 48–69. Springer Berlin Heidelberg, 2005. ISBN 978-3-540-26301-2.

Jan Martijn E.M. van der Werf, Boudewijn F. van Dongen, Cor A.J. Hurkens, and Alexander Serebrenik. Process discovery using integer linear programming. In *Applications and Theory of Petri Nets*, volume 5062 of *Lecture Notes in Computer Science*, pages 368–387. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-68745-0.

Boudewijn F. van Dongen, Ana K.A. de Medeiros, Eric H.M.W. Verbeek, Ton A.J.M.M. Weijters, and Wil M.P. van der Aalst. The prom framework: A new era in process mining tool support. In *Applications and Theory of Petri Nets 2005*, volume 3536 of *Lecture Notes in Computer Science*, pages 444–454. Springer Berlin Heidelberg, 2005. ISBN 978-3-540-26301-2.

Christos Vasilakis and Adele Marshall. Modelling nationwide hospital length of stay: opening the black box. *Journal of the Operational Research Society*, 56(7):862–869, 2005. ISSN 1476-9360.

Eric H.M.W. Verbeek and Wil M. P. van der Aalst. *An Experimental Evaluation of Passage-Based Process Discovery*, pages 205–210. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-36285-9. doi: 10.1007/978-3-642-36285-9_21.

Eric H.M.W. Verbeek and Wil M.P. van der Aalst. Decomposed process mining: The ilp case. In *Business Process Management Workshops*, volume 202 of *Lecture Notes in Business Information Processing*, pages 264–276. Springer International Publishing, 2015. ISBN 978-3-319-15894-5.

Riikka Vuokko, Paivi Makela-Bengs, Hannele Hypponen, Minna Lindqvist, and Persephone Doupi. Impacts of structuring the electronic health record: Results of a systematic literature review from the perspective of secondary use of patient data. *International Journal of Medical Informatics*, 97:293 – 303, 2017. ISSN 1386-5056.

Jianmin Wang, Raymond K. Wong, Jianwei Ding, Qinlong Guo, and Lijie Wen. Efficient selection of process mining algorithms. *Services Computing, IEEE Transactions on*, 6(4):484–496, Oct 2013. ISSN 1939-1374.

Philip Weber, Behzad Bordbar, and Peter Tino. A framework for the analysis of process mining algorithms. *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, 43(2):303–317, March 2013. ISSN 2168-2216.

Ton A.J.M.M. Weijters, Wil M.P. van der Aalst, and Ana K.A. de Medeiros. Process mining with the heuristics miner-algorithm. *Technische Universiteit Eindhoven, Tech. Rep. WP*, 166:1–34, 2006.

Allan Wiinamaki and Rainer Dronzek. Emergency departments i: Using simulation in the architectural concept phase of an emergency department design. In *Proceedings of the 35th Winter Simulation Conference: Driving Innovation*, WSC'03, pages 1912–1916. Winter Simulation Conference, 2003. ISBN 0-7803-8132-7.

Athula Wijewickrama and Soemon Takakuwa. Simulation analysis of appointment scheduling in an outpatient department of internal medicine. In *Proceedings of the 37th Conference on Winter Simulation*, WSC '05, pages 2264–2273. Winter Simulation Conference, 2005. ISBN 0-7803-9519-0.

Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005. ISBN 0120884070.

Sichao Wu and Henning S. Mortveit. A general framework for experimental design, uncertainty quantification and sensitivity analysis of computer simulation models. In *2015 Winter Simulation Conference (WSC)*, pages 1139–1150, Dec 2015.

Xindong Wu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan, Angus Ng, Bing Liu, Philip S. Yu, Zhi-Hua Zhou, Michael Steinbach, David J. Hand, and Dan Steinberg. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1):1–37, 2008. ISSN 0219-3116.

Xuanyang Xie, Xi Li, Shouhong Wan, and Yuchang Gong. *Mining X-Ray Images of SARS Patients*, pages 282–294. Springer Berlin Heidelberg, 2006. ISBN 978-3-540-32548-2.

Wei Yang and Qiang Su. Process mining for clinical pathway: Literature review and future directions. In *2014 11th International Conference on Service Systems and Service Management (ICSSSM)*, pages 1–5, June 2014. doi: 10.1109/ICSSSM.2014.6943412.

Wen Yao and Akhil Kumar. Conflexflow: Integrating flexible clinical pathways into clinical decision support systems using context and rules. *Decision Support Systems*, 55(2):499–515, 2013. ISSN 0167-9236.

Amy M.F. Yen and Hsiu-Hsi Chen. Stochastic models for multiple pathways of temporal natural history on co-morbidity of chronic disease. *Computational Statistics & Data Analysis*, 57(1):570 – 588, 2013. ISSN 0167-9473.

Illhoi Yoo, Patricia Alafaireet, Miroslav Marinov, Keila Pena-Hernandez, Rajitha Gopidi, Jia-Fu Chang, and Lei Hua. Data mining in healthcare and biomedicine: A survey of the literature. *Journal of Medical Systems*, 36(4):2431–2448, 2012. ISSN 1573-689X.

Yiye Zhang, Rema Padman, and Nirav Patel. Paving the cowpath: Learning and visualizing clinical pathways from electronic health record data. *Journal of Biomedical Informatics*, 58:186 – 197, 2015. ISSN 1532-0464.

J.Y. Zhou. Process mining: Acquiring objective process information for healthcare process management with the crisp-dm framework. Master's thesis, Eindhoven University of Technology, Eindhoven, 2009.

Zhi-Hua Zhou. *Ensemble Methods: Foundations and Algorithms*. Chapman & Hall/CRC, 1st edition, 2012. ISBN 1439830037, 9781439830031.

Zhichao Zhou, Yong Wang, and Lin Li. Process mining based modeling and analysis of workflows in clinical care - a case study in a chicago outpatient clinic. In *Networking, Sensing and Control (ICNSC), 2014 IEEE 11th International Conference on*, pages 590–595, April 2014.

Lian-Zhang Zhu and Fan-Sheng Kong. Automatic conversion from uml to cpn for software performance evaluation. *Procedia Engineering*, 29:2682 – 2686, 2012. ISSN 1877-7058.

Kiyana Zolfaghar, Naren Meadem, Ankur Teredesai, Senjuti B. Roy, Si-Chi Chin, and Brian Muckian. Big data solutions for predicting risk-of-readmission for congestive heart failure patients. In *2013 IEEE International Conference on Big Data*, pages 64–71, Oct 2013.

École Nationale Supérieure des Mines
de Saint-Étienne

NNT : 2017LYSEM009

Martin PRODEL

PROCESS DISCOVERY, ANALYSIS AND SIMULATION OF CLINICAL PATHWAYS USING HEALTH-CARE DATA

Speciality: Industrial Engineering

Keywords: Mathematical optimisation, process mining, operation research, data mining, health data, clinical pathway, simulation, predictive models.

Abstract: During the last two decades, the amount of data collected in Information Systems has drastically increased. This large amount of data is highly valuable to reveal important patterns. This reality applies to health-care where the computerization is still an ongoing process. Health-care systems are characterized by the inherent variability and complexity of care management and disease evolution. Existing methods from the fields of process mining, data mining and mathematical modeling cannot handle large-sized and variable event logs. Our goal is to develop an extensive methodology to turn health data from event logs into simulation models of clinical pathways. We first introduce a mathematical framework to discover optimal process models. Our approach shows the benefits of combining combinatorial optimization and process mining techniques. Then, we enrich the discovered model with additional data from the log. An innovative combination of a sequence alignment algorithm and of classical data mining techniques is used to analyse path choices within long-term clinical pathways. The approach is demonstrated to be suitable for noisy and large logs. Finally, we propose an automatic procedure to convert static models of clinical pathways into dynamic simulation models. The resulting models perform sensitivity analyses to quantify the impact of determinant factors on several key performance indicators related to care processes. They are also used to evaluate what-if scenarios, such as the impacts of a new drug (or medical device) on both short and long-term aspects of clinical pathways. The presented methodology was proven to be highly reusable on various medical fields and on any source of event logs. Using the national French database of all the hospital events from 2006 to 2015, an extensive case study on cardiovascular diseases is presented to show the efficiency of the proposed framework. Numerical results and relevant graphical representations are obtained and provide brand new knowledge to health practitioners and decision-makers.

École Nationale Supérieure des Mines
de Saint-Étienne

NNT : 2017LYSEM009

Martin PRODEL

MODÉLISATION AUTOMATIQUE ET SIMULATION DE PARCOURS DE SOINS À PARTIR DE BASES DE DONNÉES DE SANTÉ

Spécialité : Génie Industriel

Mots clefs : Optimisation mathématique, fouille de processus, recherche opérationnelle, fouille de données, données de santé, parcours de soin, simulation, modèles prédictifs.

Résumé : Les deux dernières décennies ont été marquées par une augmentation significative des données collectées dans les systèmes d'informations. Cette masse de données rendue disponible contient des informations riches et peu exploitées. Cette réalité s'applique au secteur de la santé où l'informatisation est un réel enjeu pour l'amélioration de l'efficience et de la qualité des soins. Les méthodes existantes dans les domaines de l'extraction de processus, de l'exploration de données et de la modélisation mathématique ne parviennent pas à gérer des données aussi hétérogènes et volumineuses que celles de la santé. Notre objectif est de développer une méthodologie complète pour transformer des données de santé brutes en modèles de simulation des parcours de soins cliniques. Nous introduisons d'abord un cadre mathématique dédié à la découverte de modèles décrivant les parcours de soin, en combinant optimisation combinatoire et Process Mining. Ensuite, nous enrichissons ce modèle par l'utilisation conjointe d'un algorithme d'alignement de séquences et de techniques classiques de Data Mining. Notre approche est capable de gérer des données bruitées et de grande taille. Enfin, nous proposons une procédure pour la conversion automatique d'un modèle descriptif des parcours de soins en un modèle de simulation dynamique. Après validation, le modèle obtenu est exécuté pour effectuer des analyses de sensibilité et évaluer de nouveaux scénarios. Un cas d'étude sur les maladies cardiovasculaires est présenté, avec l'utilisation de la base nationale des hospitalisations entre 2006 et 2015. La méthodologie présentée dans cette thèse est entièrement réutilisable dans d'autres aires thérapeutiques et sur d'autres sources de données de santé.