ROYAUME DU MAROC
Université Mohammed V - RABAT -
جامعة محمد الخامس – الرباط –
Faculté des sciences
كلية العلوم
CENTRE D'ETUDES DOCTORALES - SCIENCES ET TECHNOLOGIES

N° d'ordre : **3462**

THESE

En vue de l'obtention du : ***DOCTORAT***

Structure de Recherche : Laboratoire de Recherche en Informatique et Télécommunications (LRIT)

Discipline : Sciences de l'ingénieur

Spécialité : Informatique

Présentée et soutenue le 04 - 06 - 2021 par :

Zineb LAMGHARI

Business Process Improvement using Process Mining Techniques

JURY

Salma MOULINE	PES	Faculté des Sciences, Université Mohammed V - Rabat	Présidente
Abderrahim EL QADI	PES	École Nationale Supérieure d'Arts et Métiers, Université Mohammed V - Rabat	Rapporteur / Examinateur
Khadija RHOULAMI	PH	Faculté des Sciences, Université Mohammed V - Rabat	Rapporteuse / Examinatrice
Hamid ELGHAZI	PH	Institut National des Postes et Télécommunications, Rabat	Rapporteur / Examinateur
Maryam RADGUI	PH	Institut National de Statistique et d'Economie Appliquée, Rabat	Examinatrice
Moulay Driss RAHMANI	PES	Faculté des Sciences, Université Mohammed V - Rabat	Directeur de thèse
Rajaa SAIDI	PES	Institut National de Statistique et d'Economie Appliquée, Rabat	Co-directrice de thèse

Année Universitaire : 2020 - 2021

✉ Faculté des Sciences, avenue Ibn Battouta, BP. 1014 RP, Rabat –Maroc
☎ 00212 (05) 37 77 18 76 🖷 00212(05) 37 77 42 61; http://www. fsr.um5.ac.ma

Acknowledgement

The work presented in this dissertation was carried out at the Laboratory of Research in Computer Science and Telecommunications (**LRIT-CNRST 29)** of the Faculty of Sciences of Rabat (**FSR**), University Mohammed V in Rabat - Morocco, under the direction of Mr. **Moulay Driss RAHMANI**, Professor of Higher Education at the Faculty of Sciences of Rabat and with the co-direction of Mrs. **Rajaa SAIDI**, Professor of Higher Education at the National Institute of Statistics and Applied Economics (INSEA).

I start by expressing my deepest gratitude to my director of thesis **Mr. Moulay Driss RAHMANI**, Professor of Higher Education at the Faculty of Sciences of Rabat. I express here my deep gratitude to him and the respect I feel for him.

I would like to express my thanks to my co-director of thesis Mrs. **Rajaa SAIDI**, Professor of Higher Education at the National Institute of Statistics and Applied Economics (INSEA). I thank her very much for those years of support, for her precious scientific and human advices. I learned a lot from her professionally and humanly.

My gratitude, profound respect and thanks to Mrs. **Salma MOULINE**, Professor of Higher Education at the Faculty of Sciences of Rabat, for accepting to report my work and participate in the jury.

Special thanks are addressed to Mr. **Abderrahim EL QADI**, Professor of Higher Education at National School of Arts and Crafts of Rabat, for accepting to report my work and to participate in the jury.

I would also like to thank Mrs. **Khadija RHOULAMI**, Associate Professor at the Faculty of Sciences of Rabat. I am deeply grateful to them for agreeing to report this thesis and to be among the referees of this thesis.

I would also like to thank Mr. **Hamid ELGHAZI**, Associate Professor at National Institut of Posts and Telecommunications (INPT), for accepting to review my work and to participate in the jury.

Many thanks to Mrs. **Maryam RADGUI**, Associate Professor at the National Institute of Statistics and Applied Economics (INSEA), who agreed to examine this work. I am particularly pleased that she had read and appreciated my work.

I also want to thank all the people that I have met during the thesis preparation period, colleagues and friends, who have contributed to making these years pleasant.

At last, but not at least, I am gratefully indebted to my parents, husband and brothers, for their unfailing support, permanent presence, unconditional love and encouragement. My sincere acknowledgments are devoted to the rest of my family.

Abstract

Nowadays, many business process improvement methodologies have been developed and implemented in organisations. These methodologies allow overcoming business improvement challenges in a continuous Business Process Management (BPM) context.
Among these business process improvements methodologies, we focus on the Process Mining methodology. Process mining is a scientific discipline that combines machine learning algorithms and business process design methods, based on recorded information systems data in the form of event logs. This is done to discover (process discovery), analyse (conformance checking), and improve (enhancement) processes.
In this work, we aim to improve business processes using process mining techniques. Indeed, we use process execution data, to recognize and understand ground realities of processes that are being followed inside organizations, thus reaching their improvement or redesign. To do so, we address research challenges in which improvement operations of business processes are required. We also look beyond the impact of event logs quality on business process structures and its improvement process.

Keywords: *Business Process; business process improvement; process mining; improvement indicators; complexity; discovery tehcnique.*

Résumé

De nos jours, les entreprises utilisent diverses méthodologies pour analyser leurs processus métiers, ce qui leur permet d'apporter des améliorations en terme d'efficacité et d'efficience, ainsi que l'optimisation de ces derniers dans le contexte de la gestion des processus métiers (BPM).
Parmi les méthodologies d'amélioration des processus métiers qui ont été élaborées et déployées, on trouve la fouille des processus métiers (Process Mining). Cette méthodologie se situe entre l'intelligence computationnelle et la fouille de données d'une part, et la modélisation des processus métiers d'autre part. Les techniques de cette méthodologie visent à extraire les connaissances des journaux d'évènements, enregistrées lors de l'exécution des processus métiers, afin de les découvrir, les contrôler et les améliorer.
Dans cette thèse, nous abordons un ensemble de défis pour lesquels une vision d'amélioration des processus métiers est nécessaire, prenant en considération la qualité des données d'exécution et les différentes structures des processus métiers résultants.

Mots clés: *Processus métiers; amélioration des processus métiers; fouille des processus, indicateurs d'amélioration, complexité, technique de découverte.*

Résumé Étendu

De nos jours, les entreprises utilisent diverses méthodologies pour analyser leurs processus métiers, ce qui leur permet d'apporter des améliorations en terme d'efficacité et d'efficience, ainsi que l'optimisation de ces derniers dans un contexte de la gestion des processus métiers (BPM).
Parmi les méthodologies d'amélioration des processus métiers qui ont été élaboré et déployé, on trouve la fouille des processus métiers (Process Mining). Cette méthodologie se situe entre l'intelligence computationnelle et la fouille de données d'une part, et la modélisation et l'analyse des processus métiers d'autre part. Les techniques de cette méthodologie visent à extraire les connaissances des journaux d'évènements, enregistrées lors de l'exécution des processus métiers, afin de les découvrir (Process Discovery), les contrôler (Conformance checking) et les améliorer (Enhancement).
Dans cette thèse, nous abordons un ensemble de défis pour lesquels une vision d'amélioration des processus métiers est nécessaire, prenant en considération la qualité des données d'exécution et les différentes structures des processus métiers résultants.
Premièrement, la qualité des données d'exécution a un impact sur la qualité des modèles de processus résultants des techniques de découvert (1). En effet, nous visons à filtrer les données chaotiques qui peuvent se produire spontanément à tout moment de l'exécution du processus. Elles sont correctement consignées, mais ne sont toujours pas souhaités, car elles représentent des activités consignées par le systéme même si elles ne font pas partie du flux de processus principal.
Par conséquent, les activités chaotiques sont fréquentes, distinctes des comportements peu fréquents.
Deuxièment, les processus métiers structurés, qui possédent un modèle simplifié sont compréhensibles et facilement lisibles mais définissent un problème d'amélioration continu que nous avons traité à travers un ensemble d'indicateurs d'amélioration (2). L'accent est mis sur la manière de maintenir une amélioration du processus durant toutes les phases du cycle de vie. L'idée serait d'introduire des indicateurs d'amélioration relatifs à chaque phase du BPM tout en s'alignant aux techniques du process mining.
Troisièment, les processus métiers non-structurés sont incompréhensibles et illisibles. La structure complexe définit un enjeu qui réside dans la manière de maintenir une structuration simplifie du processus (3), en appliquant les techniques du process mining.Dans ce cas, Les améliorations qui doivent être apportées aux processus métiers varient en fonction de sa structure (Structurée ou non-structurée) et c'est ici où réside la difficulté de cette opération d'amélioration.
Par conséquent, ce manuscrit se divise en deux parties avec sept chapitres denses. Il a été rédigé en anglais avec des résumés en anglais et français et compte 133 pages complétées par une liste de 154 références bibliographiques.

Le chapitre 1 présente le contexte général ainsi que le problème traité, les principales contributions et la structuration de la thèse.

Le chapitre 2 présente les concepts de base et les définitions connexes utilisés tout au long de cette thèse. En effet, les concepts et les définitions couvrent les aspects de l'amélioration des processus métiers et du process mining. En outre, le chapitre 2 fournit des préliminaires tels que des notations et des expressions formelles utilisées dans le reste de cette thèse.

Dans le chapitre 3, les travaux connexes sont analysés dans le cadre de process mining liés aux phases de prétraitement et de traitement. Cette partie fournit brièvement le contexte nécessaire pour comprendre les contributions de cette thèse. De plus, ce chapitre illustre le cycle de vie du processus et les avantages technologiques approuvant la sélection du process mining en tant qu'une méthodologie mature en amélioration des processus métiers. Au-delà, notre recherche s'articule autour des approches du Business Process Intelligence en combinaison avec le concept d'extraction minière par procédé (chapitres 4, 5 et 6).

Dans le chapitre 4, on développe une approche permettant la reconnaissance des activités chaotiques à partir des journaux d'évènements afin d'affiner la qualité des journaux d'évènement. Ensuite, nous les utilisons pour générer un modèle de processus plus précis à la base des algorithmes de découverte. Ce processus métier est qualifèié par quatre métriques de qualité (Fitness, Generalization, Precision et Simplicity).
Généralement, l'approche proposée se compose de 4 phases :
1. Préparation des données : l'extraction au format adéquate et le filtrage basique des données.
2. La sélection des données : définir le groupement des données sur lequel on va travailler.
3. Extraction des caractéristiques : la définition des caractéristiques via lesquels on analyse les journaux d'évènements.
4. Reconnaissance non supervisée : la technique pour la détection des activités chaotiques.

Dans le chapitre 5, on propose une approche qui peut maintenir l'amélioration des processus métiers tout au long de son cycle de vie BPM. L'idée est de définir des indicateurs d'amélioration relatifs à chaque phase du BPM, afin d'obtenir un processus métier qui s'améliore d'une phase à une autre à la base de cette relation (la phase : les indicateurs) :
Define Design metrics; Model : Analysis metrics; Execute : Execution metrics; Monitor : KPIs; et Optimize : Enhancement metrics. Tout en s'alignant aux techniques du process mining.
A chaque phase, on aura une propre sortie. Par exemple, aprés avoir appliqué les indicateurs d'amélioration de la phase Model un Optimized Model est généré et pour la phase Monitor, un Monitored Model est généré, etc.
Mais à tout moment, on risque d'avoir des interventions humaines pendant l'exécution de ces processus métiers, ce qui peut produire un processus métier non-structuré. Nous définissons La non-structuration par la complexité observée sur les journaux des évènements et du comportement qui peuvent être obtenu depuis les algorithmes de découverte. A ce stade, on peut déduire trois enjeux qui sont :
1) La difficulté de répondre en temps réel aux processus métiers non-structurés.
2) La difficulté de gérer la variabilité de ces processus métiers.
3) la difficulté de réagir dynamiquement selon les conditions métiers du processus métier non-structurés.

Dans le chapitre 6, on propose trois approches dont l'objectif est de répondre aux enjeux lies aux processus métiers non-structurés (Chapitre 6).
1. Operational Support Approach : le support opérationnel des processus métiers non-structurés via la simplification de ces derniers, la détection de l'anomalie, la prédiction des actions correctives et la recommandation de ces actions via les techniques du process mining.
2. Managing Vriability approach : Gérer la variabilité a chaque point de variation des process métiers non-structure, plus précisément les processus métiers auto-définis (les utilisateurs effectuent leurs recherches en fonction de leurs objectifs de diverses maniÃ¨res en produisant des comportements différents. Même les utilisateurs ayant le même objectif peuvent suivre des chemins différents et supporter différents sous-processus) via trois phase (la préparation des event logs, la nomination d'un algorithme de découverte adéquat au comportement du processus métier et la gestion la variabilité via les techniques du process mining et le concept des ontologies).
3. Dynamicity Approach : Etablir un processus métier Ad-hoc (processus caractérisés par une conception de flux de travail non définie) répondant aux changements d'environnement via deux grandes parties : Off-line (La préparation des données, La découverte du processus métier, Raffiner le processus métier par rapport les conditions métiers) et Online (la sélection dynamique du processus selon différentes conditions).
En effet, l'amélioration des processus métiers en traitant la complexité, la variabilité et la dynamicité. Dans le défi de la complexité, nous nous concentrons sur l'action de soutien opérationnel des processus métiers non-structurés. Pour le défi de la variabilité, nous proposons une approche permettant la gestion des points de décision, tout au long d'un processus métier auto-définis, en fonction des conditions spécifiques de l'entreprise. Enfin, la contribution de la dynamicité vise à construire un modèle de processus métier adaptable aux conditions métiers de l'entreprise.

Chapitre 7 conclut et résume nos contributions et développe les travaux futurs. Par conséquent, cette thèse s'articule autour de trois contributions principales :
1. Une approche permettant le pré-traitement des données d'exécution. Cette approche filtre les données chaotiques des journaux d'évènements.
2. Une approche qui maintient, dès la phase de conception, l'amélioration des processus métiers structurés.
3. Un ensemble d'approches qui traitent la non-structuration des processus métiers en traitant des problématiques liées à la complexité, la variabilité et la dynamicité des processus.
Toutes ces approches ont été systématiquement évaluées et appliquées dans le cadre de différentes études de cas d'illustration.

Contents

List of Figures

List of Tables

List of Acronyms

AI	**A**rtificial **I**ntelligence.
BP	**B**usiness **P**rocess.
BPI	**B**usiness **P**rocess **I**mprovement.
BPM	**B**usiness **P**rocess **M**anagement.
BPMN	**B**usiness **P**rocess **M**odelling and **N**otation.
CSV	**C**omma–**S**eparated **V**alues.
EL	**E**vent **L**ogs.
ETL	**E**xtract, **T**ransform, and **L**oad.
ERP	**E**nterprise **R**esource **P**lanning.
FZ	**F**uzzy **M**iner.
HM	**H**euristics **M**iner.
IF	**I**solation **F**orest.
IM	**I**nductive **M**iner.
IS	**I**nformation **S**ystem.
Iot	**I**nternet of **T**hings.
IT	**I**nformation **T**echnology.
KPI	**K**ey **P**erformance **I**ndicators.
ML	**M**achine **L**earning.
PAIS	**P**rocess–**A**ware **I**nformation **S**ystem
PCA	**P**rincipal **C**omponent **A**nalysis.
PM	**P**rocess **M**ining.
PN	**P**etri **N**ets.
QM	**Q**uality **M**etrics.
SBP	**S**tructured **B**usiness Process.
UBP	**U**nstructured **B**usiness **P**rocess.
XES	e**X**tensible **E**vent **S**tream.

Chapter 1

Introduction

This chapter presents the general context of this thesis, as well as the problem addressed, the main contributions and the thesis structure. It consists of six sections:
The first section (1.1) explains the general context of our research topic with brief definitions that will be extended in the next chapters. The second section (1.2) introduces the Business Process Improvement discipline. The third section (1.3) identifies main challenges that emerge this thesis content. The fourth section (1.4) explores shortly all contributions that are treated throughout this dissertation, related to the Business Process Improvement and process mining techniques. The fifth section (1.5) illustrates how this report is structured. The sixth section presents the publications resulted from this thesis (1.6).

1.1 General context

For some years, the usage of information systems has been rapidly growing in companies of all kinds and sizes. New systems are moving from supporting single functionalities towards a Business Process Management (BPM) orientation [1–3].
In this context, activities that companies are required to perform, to complete their own business, are becoming more complex and require the interaction of several persons and heterogeneous systems. A possible approach, to simplify the management of the business, is based on the division of operations in smaller "entities" and on the definition of the required interactions among them. The term "Business Process" (BP) refers to this set of activities and interactions. As example, we cite the process of handling of a loan application (service), the process of emergency (healthcare), the process of car manufacturing (production), etc. Indeed, several tasks or activities are executed in one instance of such a process. A process instance is commonly denoted as a case, i.e., the activities of the process that operate on the case. Each case of a process has a defined start point and end point.
The seminal articles on business re-engineering by Hammer and Champy [4] and Davenport [5] have established the focus on the processes of an organization in management practice. Furthermore, organizations should radically reorganize their work along their value-adding processes. A large body of work, both from industry and from academia, has been organized around the belief that performant processes are the foundation of any successful organization. The basic problem

that is being tackled is: How do organizations define and execute performant processes?
This problem has been addressed from various viewpoints and with several approaches. For example, management trends and strategies such as Business Process re-engineering [6], Lean management [7], Six sigma [8], research fields and methods such as workflow management, and adaptive case management. Moreover, many software systems for BP execution tools have been proposed. For example, Staffware, COSA, YAWL, Bizagi, Bonita, Camunda, jBPM, IBM Business Process Manager, Oracle BPM Suite, etc.
Beyond, BPM can be seen as the umbrella-term that encompasses all those methods that are concerned with the design, enactment, monitoring, and optimization of processes. The main objective of BPM is to align the process with the objectives of the organization. In this sense, each process must be configured, so that the results of the process lead to the achievement of the business goals. The BPM approach tends to provide more support for various forms of analysis (e.g., simulation) and management support (e.g., monitoring). In many cases, analysts are interested in a real system behaviour, which may be hidden from domain experts and system engineers.
To this purpose, most of the software that is used to define and to help companies in executing such processes, typically, leaves a trace of the executable activities. These traces called (Event log). Log consists of the name of the activity and the time the activity is executed; moreover, it is important to note that the traces are grouped in "instances" (can emerged a set of "cases"). Typically, it is necessary to handle several orders at the same time, and therefore the process is required to be concurrently instantiated several times. These instances are identified by a "case identifier" (or "instance id"), which is another field typically included in the log of the traces. Sometimes, especially small, and medium companies do not perform their work according to a formal and explicit BP; instead, they execute their activities with respect to an implicit sorting. Even if such model is not available, the presence of a log of activities is very frequent. So, the key idea is that a log can exist even if no process model is present.
Actually, the Business Process Improvement (BPI) paradigm can be implemented on recorded data of real process execution (traces). This is done by analysing this information and come up with real insights to BP improvement. The spread way of existing BPI methodologies put forward the complexity of their achievement the BP improvement goal. Moreover, they could be driven by many factors. Nonetheless, the common goal is to speed up generating an improved BP.

1.2 Business Process Improvement

BPI is the less drastic and disruptive, and more incremental approach. The main objective is to improve BPs by achieving the company business goals such as effectiveness, efficiency, etc. For the last hundred years, many approaches have been developed and deployed to obtain this strategic advantage.
First, we cite the Lean methodology [9]. It aims at increasing productivity by reducing operating costs through the elimination of all waste. Waste is everything that does not add value to the product or service. The pros of Lean are the reduction of costs and time needed for the production, logistic, and maintenance processes. On the other hand, Lean can be seen as a threat by employees. Indeed, frustration can arise when more work is needed to be done with less resources. Effective

human resource management is then a key in this process improvement.
Second, we cite the Six Sigma [10]. It can be seen as a business strategy to reduce costs while improving customer satisfaction. One of the key advantages of Six Sigma is its statistical base to analyse the whole cycle. A serious difficulty is the fact that it is so tightly defined. Work is carried out in teams and all members need to be committed to Six Sigma and trained in the process.
Third, we determine the fused Lean Six Sigma [11] that aims at maximizing performance by achieving the fastest rate of improvement in customer satisfaction, cost, quality, process speed, and flexibility as possible. While Lean is a philosophy carried out by the whole organisation, Six Sigma is a team-based program. The entire organisation is involved in the waste-elimination and, a culture of learning and continuous improvement is created. However, it is not expected that the whole organisation gets the same profound training that the Six Sigma teams are getting, since these trainings are quite expensive. The theory around Lean Six Sigma exists for less than two decades and the literature gives few downsides of its application.
Fourth, the Total Quality Management (TQM) [12] can be defined as the goal to accomplish total quality by involving everyone's daily commitment. Total quality means the achievement of satisfying customers' requirements continually at low costs. TQM does not have strict specifications or steps. It can be seen as a toolkit with a specific goal and the underlying philosophy of continuous improvement, focusing on processes, measuring performance, involving, empowering employees and several techniques.
Compared to Six Sigma, TQM is a sub-application of Sigma that includes all tools and philosophies of TQM. Six Sigma could be seen as the improved successor of TQM. TQM only succeeds when management participates and when group work is required enough while not having the prescriptive methodology for its implementation like Six Sigma does.
Fifth, we find the Lean MRO (Maintenance, Repair, and Overhaul) [13]. It handles the same philosophy as "regular" Lean, but with the distinction between production and service-oriented activities. According to the limited available literature on this topic, successful implementation is scarce.
Sixth, we note the existence of the Total Productive Maintenance (TPM) [14] methodology. Its main advantage is the reduction in delays and downtime by reducing breakdowns and equipment failures. However, TPM is difficult to implement since it requires the cultivation of ownership and awareness. Next to that, the training of personnel can lead to high initial costs.
Seventh, BP re-engineering (sometimes called BP redesign) [15] tends to radically redesign BPs. Processes are modelled to help the organization understand how these are acting. Modern information technology is used to achieve dramatic performance improvements. Breaking away from conventional wisdom and the constraints to enable a new process instead of automating an existing one.
Based on the approaches mentioned before, we can identify many internal and external reasons that promote for process improvement in companies. The first reason is that the performance of most processes tends to slowly go down over a time if it is not maintained. This means that even for supporting the current standards it needs to spend some amount of work for service maintaining. The second reason focuses on the case where the organization encountered difficulties in its process improvement, then it can be sure that it is actively engaged by its competitors. If even we

imagine the situation where the improvement is not engaged by any company or its competitors, there will be always a third person who will decide to take this market segment. The third reason is that modern consumers are becoming more and more demanding and even spoiled. The level of supply and quality is growing all the time. This leads to a rapid increase of expectations by consumers.

The methods mentioned here still considered as traditional ones that need data analyst efforts to bring significant improvement results, parallelly to the complexity of BPs in organisations. Therefore, we need a discipline that combines computational intelligence and data mining on the one hand, and business process modelling and analysis on the other hand.

In this regard, Process Mining (PM) [1], as a novel BPI method, has been defined to discover, monitor and improve BPs. This method analyses company's BPs process execution. Based on these data, PM can extract knowledge to improve BPs and upgrade their perspectives in terms of the execution time, the staff efforts (organisational), the BP control-flow and Data-flow (see Figure 1.2.1).

To conclude, all discussed BPI approaches have a statistical or analytical background, but process mining excels in its ability to automatically convert data into organized information. Compared with existing BPI methodologies, process mining had more computer capabilities to implement BP improvements results. However, there are several drawbacks that must be addressed.

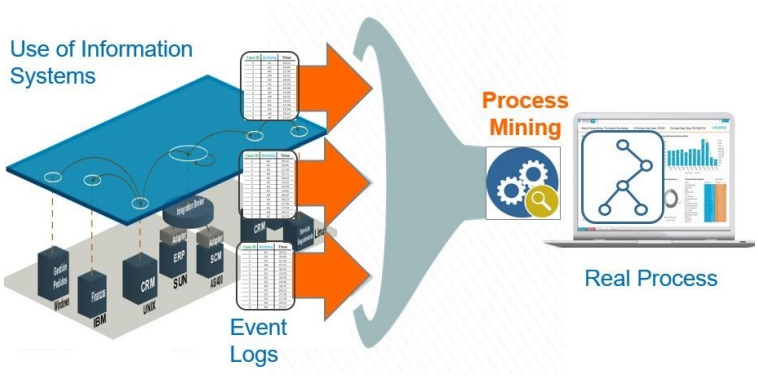


Figure 1.2.1: Process mining overview

1.3 Thesis problematic

Since most BPs are supported by at least one information system, the amount of data being stored about process executions is rapidly growing. This data might be recorded by a Process–Aware Information System (PAIS) [16]; software system that supports, manages and executes operational processes involving people, applications, and/or information sources based on process models to support). But also, information systems that are not process-aware record data about process execution. For example, an Enterprise Resource Planning (ERP) system might be used to support the process execution, or a purpose-made application might record data about the execution of process instances or cases in log files. Typically, the execution of a case results in a sequence of events (**execution trace**) being recorded. In general, such a log trace contains at least: the timestamps of activity executions (**events**) and the names or identifiers of the executed activity (**activity names**).

At this stage, it is important to refine the **quality of Event Logs** (EL) for more BP improvement. As more and more event data become available, the practical relevance of process mining is increasing. Indeed, a very large number of data provides unstructured BPs (from **structured BPs to unstructured** ones). Here, the challenge is **How to maintain BP simple structure ?**
However, during the BP execution , its simple structure (understandable and readable) can be transformed to a complex structure (incomprehensible and unreadable), due to human interventions. At this point, a new challenge is emerged: **How to improve BPs, when they are unstructured?**

1.3.1 Event log quality impact

The process mining discipline uses historical records of process executions, called event logs, to derive insights into business process behaviours and performance. Indeed, the quality of event data is a crucial element to achieve process mining objectives. It is widely observed that the poor event logs quality poses a significant challenge to the process mining project both in terms of choice of process mining treatment (algorithms) and in terms of the quality of the discovered process model. Moreover, the poor quality of data leads to poor quality of information.
Therefore, it is important to control the quality of event logs prior to conducting a process mining analysis, because it can be impacted by different deficiencies as: noise (data incorrectly logged), infrequent (infrequently executed) and incompleteness (like missing event logs). In this context, event logs still need a mature quality to provide more precise process models and specific BP improvement.

1.3.2 Structured Business Process Improvement

Adequately and on the monitor phase, improvement indicators focus on how a task is being performed by measuring performance and how individual goals are being achieved. These indicators should be measured by a ratio (generally represented by a number) which portrays the progress of the process as a whole or in part. Its serve to show if the organization is achieving the objectives set. These indicators are applied adequately in the last two phases of the BPM life-cycle (Monitor and Optimize).
In this context, a challenging problem is emerged for companies that are in need to improve their BPs during the whole BPM phases. Thus, it is a required to find an objectively way to measure BP improvement during the whole BPM life-cycle phases, in terms of what can be optimized and how can be improved with each passage from one phase to another.

1.3.3 Unstructured Business Process Improvement

Executing loosely structured processes generate unstructured behaviours. Thus, an Unstructured Business Process (UBP) still has more issues that are difficult to be analysed and hard to be understood due to its complexity and variability. Moreover, the need of instantiate response is clearly appeared in operational systems. Therefore, it is required to study related challenges that can be acquired during the transition from the structured BP to the unstructured one.

Mainly, the first challenge is how to support UBPs at runtime using process mining techniques. The second challenge is how to manage UBP variability taking into consideration variant conditions. The third challenge is how to adapt dynamically UBPs according to the company business rules and conditions.

1.4 Thesis contributions

This thesis is about BPs improvement using process mining techniques based on analysing recorded BP execution data. It provides algorithms and techniques to improve those processes and the systems that are used to put them into action. The aim here is to use logs to extract a BP model that must be coherent with recorded event logs. This model can then be used to represent the company business to detect and resolve deadlocks, bottlenecks, non-conformities, etc.
The main application of process mining is devoted to event logs. This later impact the BP quality and improvement. In this regard, the originality of this thesis contributions consists on proposing new approaches for BP improvement, using process mining techniques by taking into consideration recent event logs deficiencies and different BP structures. Indeed, we present for each BP type an appropriate approach, to overcome related challenges in terms of event logs quality and BP structures. For this purpose, we focus on three main goals.

a- Refining the quality of event logs in the objective of generating process models with high quality. We start from the quality of event logs and their impact on the quality of resulted BPs that can be generated from process discovery techniques. In this respect, we treat chaotic data, especially as activities, which can be occur arbitrarily in the process execution. Their existence degrades the quality of BPs and demineralises their improvement processes. This contribution is validated in [17].

b- Improving Structured BPs: There is a lack of common approach that addresses the improvement of BPs throughout the whole BPM life-cycle taking into consideration process mining techniques. To that end, we propose an integrated approach that defines improvement indicators for the whole BPM life-cycle. This approach can be used to improve BPs performance from the design phase to the optimize phase. This contribution is validated in [18, 19].

c- Improving Unstructured BPs: During the BP execution, a BP can be changed from simple (understandable and readable) to complex structure (incomprehensible and unreadable). This transition may be caused by human knowledge intervention, to correct violations that occur in a BP and that can prevent its daily progress. Here, a set of dramatical scenarios can be acquired and new challenges related to the BP improvement and structure can be defined. First, we propose an approach to treat the complexity of UBPs at runtime. Second, we define an approach that manages the variability challenge, in order to decide which path should be chosen during the BP execution, Third, we present an approach that tackles the dynamicity challenge to execute in a dynamic manner UBPs according to the company business rules and conditions. These approaches are respectively validated in these publications [20–22]

1.5 Thesis structure

This thesis is structured in two parts and 7 chapters (see Figure 1.5.1):

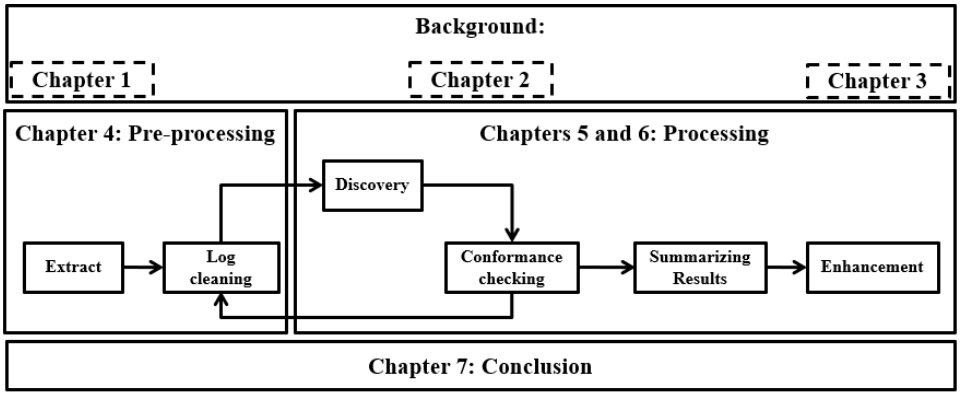


Figure 1.5.1: Thesis structure

- **Part I : State of the Art**:
  - **Chapter 2: Process mining**. This chapter provides the necessary definitions to understand this thesis contributions. Also, the chapter presents a comparative study between process mining and traditional BPIs methodologies.
  - **Chapter 3: BPI approaches using Process Mining**. This chapter reports the still encountered issues in the BPI discipline. Moreover, related works are discussed in the scope of process mining, event logs quality, structured and unstructured process approaches.
- **Part II : Contributions**:
  - **Chapter 4: Recognizing chaotic activities**. This chapter presents a new approach for recognising chaotic activities during the pre-processing phase to filter out deficiencies from event logs.
  - **Chapter 5: Structured Business Process Improvement**. This chapter presents an integrated approach for maintaining BP simplified structure and its improvement. This is done by defining improvement indicators throughout the BPM life-cycle.
  - **Chapter 6: Unstructured Business Process improvement**. This chapter discusses three BP improvement approaches that treat unstructured BPs using process mining techniques: Complexity, variability, and dynamicity.
- **Chapter 7: Conclusion**. This chapter concludes this thesis and illustrates future work.

1.6 Publications

The contributions presented in this thesis have resulted in the following publications:

Journals:

1. **Z. Lamghari**, R. Saidi, M. Radgui, and M.D. Rahmani. Mining self–defined business process in electronic administration [in–press]. International Journal of E-Services and Mobile Applications, 14(1), IGI Global, **2022** (dblp, Scopus).
2. **Z. Lamghari**, R. Saidi, M. Radgui, and M.D. Rahmani. Chaotic activities recognizing during the pre-processing event data phase [in–press]. International Journal of Business Intelligence and Data Mining, 18(2), Inderscience, **2022** (dblp, Scopus).
3. **Z. Lamghari**, M. Radgui, R. Saidi, and M.D. Rahmani. An operational support approach for mining unstructured business processes. Revista de Informatica Teorica e Aplicada, 28(1):23–38, **2021** (dblp, Scopus).
4. **Z. Lamghari**, M. Radgui, R. Saidi, and M.D. Rahmani. Defining business process improvement metrics based on bpm life cycle and process mining techniques. International Journal of Business Process Integration and Management, 9(2):107–133, Inderscience, **2019** (dblp, Scopus).

Proceedings:

1. **Z. Lamghari**, R. Saidi, M. Radgui, and M.D. Rahmani. Guided Process Discovery approach according to Business Process Types. The second International Conference on Advanced Technologies for Humanity–SCITEPRESS, **2020** (dblp, Scopus).
2. **Z. Lamghari**, R. Saidi, M. Radgui, and M.D. Rahmani. Leveraging dynamicity and process mining in Ad–hoc business process. Innovations in Smart Cities Applications. The Proceedings of the 5th International Conference on Smart City Applications–Lecture Notes in Network and Systems, vol. 183–Springer, **2020** (Scopus).
3. **Z. Lamghari**, M. Radgui, R. Saidi, and M.D Rahmani. A Framework Supporting Supply Chain Complexity and Confidentiality Using Process Mining and Auto Identification Technology in International Conference Europe Middle East & North Africa Information Systems and Technologies to Support Learning. Springer, **2019** (Scopus).
4. **Z. Lamghari**, M. Radgui, R. Saidi, and M.D. Rahmani. Predictive Process Monitoring related to the remaining time dimension: a value–driven framework. 1st International Conference on Smart Systems and Data Science–IEEE, **2019** (Scopus).
5. **Z. Lamghari**, M. Radgui, R. Saidi, and M.D. Rahmani. Passage challenges from data–intensive system to knowledge–intensive system related to process mining field. Proceedings of the ArabWIC 6th Annual International Conference Research Track–ACM, **2019** (dblp, Scopus).
6. **Z. Lamghari**, M. Radgui, R. Saidi, and M.D. Rahmani. Set of indicators for BPM life cycle improvement. The International Conference on Intelligent Systems and Computer Vision–IEEE, **2018** (Scopus).

Part I

State of The Art

Chapter 2: We introduce preliminaries such as basic notations and definitions used in this thesis.
Chapter 3: We introduce the still encountered issues related to business process imrpoevement and process mining.

Chapter 2

Process Mining Techniques

This chapter gives a general introduction to the Process Mining field, starting from the PM definition (Section 2.1). The chapter continues with illustrating PM categories (Section 2.2) and defining the very basic notion of event logs (Section 2.3). Also, this chapter discusses different process models representations (Section 2.4). Moreover, a list of process mining algorithms (Section 2.5) and tools 2.6) are given. The chapter finishes with a comparative study between traditional BPIs methodologies and the process mining methodology, showing process mining advantages (Section 2.7).

2.1 PM Definition

PM [1] is a relatively new field incorporating techniques for the discovery, monitoring, and enhancement of real processes by extracting knowledge from the information system event logs. Indeed, PM bridges two different fields: Process Science and Data Science (see Figure 2.1.1). Process Science is a broad area of process modelling, analysis, and optimization. It incorporates Stochastics (analysis of random processes, using Markov chains, queuing networks, and simulation), Optimization (finding the best possible process implementation by applying mathematical optimization techniques), Operations Management & Research (designing and controlling production processes from management and mathematical modelling perspectives), Business Process Management (methods and techniques for the modelling, execution, and enhancement of processes). Business Process Improvement (for instance, Six Sigma techniques and Business Process Re-engineering), Process Automation & Workflow Management (tools and methods for the processes execution, including routing and resource allocation), Formal Methods & Concurrency Theory (analysis of process behaviours, using Petri nets, finite state machines, and other formal models).

Data Science incorporates all aspects of data analysis and includes Statistics, Algorithms (providing efficient data processing), Data Mining (methods revealing unsuspected relationships in data sets), Machine Learning (techniques for giving computers capability to learn without being explicitly programmed), Predictive Analysis (methods predicting the future trends), Databases (techniques for storing data), Distributed Systems (infrastructure for data analysis), Visualization & Visual Analytics, Business Models & Marketing (techniques for turning data into real value),

Behavioural/Social Science (methods for the analysis of human behaviour), Privacy, Security, Law & Ethics (principles protecting individuals from "bad" data science practices).

Figure 2.1.1: PM as the bridge between Data Science and Process Science [1].

2.2 PM Categories

PM (see Figure 2.2.1) is defined by three categories [23]: (i) process discovery, (ii) conformance checking, and (iii) enhancement.

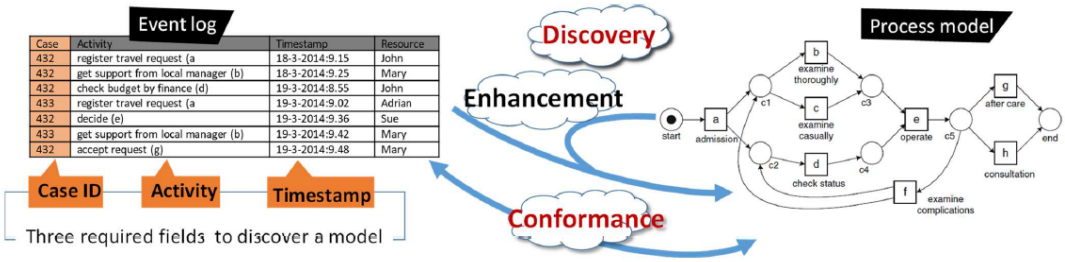


Figure 2.2.1: Process mining categories

a- Process discovery is a challenging task for many reasons. Often event logs are incomplete, i.e., only a fraction of possible behaviours is observed. The other issue is that it can be difficult to uncover the composition of choices, iterations, or parallel executions, represented in the form of a flat event log.

It describes process mining only in the offline setting, i.e., only finished process cases are analysed. Generally, process mining is not limited to the offline setting. It also entails methods such

as prediction and recommendation based on current process data in an online setting. In the scope of this thesis, we consider the offline and the online setting. The challenge here is the demand of high-quality data and the structured form of resulted processes.

b- Conformance checking methods find deviations from the expected behaviour. The expected behaviour can be represented in the form of a process model or an event log. One of the main challenges is the computational complexity. Typically, complicated process models and event logs lead to an exponential growth of possible alignments. The other challenge is to provide an intuitive visualization of alignments, helping analysts to reveal important discrepancies.
Beyond activity names and timestamps, an event log may contain additional information, such as performers, costs, IP addresses, or other domain specific data.

c- Enhancement techniques enrich process models with this information. Besides additional attributes taken from the event logs, these could also be results of conformance checking or performance analysis techniques. Model enhancement also considers an event log and a process model as inputs. This means, it is possible to improve an existing process model by looking in the past. Common aspects in model enhancement are time and cost. After discovering a process model from an event log, the discovered process model can be used to analyse for performance indicators, for example average throughput time and costs for improving or re–engineering the process. The bottleneck problem can be identified by analysing waiting times between activities. After identifying the cause of bottlenecks, the process model can be enhanced at the right places. Enhancing resource performance is one important aspect. A social network in a workplace can be constructed by process discovery. It can give an idea of work collaborations and balance workload to improve resource performance. There is no restricting procedure how a process model can be enhanced. It depends on what problems an organization discovers and how an organisation wants to improve.

2.3 Event Logs

PM typically assumes that BP execution data are stored as event logs. An event can be considered as the starting point of process mining. The event log structure is illustrated in Figure 2.3.1, where the process consists of cases or completed process instances. Each case is made of a sequence of events, called a trace. An event can have any kind of additional attributes (timestamps, cost, resource, etc.) depending on the organization purposes. These additional attributes are important for monitoring the BP improvement. For example, bottlenecks causes that can slow-down the process flow.
The event logs notation may depend on the information system treatment or purposes. However, the important point is the quality of these events that can heavily affect the process model representation and by necessity the main business of the organization. Therefore, event logs should be treated as first-class.

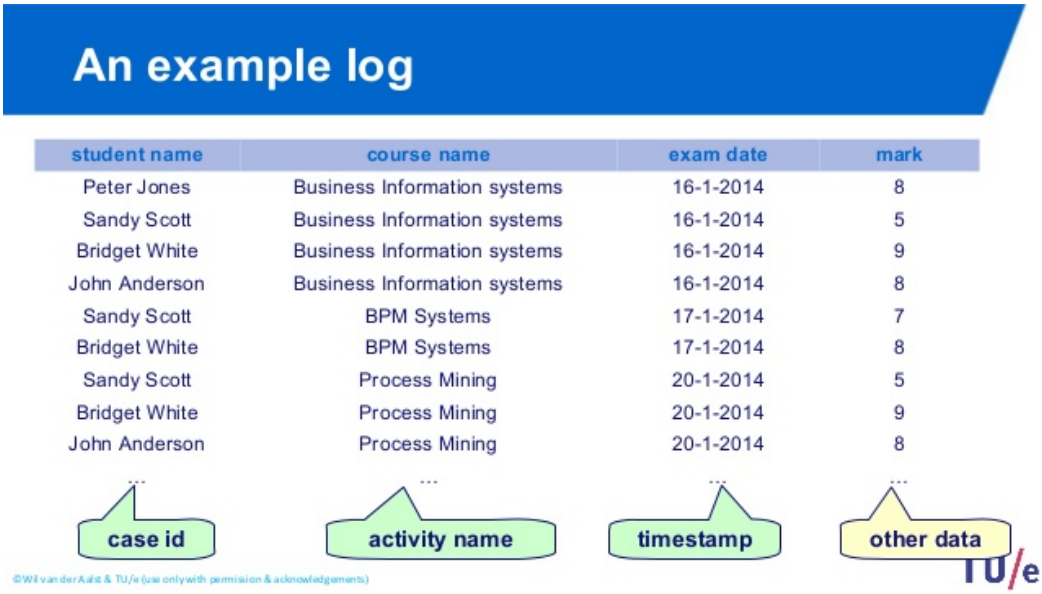


Figure 2.3.1: Common structure of event logs [1].

2.3.1 Notations 1 (Event, Trace and sequence or case)

From a mathematical standpoint, each event in an event log is assigned to an activity executed for a singular process instance (one trace). For each trace, all events belonging to that case are ordered in a chronological style (see Figure 2.3.1). In this regard, $A = \{a_1, a_2, \ldots, a_n\}$ denotes a finite set, where a_i, $i = 1, 2, \ldots, n$ is an activity of a case or sequence of length n. Thus, one case l can be expressed as $1 =< a_1, a_2, \ldots, a_n >$ and $<>$ denotes an empty sequence. The event logs L with n cases and r repetitions can be expressed as $\mathrm{L} = [\mathbf{l}_1^r, \mathbf{l}_2^r, \ldots, \mathbf{l}_n^r]$. For instance, the event logs $\mathrm{L} = [< a, \mathrm{b,c,d} >^{20}, < \mathrm{a,c,b,d} >^{15}]$ signifies $\mathbf{l}_1 =< a, \mathrm{b,c,d} >$ repeated 20 times, $\mathrm{l}_2 =< \mathrm{a,c,b,d} >$ repeated 15 times, etc.

2.3.2 Notations 2 (Noise, Infrequent, incomplete, and chaotic)

Event log can be affected by deficiencies, which are: noise, infrequent, incomplete, and chaotic activities: $L = [l_1^r, l_2^r, \ldots, l_n^r]$ is the event logs, where its composite sequences can be denoted as $l_0, l_1, \ldots, l_n$ and one sequence or case is expressed as $l = \langle a_1, \ldots, a_n \rangle$. In this context, we define:

Noise: entail outliers that were recorded due to errors (Incorrectly logged) $A = \{a_1/a_1, \neq a_i, \mathrm{i} = 2, \ldots, \mathrm{n}\}$. This point describes events of activities that were executed out of the normal order.

Infrequent: low-frequent behaviour. For instance, events recorded due to temporary workarounds and they are correctly logged. $L = \{\min_{l \in L} \text{ where } \min_{l \in L} \neq \max_{l \in L}\}$

Incompleteness: Partial traces. In this chapter, we mean by incompleteness the problem of missing events. For instance, $U = \{<> / \exists \mathrm{i} = 1, \ldots, \mathrm{n}, i \in l\}$. To do so, traces are not complete in term of execution, i.e., events must be executed in the normal process but are not observed in the recorded traces.

Chaotic activity: can happen anywhere in the process $C_h = \{a_1, a_2, \ldots, a_n / i = 1, \ldots, n\}$. For example: $L = [< a, b, c, d, C_h >, < a, b, c, C_h, d >, < a, b, C_h, c, d >, < a, C_h, b, c, d >, < C_h, a, b, c, d >]$. It takes any position from a_1 to a_n. C_h is a chaotic activity. These activities are executed arbitrarily in the process. They are correctly logged, but are still undesired, because they represent activities that are logged by the system even though they are not part of the main process flow. Therefore, chaotic activities are frequent and distinct from infrequent behaviour. We will learn how to filter chaotic activities in the following sections.

2.4 Process models

Process models are used to visualise, describe, prescribe, and explain the behaviour of processes of an organization for a wide range of objectives such as: communication among stakeholders, process improvement, process management, process automation, and process execution support [24]. Concrete examples are the comparison of the "as-is" and the "to-be" process, documentation for complying with regulatory requirements such as ISO 9001 [25], and the analysis of performance-related problems such as bottlenecks and inefficiencies.
Depending on the goal of the event logs analysis and on the analyst's personal taste, several ways of process visualisation can be used. The most common are Petri Net (see Figure Transition Systems 2.4.2), Petri Net (see Figure 2.4.3), BPMN (see Figure 2.4.4).

2.4.1 Process Modelling Notations

Many different process modelling notations have been proposed [26]. In this section, we focus on three notations (see Figure 2.4.1). Each notation has different properties, which make it applicable in a certain setting. We position them with regard to other notations and show that results can be transferred to standard notations used in practice (e. g., to BPMN).

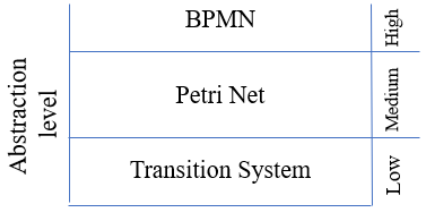


Figure 2.4.1: Process Modelling Notation

The main benefits of adopting a clear business model, through different levels of abstraction, are summarized in the following list:

- It is possible to increase the visibility of the activities, that allows the identification of problems (e.g., bottlenecks) and areas of potential optimization and improvement;
- Grouping the activities in "department" and grouping the persons in "roles", in order to better define duties, auditing and assessment activities.

1- Transition System:

The most basic process modelling notation is a transition system. A transition system consists of states and transitions.

Definition. A transition system [27] is a triplet TS = (S,A,T) where S is the set of states, $A \subseteq \mathscr{A}$ is the set of activities (often referred to actions) and $T \subseteq SxAxS$ is the set of transitions. $S_{start} \subseteq S$ is the set of initial states (sometimes referred to as "start" states) and $S_{end} \subseteq S$ is the set of final states (sometimes referred to as "accept" states). The sets S_{start} and S_{end} are defined implicitly. In principle, S can be infinite.

Figure 2.4.2 shows a transition system consisting of seven states. It models the handling of a request for compensation within an airline. The states are represented by black circles. There is one initial state labelled s1 and one final state labelled s7. Each state has a unique label. This label is merely an identifier and has no meaning. Transitions are represented by arcs. Each transition connects two states and is labelled with the name of an activity. Multiple arcs can bear the same label. For example, check ticket appears twice. However, the transition system depicted in figure 2.4.2 can be formalized as follows:
S = {s1, s2, s3, s4, s5, s6, s7}, S_{start} = {s1 }, S_{end} = {s7}, A = {register request, examine thoroughly, examine casually, check ticket, decide, reinitiate request, reject, request pay compensation}, and T = {(s1, register request, s2), (s2, examine casually, s3), (s2, examine thoroughly, s3), (s2, check ticket, s4), (s3, check ticket, s5), (s4, examine casually, s5), (s4, examine thoroughly, s5), (s5, decide, s6), (s6, reinitiate request, s2), (s6, pay compensation, s7), (s6, reject request, s7)}.
Given a transition system one can reason about its behaviour. The transition starts in one of the initial states. Any path in the graph starting in such a state corresponds to a possible execution sequence. For example, the path register request, examine casually, check ticket in figure 2.4.2 is an example of an execution sequence starting in state s1 and ending in s5. There are infinitely many execution sequences for this transition system. A path terminates successfully if it ends in one of the final states.
A path deadlocks if it reaches a non–final state without any outgoing transitions. Note that the absence of deadlocks does not guarantee successful termination. The transition system may live lock, i.e., some transitions are still enabled but it is impossible to reach one of the final states.

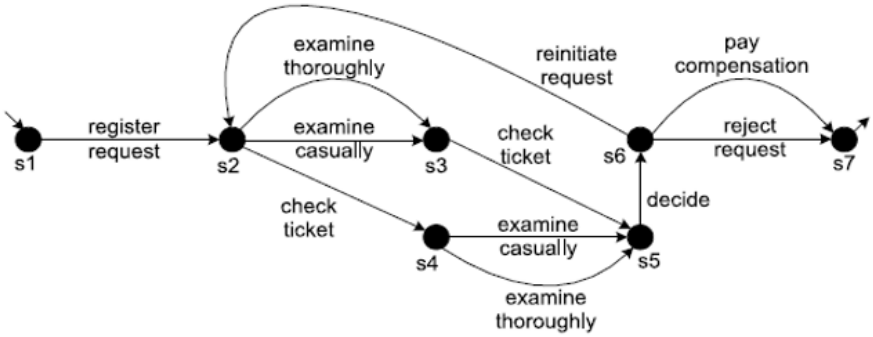


Figure 2.4.2: A transition system having one initial state and one final state

2- Petri Net:

Petri Nets [28] constitute a graphical language for the representation of a process. In particular, a Petri Net is a bipartite graph, where two types of nodes can be defined: transitions and places. Typically, transitions represent activities that can be executed, and places represent states (intermediate or final) that the process can reach. Edges, always directed, must connect a place and a transition, so an edge is not allowed to connect two places or two transitions. Each place can contain a certain number of tokens and the distribution of the tokens on the network is called "marking". Petri Nets have been studied from many points of view: from their clear semantic to a certain number of possible extensions (such as time, colon, etc).

Definition. A Petri Net is a tuple (P,T,F) where: P is a finite set of places; T is a finite set of transitions, such that P T = Æ, and $F \subseteq (PxT)$ [(TxP) is a set of directed arcs, called flow relation.

The "dynamic semantic" of a Petri Net is based on the "firing rule": a transition can fire if all its "input places" (places with edges entering into the transition) contain at least one token. The firing of a transition generates one token for all its "output places" (places with edges exiting from the transition). The distribution of tokens among the places of a net, at a certain time, is called "marking". With this semantic, it is possible to model many different behaviours, for example, in figure 2.4.3, a small Petri Net is shown; circles represent places, squares represent transitions, where three basic templates are proposed. The sequence template describes the causal dependency between two activities (in the Figure example, activity B requires the execution of A); the AND template represents the concurrent branching of two or more flows (in the Figure example, once A is terminated, B and C can start, in no specific order and concurrently); the XOR template defines the mutual exclusion of two or more flows (in the Figure example, once A is terminated, only B or C can start).

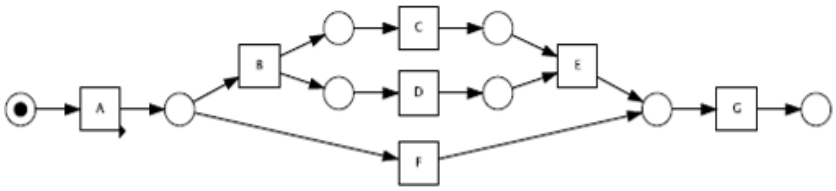


Figure 2.4.3: Petri net example

3- BPMN:

Definition: BPMN (Business Process Modelling and Notation) [29] is the result of an agreement among multiple software vendors, that agreed on the standardization of a single notation. For this reason, it is used now in many real cases. BPMN provides a graphical notation to describe business processes, which is both intuitive and powerful (it is able to represent complex process structure). It is possible to map a BPMN diagram to an execution language, BPEL (Business Process Execution Language).

The main components of a BPMN diagram, presented in figure 2.4.4, are:

Events: defined as "something" that "happens" during the course of a process; typically, they have a cause (trigger) and an impact (result). Each event is represented with a circle (containing an icon, to specify some details), as in Figure 2.4.4(d). There are three types of events: start (single narrow border), intermediate (single thick border) and end (double narrow border).

Activities: this is the generic term that identifies the work done by a company. In the graphical representation they are identified as rounded rectangles. There are few types of activity like tasks (a single unit of work, Figure 2.4.4(a)) and subprocesses (used to hide different levels of abstraction of the work, Figure 2.4.4(e)).

Gateway: structure used to control the divergences and convergences of the flow of the process (fork, merge and join). An internal marker identifies the type of gateway, like "exclusive" (Figure 2.4.4(b), on the left),"inclusive" and "parallel" (Figure 2.4.4(b), on the right). Sequence and message flows and associations: connectors between components of the graph. A sequence flow (Figure 2.4.4(c), top) is used to indicate the order of the activities. Message flow (Figure 2.4.4(c), bottom) shows the flow of the messages (as they are prepared, sent and received) between participants. Associations (Figure 2.4.4(c), middle) are used to connect artifacts with other elements of the graph.

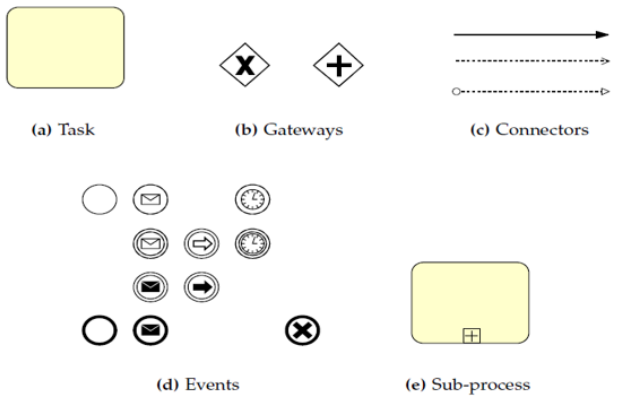


Figure 2.4.4: Example of some components of BPMN

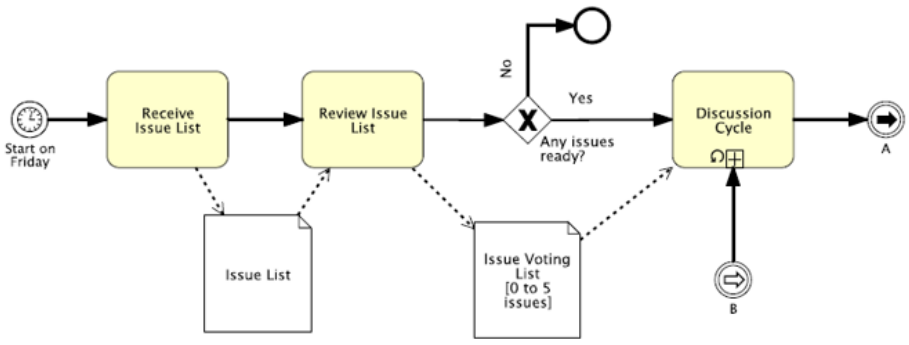


Figure 2.4.5: A simple process fragment, expressed as BPMN diagram

Beyond the components just described, there are also other entities that can appear in a BPMN diagram, such as artifacts (e.g., annotations, data objects) and swimlanes.
Figure 2.4.5 proposes a simple process fragment. It starts on Friday, executes two activities (in the figure, "Receive Issue List" and then "Review Issue List") and then checks if a condition is satisfied ("Any issues ready"); if this is the case, a discussion can take place a certain number of times ("Discussion Cycle" sub process), otherwise the process is terminated (and the "End event" is reached, marked as a circle with the bold border). There are, moreover, intermediate events (marked with the double border): the one named A is a "throw event" (if it is fired, the flow continues to the intermediate catch event, named A, somewhere in the process but not represented in this figure); the B is a "catch event" (it waits until a throw events fires its execution).

2.4.2 Business Process Structures

A traditional workflow language aims at constructing a well-defined and highly automated. As a result, processes become more structured. Structured process is referred to rigorously defined process, less complex and with high repetition frequency.
The definition of a structured process as given by Devonport is as follows:

Structured Process/Lasagna (see part **a** of Figure 2.4.6) is defined as a specific ordering of work activities across time and place, with a beginning, an end, and clearly identified inputs and outputs: a structure for action [30].

Therefore, structured processes comprehend those processes whose activities execution consistently follows a predefined process model references [31]. Since a formal representation of these processes can be easily described prior to their execution, tightly framed processes are characterized as fully predictable and repetitive and after their design-time description, they can be repeatedly instantiated at runtime. Examples of this category are production and administrative processes and as well as bank transactions that are executed in an exact sequence to comply with legal norms.

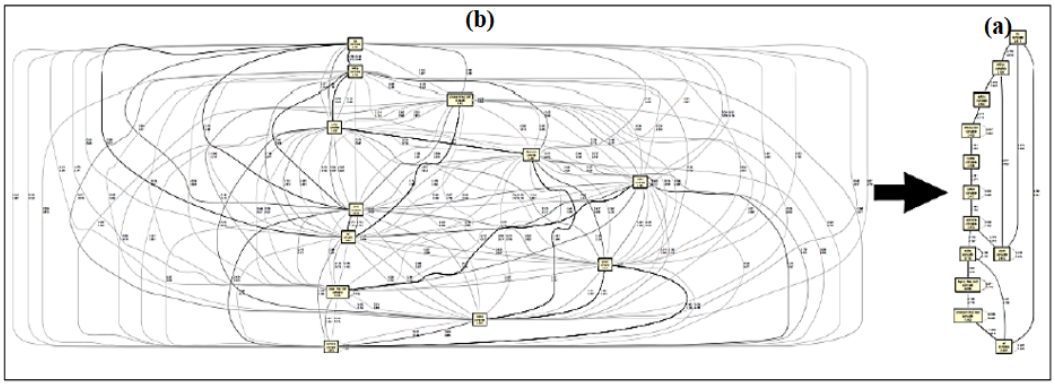


Figure 2.4.6: Unstructured process model Vs Structured process model [2].

Unstructured process/spaghetti: every instance of the process can be different from another based on the environment, the content and the skills of the people involved. These are always

human processes. These processes may have a framework or guideline driving the process but only as a recommendation [32].

Therefore, unstructured process (see part **b** of Figure 2.4.6) is partially or totally not predefined, adaptable, content–driven and knowledge worker involved. We can then define their characteristics as follows:

- **Diversity**: processes that can generate a set of execution cases that are structured very differently.
- **Knowledge–Intensive**: Decisions in an unstructured process that are based on a lot of information, which may be provided from different resources.
- **Uncertainty**: There is no single answer or end–result in an unstructured process.
- **Flexibility**: There is no single way to complete an unstructured process. Each next step depends on the previous one and could be completely different every time we run the process.

Moreover, figure 2.4.6 illustrates the transformation of a spaghetti process to become more comprehensible as a simplified process model. The operation simplifies process models by keeping high frequent behaviours and filtering out low frequent behaviours. Indeed, executing loosely structured processes generates unstructured behaviours [33]. After mining an unstructured log, a spaghetti-like process can be revealed. "Spaghetti processes" is a metaphor of unstructured ones. It cannot be assumed that a spaghetti–like process is wrong or that it has a problem caused by process discovery algorithms. It rather means a process model accurately reflects reality. However, spaghetti processes still have issues that are difficult to be analysed and hard to understand due to its complexity. In this context, it is a very interesting challenge to simplify an unstructured process into a more structured one.

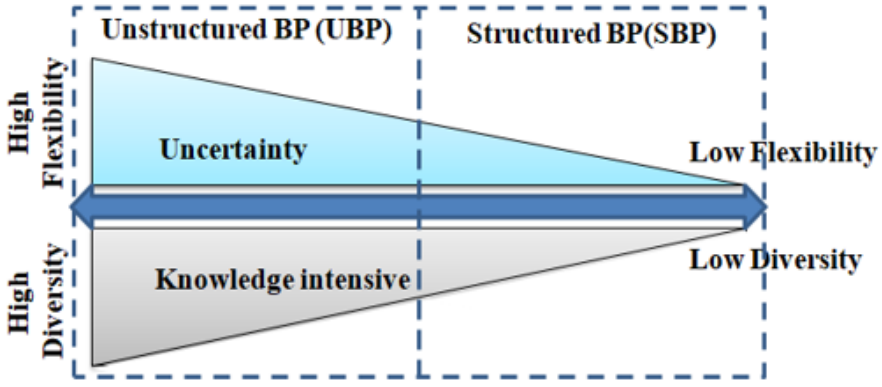


Figure 2.4.7: Summary of SBP & UBP characteristics [2].

The BP characteristics (see Figure 2.4.7) can help to determine the system within we will implement our business processes. These systems can be defined as systems with intensive data or with

intensive knowledge (see Figure 2.4.7). Here, we illustrate the passage between data–intensive system and knowledge–intensive system. Obviously, unstructured processes are composed of a set of structured processes. Indeed, process mining concept consists on treating well-structured processes. For this reason, PM still meaningfully usable for unstructured processes. Consequently, PM can support system with intensive knowledge, like Enterprise Content Management (ECM) and Adaptive Case Management (ACM) system, and systems with intensive data like BPM.

2.5 PM Algorithms

At the heart of a PM discovery technique lays process mining algorithms. These algorithms, often embedded in the process mining software, translate the data from event logs into readable models. Several algorithms are available, each having its own properties concerning the form of input, conversion of data, and form of output. One must pick the right algorithm for a dataset for the right goal and right way of visualisation. Much has been written about the mining algorithms. Thus, a process discovery algorithm constructs a generic process model based on event logs. It is an abstracted and general representation of real event logs. Several discovery algorithms are described with basic representation of process models, like alpha algorithm. Other algorithms are representing different levels of abstraction combined with clustering and classification techniques, to model processes from unstructured and complex events. In this regard, we are inspired by [34] to list the following process discovery algorithms:

The first miner developed is the **Alpha–algorithm** [35]. This algorithm is based on eight simple mathematical definitions and visualises its models in the Petri Net modelling language. Because of its simplicity, it is popular among scholars, but it is unpractical in real-life, because of its difficulty in handling noise, infrequent/incomplete behaviour, and complex routing constructs. A second miner is the **Heuristic miner** (previously called Little Thumb), is better equipped to handle complex routing and it can abstract exceptional behaviour and noise, making it suitable for actual logs [36]. The third algorithm is **Fuzzy miner** that focusses on unstructured behaviour and large event logs. Its output is configurable to reach a desired level of abstraction, but can only be visualized in a fuzzy model [26]. Indeed, it deals with process complexity. It highlights significant information and hides less significant activities. In this sense, fuzzy miner simplifies unstructured processes. The simplification process aims at preserving significant behaviour, while less significant but highly correlated behaviours are aggregated into clusters, and less significant or less correlated behaviours are abstracted.

Other interesting algorithms are:

- **Inductive Miner (IM)**: treats events by grouping them into sub-logs. For each sub–logs, a sub-process is generated. Then, a combination between the resulted sub-processes are released to obtain the generic process model. In this respect, the IM algorithm produces sound models [37], i.e., less none-conformities detected, and it fits with the majority of present logs. Besides, it cannot identify complex and non-local process control patterns.

- **Genetic Miner (GM)**: randomly creates process models from logs. For each process, the

precision metric is calculated. Then, sound models are combined based on the mutation operation. The genetic algorithm improves models according to specific objective. The main limitation of this approach is their complexity in discovering and representing process models from real data sets [36].

- **State Based Regions (SBR)**: generates a Petri net from a Transitions System (ST) based on specific abstractions, such as: Set, Multi-Set, Sequence and other types of abstractions, in which each state of the ST can be represented by a complete or partial trace. This algorithm ensures the fitness metric, as well as the identification of complex control structures. On the other hand, SBR is unable to process incomplete and noisy logs [38].

- **Language Based Regions (LBR)**: The main idea of this algorithm is to find places based on the language process. All candidate places correspond to a language region. The discovered Petri net will be obtained by adding a place for each positive solution based on linear resolution. Each solution is represented as a triplet (A; B; C), where A is the set of inputs arcs, B is the set of outputs arcs and C is the number of tokens in the square. Indeed, the LBR algorithm uses properties derived from logs (causal relationships), to determine the final model by describing different places. Unfortunately, this algorithm is unable to process incomplete and noisy logs [39].

2.6 PM Tools

Currently, there is a wide range of research and commercial tools available in PM area: ProM [1], Disco [2], ARIS Process Performance Manager [3], QPR Process Mining [4], ProcessGold [5], Celonis [6], Minit [7], and myInvenio [8]. Indeed, several process mining tools are available. Choice can be based on a specific needed set of functionalities, supported data formats, but also costs. Finding the right tool can be difficult, since no comprehensive comparison exists.
Table 2.1 shows a list of process mining tools. Overall, three earnings models can be distinguished. First the common licencing structure where an organisation can buy the tool for a certain period or indefinitely. The tool can be sold in combination with or without support for implementation or analysis. This is the case with, for example, ARIS BPM (Process Performance Manager), Celonis Process Mining, and Disco. Another earnings model is offering process mining as a service (PMaaS). This is provided by Icris and Coney. And lastly, several open-source tools are available. Most famous example is ProM, but also Apromore is popular. Based on the number of academic publications on the topic, open-source process mining platform ProM seems to be the most popular. ProM is an extensible framework that runs on Java and obtains its functionality by a

[1] http://www.promtools.org/doku.php
[2] https://fluxicon.com/disco/
[3] http://www2.softwareag.com/in/products/aris_alfabet/bpa/aris_ppm/default.aspx
[4] https://www.qpr.com/solutions/process-mining
[5] http://www.processgold.com/en
[6] https://celonis.com/
[7] https://www.minit.io/
[8] https://www.my-invenio.com/process-mining-vision/

wide variety of plug-ins. Because it is an independent platform and is developed by process mining "creator" Will van der Aalst, it is popular among scholars performing applied research. There are over 1500 free plug-ins available, each with different functionalities and options [1]. For example, the use of different miners (heuristic, alpha-algorithm, and fuzzy), sorts of output (Petri Net and BPMN), and types of analysis (process discovery, dotted chart, and social networks). However, the academic character makes it difficult to use. Manuals and instructions are missing, and support can only be found in its community of volunteers. It seems that only few commercial organizations use this tool and even if they do so, they mostly use it to learn the concept of process mining before buying more user-friendly tool.

Table 2.1: List of process mining developers till January 2020

Start	Tool name	Tool Developer	Country
2005	ProM	The Process Mining Group	The Netherlands
2007	ARIS Process Performance Manager	Software AG	Germany
2007	Interstage Automated Process Discovery	Fujitsu, Ltd.	Japan
2008	StereoLOGIC Discovery Analyst	StereoLOGIC	The United States
2009	The Process Mining Factory	Icris	The Netherlands
2010	Apromore	The Apromore Initiative	Australia
2011	Celonis Process Mining	Celonis GmbH	Germany
2011	Perceptive Process Mining	Perceptive Process Mining	The United States
2012	Disco	Fluxicon	The Netherlands
2012	QPR Process Analyzer	QPR	Finland
2012	Process Mining Solution	Coney	The Netherlands
2013	SNP Business Process Analysis	SNP Schneider Neureither & Partner AG	Germany
2013	minit	Gradient ECM	The United States
2009	myInvenio	Cognitive Technology Ltd	Malta
2015	XMAnalyzer	XMPro	The United States
2016	Lana	Lana Labs GmbH	Germany
2017	ProcessGold	ProcessGold International B.V.	The Netherlands
2018	Kofax insights	Kofax INC.	The United States
2018	Appnomic OpsOne	Appnomic self-healing enterprise	India
2019	MPM process mining	GmbH	The Netherlands

A more user-friendly process mining tool package is Disco, Developed by Fluxicon. Disco

lacks some functionalities when compared to ProM, but distinguishes itself with a fast, well-documented, and clear interface. Very limited knowledge on process mining is required to perform an analysis. But the lack of real-time connections to databases makes it less useful for large companies. It seems logical that Fluxicon focusses on small and medium-sized enterprises.
A third tool package, used by MoD, is ARIS Process Performance Manager. This package contains three components: the administrator's section, the business analyst's section, and the dashboard. The section for the administrator is used to load the data. This can be a single file, but also a connection to a database. A good example is SAP. With the right connector, the SAP databases can be periodically and automatically loaded into the ARIS databases. Initializing this connector will cost some effort but can be a good investment. The business analyst's section of the tool is used for in-depth analysis. With the use of filters, selections of the dataset can be made. Analysts can use several techniques and models to answer their process related questions. The steps of this process (the query) can be saved, so when the data are refreshed, the analysis can be updated. The last section, the dashboard, is meant for tracking the organisation's process performance. The analyst can develop certain queries. For example, the average lead time of preventive maintenance of a specific weapon system, and this can be loaded into the process-centric dashboard as a Key Performance Indicator (KPI). The dashboard periodically collects recent data from the ERP and the process owner can follow the progression. Based on the average process and on its excesses, the process owner can decide to intervene. Also, the [40] developed a model to compare ProM, Disco, and Celonis on fifteen characteristics. The model was updated to the lasted tool versions and Celonis was replaced for ARIS PPM.

2.7 Summary

A basic discussion of several traditional BPI methodologies has already been given in chapter 1 (section 1.2). Based on this research results, the overview can be supplemented with the information mentioned in Table 2.2 (Table 2.3 is the exhaustive list), in order to compare process mining with these traditional methodologies. On one hand, they find their origin in Lean and while throughout the years many different BPI approaches arose, the boundaries between these BPI approaches remain vague. In this regard, many approaches have been combined to form new BPI approaches and users often interchange the terms. As a science, this makes it difficult to investigate their characteristics. An overview is presented to order eight widely used BPI approaches on nine properties.
This overview contributes to the knowledge on BPI. However, this is only the tip of the iceberg. Scholars and management consultants frequently come up with new BPI approaches and combine them till distinction is long gone. For example, next to Lean Six Sigma and Lean MRO the research shows us that the variations Lean Start-up, Lean Manufacturing, Lean Management, Lean Thinking, Lean Enterprise, and Lean Maintenance also are being used. These variations could be completely new methodologies, slightly adjusted methodologies, or identical to Lean as described in this chapter. On the other hand, all discussed BPI approaches have a statistical or analytical background, but process mining excels in its ability to automatically convert data into organized information. It is hypothesized that PM is a mature BPI approach. In its short history, process

mining has made an impressive development. Every year, more process mining tool is developed, papers on the topic are published, courses in process mining are given, and case studies are conducted [41]. Since process mining is a data driven activity, and with data storage becoming cheaper and cheaper and initiatives like the Internet of Things (IoT) boosting data production, new possibilities do arise. Mechanics can, for example, enter their activity data in the ERP with wireless tablets, giving real time analysis possibilities. Combining this with Artificial Intelligence (AI) and Machine Learning (ML) can offer even bigger opportunities. AI can find deviating process instances and even suggest improvements without human intervention. PM provides data on all activities of a process: its throughput, lead times, its delays, etc. These data can be used for building an accurate model in simulation tools. By reasoning, but also by trial-and-error, elements in the simulation can be changed till the model cannot be improved anymore. The changes can then be applied in the real world. Integrating process mining tools with simulation tool can create significant opportunities, and from the managerial standpoint, PM accumulates teams experiments to produce more significant results.

Table 2.2: Process mining compared to traditional BPI methodologies.

Concepts	Process Mining (Mature version)
Origin	The rise of big data and accessibility of computing power
First mentioned	2015
Align	Align de facto with de jure process models
Process view	Discovering, conforming, and enhancing business processes
Involvement	Multidisciplinary team
Methodology	Plan, extract, process data, mine and analyse event data, evaluation, and process improvement & support in a team
Primary Effects	Gain quantitative and factual knowledge about processes
Secondary Effects	Improvements can be monitored and verified
Criticism	Demanding high quality data and structured processes

To conclude, all discussed BPIs have a statistical or analytical background, but process mining excels in its ability to automatically convert data into organized information. Compared with existing BPI methodologies, process mining had more computer capabilities to implement BP improvements results. However, there are several drawbacks that must be addressed. Indeed, these drawbacks will be discussed in the following chapter. They mainly concerns event logs quality and business process structures.

Table 2.3 Process mining compared to traditional BPIs (exhaustive list)

Concepts	Lean	Six Sigma	Lean Six Sigma	TQM	Lean MRO	TPM	BPR	BPI	Process Mining
Origin	The quality evolution in Japan and Toyota	The quality evolution in the United States and Motorola	The combination of Lean and Six Sigma	The United States Navy	Aviation and aerospace industries	The quality evolution in Japan and Toyota	The increased competition in the United States	The increased competition in the United States	The rise of big data and accessibility of computing power
Year	1988	1986	2001	1985	2005	1971	1990	1990	2015
Theory	Remove waste	Remove variation	Remove waste and remove variation	Focus on customers	Remove waste and improve scheduling	Auto maintenance and employee commitment	Use modern information technology	Understand and streamlining processes	Align real time process with business process rules and conditions
Process view	Improve flow in processes	Reduce variation to reduce defects and improve processes	Achieve fastest rate of improvement, cost, quality, process, speed and flexibility	Improve and uniform processes	Improve and speed up flow in processes	Develop preventive maintenance programme for the life cycle of the equipment	Radically redesign business processes with modern information technology	Incremental process improvement	Discovering, conforming, and enhancing business processes

Concepts	Lean	Six Sigma	Lean Six Sigma	TQM	Lean MRO	TPM	BPR	BPI	Process Mining
Involvement	Whole organisation	Project team	Separate project teams	Whole organisation	MRO department	Small maintenance teams	Whole organisation	Project team	Multi disciplinary team
Method	Understanding customer value, value stream analysis, 7 deadly wastes, 5S, flow, pull, perfection, just-in-time	Define, measure, analyse, improve (or design), control (or verify)	Define, measure, analyse, improve, control, redesign, single unit flow, total productive maintenance	Plan, do, study, act	Value stream analysis, 7 deadly wastes, perfection, just-in-time, theory of constraints Creating new attitude toward maintenance based on six elements	Redesign the organisation based on six principals of reengineering	Organizing for improvement, understanding the process, streamlining, measure and control, continuous	Extract, process, mine, and analyse event data	Evaluation and improvement in a team
Primary effects	Reduce lead time	Save money	Add value	Increase customer satisfaction	Reduction	Reduces breakdowns, accidents, and equipment failures	Performance improvement	Reduces waste and bureaucracy	Gain quantitative and factual knowledge about processes

Concepts	Lean	Six Sigma	Lean Six Sigma	TQM	Lean MRO	TPM	BPR	BPI	Process Mining
Secondary effects	Reduces inventory, increases productivity and customer satisfaction	Improves financial performance, and product quality	Reduces inventory, increase productivity, customer satisfaction, improves financial, and product quality	Achieves customer loyalty and improves performance	Maximize asset availability and minimizing downtime	Reduces delays, downtime, and increase employee commitment	—	Simplified operations and increased (internal and external) customer satisfaction	Improvement can be monitored and verified
Criticism	Reduces organisations flexibility, causes congestion in the supply chain	Does not involve everybody, does not improve customer satisfaction, does not have a system view	Not mentioned	Not tangible improvements, resource-demanding, unclear notion	Documented successful implementations are scares	No strict activities	A strict focus on technology and efficiency while not mentioning the soft side	Outdated and inflexible as a methodology	Demanding high quality data and structured processes

Chapter 3

BP Improvement using Process Mining

This chapter illustrates the still encountered issues related to the Process Mining field in terms of event logs quality (Section 3.2) and BP structures. Here, we focus on BPM life-cycle phases (Section 3.3), and different challenges that can be emerged during unstructured BPs generation, such as event log complexity, process variability, and dynamicity (Section 3.4). Moreover, this chapter aims at positioning our contributions throughout to existing process mining techniques (Section 3.5). Therefore, all sections are released as survey studies.

3.1 Background

Process mining improves business operations based on different methodologies that are developed chronologically. The first methodology that addresses process mining was proposed by Bozkaya, Gabriels, & van der Werf [42]. Their Process Diagnostics Method (PDM) consists of six steps: log preparation, log inspection, control flow analysis, performance analysis, role analysis, and transferring the results. Each step is accompanied by understandable instructions and aims to give a broad process overview within a brief period without any prior and domain specific knowledge required. Also, the authors claim that existing process mining methodologies do not perform well in the complex and Ad-hoc healthcare environment. They continued to build a sequence clustering analysis between the earlier proposed log inspection and control-flow analysis. They successfully analysed the care-flows of emergency patients in a Portuguese hospital.
The second methodology [36] is designed for "lasagne processes" (structured processes), containing five stages: plan and justify; extract; create control-flow model and connect event log; create integrated process model; and operational support. This L* life–cycle model can also be applied on "spaghetti processes" (unstructured processes), but only after another sub-process of simplification. Since van der Aalst mentions significant respect regarding the topic, the L* life-cycle model is frequently mentioned in the literature. To better address lasagne processes, van Eck, Leemans, & van der Aalst in [26] propose PM. Their six-step methodology (consisting of planning, extraction, data processing, mining & analysis, evaluation, and process improvement and support) is a more project-based approach, deliberately starting with setting a good research question. Unfortunately, the process improvement and support steps seem to require more elaboration. The authors suggest that other BPIs can be used to carry out the actual implementation of improvements, but

the integration of PM with other BPIs is hardly mentioned. The third methodology [43] addressed the monitoring view, where authors propose significant steps to refine UBP. In this context, it consists of two major phases: Pre-processing and Processing. Depending on the process subject, they extract data from Repository. Once the data has been extracted, it is possible to filter them from deficiencies as the log cleaning step. By doing so, the discovery task can be applied. Next, the conformance checking step can begin. This step aims to evaluate how well this discovered process model corresponds to reality. This gives input for changing parameters in the main process, and requires going back to the data cleaning step, to re-apply the discovery technique. After evaluation, all obtained results can be used to provide inputs for BP improvement.
In this regard, **the pre-processing** phase works directly on cleaning event logs, while the processing phase focuses on analysing events, generating and enhancing process models. In this context, process mining envisages different challenges in terms of the pre-processing event data phase and the processing phase. From the **pre-processing** phase, event logs still need a performant filtering technique to provide high event log quality and more precise process models. From the processing phase, the problematic of BP structure needs a focus on different challenges of **complexity**, **variability** and **dynamicty**. Here, every author on the topic emphasises the fact that no methodology is perfect and that they act as high-level guidelines instead of a clear step-by-step details. Therefore, we discuss the still encountered issues related to the pre-processing and the processing phases, to position our thesis contributions related to process mining.

3.2 Pre-processing and event logs quality

The quality of event data is a crucial element to achieve process mining objectives. It can be impacted by different deficiencies as: **noise** (data incorrectly logged), **infrequent** (infrequently executed) and **incompleteness** (missing event logs). In the literature, we find many approaches dealing with event data deficiencies, which are nominated as filtering techniques in the pre-processing task of process mining. We classify them into different categories: i) Trace and activity filtering techniques [37, 38] with impacts on event logs quality [39], ii) Process mining algorithms performances related to filtering techniques. iii) Process discovery techniques with integrated filtering mechanisms [30, 44]. The aforementioned categories can be grouped as: 1- early pre-processing event logs that can be applied from the extraction phase. 2- later pre-processing operation that can be applied within the discovery technique.

3.2.1 Early pre-processing event data

In [45], authors propose a mechanism for cleaning event data with respect to its expected properties, using domain-specific constraints based on the knowledge that typically complements processes for the purpose to enhance the quality of a mined process model. Also, the [46] introduces new methods for noise filtering problem. They used two unsupervised noise filtering algorithms for this purpose. In addition, [47] develops a technique for revealing out noisy logs from event data. The technique builds a classifier on a subset of the log data and then starting by removing noisy traces based on the classifier rules. Further, [48] presents an approach for cleaning

event logs based on a graphical repair concept. The method repairs using a recommended process model and event log with inconsistent labels (labels which do not correspond to the labels of the recommended model). Nevertheless, this approach necessitates the availability of a selected process model which is usually unavailable. As well as [49] recommends an integrated tool named Log to Model Explorer that allows users; in an interactive and iterative way, exploring and pre-processing a log by clustering, filtering infrequent events and renaming event label. In [50], the authors present examples of event log imperfection patterns that reduce the trustworthiness of process mining results. Moreover, the study of [51] illustrates regular data quality problems in event data and outlines approaches to deal with the studied quality problems.
Besides, one study has proposed four techniques for filtering out chaotic activities from event data. The approach [46] aims at generating more precise process models, i.e., process model with high quality, based on supervised learning techniques from the data mining field. The supervised learning requires a set of labelled training data, normal activities, or chaotic ones, to learn from normal or anomalous behaviours in a predictive model representation. In the case of new action, the predictive model is applied to classify cases. Examples of such approaches, we cite support vector machines [52] and Bayesian networks [53]. The problem with the supervised approach is that the system can only distinguish deficiencies that are conform to its predictive model due to the lack of deficiencies definition, instances or examples. A truly novel anomaly event may escape detection, as the training data contains no archetype for it, while unsupervised anomaly detection approach does not require labelled training data. It operates on the assumption of normal data instances, which are the most common within the data set. There are other determining factors, such as clustering where normal data will congregate in large, dense clusters and anomalies that are occupied isolated positions within the data set. Therefore, unsupervised mode can be better at recognizing novel events. Also, past work has also explored unsupervised learning for anomaly user behaviour detection [47, 54, 55].

3.2.2 Later pre-processing event logs

In this category, we find different studies dealing with the performance of process mining algorithms, using statistics and statics oriented probabilistic solutions. First, authors in [56] present a new approach with a statistical trace-based sampling method to diminish the time for discovering process models and for providing memory footprint. In the same point of view, [47] treats in a statistical manner event log and applies the conformance checking algorithm on the sampled data. Second, this paper [48] illustrates an example of sampling method related to the Heuristic miner algorithm. Third, the study in [57] focuses on the concept of random biasing that aims at parameter the size of the sampled data designated to the process discovery task. Fourth, [49] combines between the pre-processing event logs phase and the conformance checking techniques, in order to approximate the alignment value. There is also a research presented in [37] that proposes an approach that focuses on applying the conformance checking on the majority of event data behaviours, to provide a suitable approximation of it. Moreover, [56] uses traditional pre-processing methods, to obtain an approximation in performance analysis results. Last, [58] aims at modifying infrequent behaviour to generalize them in each process instance instead of removing infrequent

behaviours. We observe that the aforementioned approaches, in this category, measure the quality of discovered models by only using F1-Measure, i.e., the combination of precision and fitness [59], without considering chaotic activities.
Tremendous algorithms, e.g., [60, 61], were designed to treat infrequent behaviour in event data and improve the quality of discovered process models. But these filtering methods are adapted to the internal working of the corresponding algorithms and they are not generalized on event log as a pre-processing phase. They typically focus on a specific type of behaviour, e.g., incompleteness. Likewise, [37] suggests the use of an anomaly free automation method to remove infrequent behaviour. Furthermore, it is also recommended to filter out process instances which contain infrequent behaviour from event data using probabilistic [62] and sequence mining [58] approaches. Moreover, [63] uses clustering techniques on large real event logs considering a sample of traces. It proposes a prototype selection approach based on a clustering algorithm. It is used to improve the F1-Measure and the simplicity metric of discovered process models [64] .

3.2.3 Synthesis

Table 3.1 illustrates the pre-processing techniques used in the literature to filter out deficiencies from event data.

Table 3.1: Pre–processing event logs techniques summary

Approaches	**Deficiencies**			
	Noise	**Infrequent**	**Incomplete**	**Chaotic**
Conforti et al., 2017 [37]	**x**	**x**	**x**	
Van der Aalst, 2018 [62]		**x**		
Sani et al., 2018 [58]		**x**		
Lu et al., 2016 [63]	**x**			
Sani et al., 2019 [65]	**x**			
Tax et al., 2019 [3]				**x**
Cook et al., 2006 [56]	**x**			
Eberle et al., 2007 [66]	**x**			
Berti et al., 2019 [57]	**x**			
Sani et al., 2019b [64]		**x**		
Cheng et al., 2015 [47]	**x**	**x**	**x**	
Ly et al., 2012 [45]	**x**			
Priyadharshini and Malathi, 2014 [46]	**x**	**x**	**x**	
Wang et al, 2015 [48]	**x**			
Lu et al., 2019 [49]	**x**	**x**	**x**	
Suriadi et al., 2017 [50]	**x**	**x**	**x**	
Wynn et al., 2019 [51]	**x**	**x**	**x**	

All these techniques aim at refining the quality of event logs, in order to obtain a more precise process model after the application of the process discovery algorithms. The quality of each process model can be measured using quality metrics (Section 3.3.2). Beyond, one method is defined to handle events with chaotic data.

3.3 Processing and Structured Business Process

After the event data extraction and the pre-processing phases, process discovery algorithms can be applied to generate a representative process model. From the BPM standpoint, the obtained model is structured, due to the low level of variability, the execution of predefined and repeated activities, no human-interaction during the BP execution, etc. Indeed, the link between BPM and process mining starts after the BP execution. Here, event logs are obtained, and process mining can apply its techniques on the recorded events logs to generate process models for future improvements. This link is clearly described in the figure 3.3.1, where PAIS [16] (sources of data) is represented as a Business Process Management System (BPMS). Through this BPMS, we extract event logs on which we will apply process mining techniques (as the improvement stage) by combining BPM knowledge and tools, to obtain an improved BP. Note that PM is not about implementing BPM systems. Instead, BPM is a practice, about improving processes and might not involve any technical system implementation.

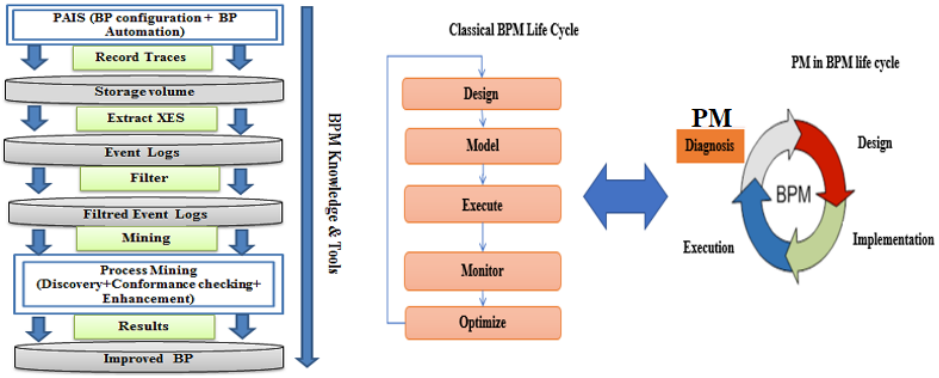


Figure 3.3.1: Mapping Process Mining to the classical BPM life-cycle

Two key benefits are easily observed: 1) No need of spending time collecting and calculating KPIs. 2) Real-time data is feeding KPIs, so off–track behaviours can be quickly identified and corrected. Therefore, the aim of PM is to use such logs to extract a BP model coherent with the recorded events. This model can then be used to improve the company business by detecting and solving deadlocks, bottlenecks, etc.

Beyond, BPM had only KPIs of the monitor phase that are defined to evaluate the BP improvement. Also, process mining demonstrates different Quality Metrics (QM) after the monitoring phase. Thus, the maintenance of the BP simplified structure is not guaranteed. To this end, we release a literature review of existing BPM methods and improvement metrics to reveal out BPM life-cycle

limitations in the BP improvement context. In the literature several methods are used to improve BPs according to the BPM life-cycle phases (see Table 3.2).

Table 3.2: BPM methods summary

Methods	**Supported phases**
Dumas et al. [67]	Redesign
Classical BPM [68]	Monitor
Koster [69]	Model, Design, Monitor
Scott Simmons [70]	Design
Semantic Business Process Management (Required functionalities) [71]	Model, Execute, Monitor
Business Process Monitor & Alignment [72]	Monitor, optimize
Grey–based DEMATEL Model (Factor implementation) [73]	Execute
Evaluation Framework for Business Process Evaluation Approaches (Implementing Data warehouse) [74]	Execute, Monitor, Optimize
Intelligent Redesign [75]	(Re)Design, Monitor

In [67], the authors focus on how redesigning BPs by presenting four dimensions to quantitatively measure process performance (time, cost, quality and flexibility). This method cannot treat the three first phases (Design, Model, Execute) of the BPM life-cycle because it is used after the execution phase. Further, the work in [68] defines improvement metrics only in the monitor phase by controlling data related to the BP objective. Then, as stated in [69], the authors present a scientific method to evaluate BPM products by defining a set of criteria for each BPM phase except the optimization phase. Also, as reported in [70] the author focuses on the objectives of performing BPM design. As well as cited in [71], the authors identify the new functional requirements for a Semantic Business Process Management (SBPM) System for each phase of the BPM life-cycle and explain the benefits of adopting semantic technologies in SBPM. The authors specify requirements, rather than solutions and metrics. Thus, in [72] the authors propose a method to evaluate and monitor the business process against performance requirements and show the effects of ongoing processes on business goals, in a real-time manner. Additionally, the paper of [73] treats only the implementation phase of the BPM life-cycle. Furthermore, the authors in [74] use data warehouse in the BPM life-cycle in order to support the tree following phases (Execute, Monitor, Optimize). Last, the work in [75] considers the whole BPM life–cycle by implementing techniques for process mining and intelligent (re)design to support the redesign and diagnosis phases and thus close the BPM life-cycle.

3.3.1 Improvement indicators

Improvement indicators focus on how the task is being performed by measuring performance and how not if individual goals are being achieved. These indicators should be measured by a ratio (generally represented by a number) which portrays the progress of the process as a whole or in part. It serves to show if the organization is achieving the objectives set. Therefore, these indicators monitor the progress of BPs.

In the literature, many improvement indicators are defined. There is no work classifies these indicators. Therefore, We group them according to their objectives [18, 19].

We denote firstly improvement indicators that aim at managing BP in term of financial and innovation roles. **These management BP metrics**, as shown in table 3.3, are based on the logic combined with the following order: The company framework is the first step to collect information from related areas. The next step to evaluate the existence of the company is to know its financing and its consumers category (satisfaction, consumption rate, etc.). This will help in defining suppliers with whom we will work. The last step is the information gathering in order to upgrade the company business objectives.

The second set of improvement indicators is **Process Metrics**. It is considered as the main group (see Table 3.4). Here, we observe the existence of the Efficiency and the Effectiveness Metrics. This type aims at looking for managing the BP during its life-cycle.

The third type of metrics is **improvement indicators related to human Resource**. These metrics are required, especially with BPs described with an inordinate proportion of manual activities. For example, insurance underwriting of commercial properties. Thus, people metrics become even more important when BPs are outsourced (see Table 3.5).

The fourth type is **improvement indicators related to Technology**. This type is important, especially with BPs that involve computer systems and application software. For instance, when customers can apply for insurance online or file claims on-line or pay their annual road tax at the local department of motor vehicles. Indeed, availability of servers and response time become important metrics that may need to be monitored (see Table 3.5).

This classification is released according to [76–84] and based on the management BP characteristics.

Table 3.3: Management metrics

Metrics	indicators	Finality or Operationalization
Society Metrics	Perceived society satisfaction	% of Society satisfied with the organization's outcomes
	Societal responsibility, sustainability, ecology, green	Number of realized ecology measures (e.g., waste, carbon dioxide, energy, water)
Financial Metrics	Sales performance	[Achieved total sales]/[planned sales] * 100
	Inventory turnover	[Annual total sales]/[average inventory] * 100
	Market share	% of growth in the last years [Sales volumes of products and services]/[total market demands] * 100
	Earnings per share	[After-tax net earnings − preferred share dividends]/[weighted average nr of shares outstanding]
	Average order value	[Aggregated monthly sales]/[monthly nr of orders]
	Order growth	[Number of orders in the current month]/[total nr of orders]
	Revenue growth	[Revenue from new sources]/[total revenue] * 100
	Operating revenue	Sales revenues
	Return on investment (ROI)	[After-tax profit or loss]/[total costs] [Revenue − cost]/[cost]
	Return on assets	[After-tax profit or loss]/[average total assets]
	Circulation of assets	[Operating revenues]/[assets] * 100
	Current ratio	[Current assets]/[current liabilities] * 100
	Net profit margin	[After-tax profit or loss]/[total operating revenues] [Total operating revenues − operating expenses − non-operating expenses]/[total operating revenues]
	Management efficiency	[Operating expenses]/[operating revenues] * 100
	Debt ratio, leverage level	[Debts]/[assets]
	Profit per customer	[After-tax earnings]/[total nr of online, offline or all customers]
Customer Metrics	Customer complaints	Number of complaints, criticisms or notifications due to dissatisfaction about or non-compliance of orders (e.g., incorrect deliveries, incorrect documentation)
	Perceived customer satisfaction	Qualitative scale on general satisfaction, possibly indexed as the weighted sum of judgments on satisfaction dimensions
	Perceived customer easiness	Qualitative scale on the degree of easiness to find information and regulations, to fill out applications, and to understand the presentation of bureaucratic language
	Customer retention	Number of returning customers
	Customer growth	Number of new customers
	Customer query time	Average time between issuing and addressing a customer problem or inquiry for information
	Customer waiting time	[Time for information about a product or service] + [time for following status updates] + [time for receiving the product or service] Max nr of customers in the queue or waiting room [Handled requests]/[total requests]
	Punctuality, delivery reliability	[Late deliveries or requests]/[total nr of deliveries or requests] % of On-time deliveries according to the planning or schedule
	Payment reliability	[Number of collected orders paid within due date]/[total nr of orders] * 100
	information availability	Information provided/not provided Time spent in asking for information about a product or service (in days)
	Customer cost	Product cost or the cost of using a service (euro)
Supplier Metrics	External delays	Number of delayed deliveries due to outage or delays of third-party suppliers
	External mistakes	% of Incorrect orders received
	Transfers, partnerships	% of Cases transferred to a partner
Innovation Metrics	Degree of digitalization	Number of digital products or services
	Degree of rationalization	% of Procedures and processes systemized by documentation, computer software, etc.
	Time for training on the procedure	Measured in hours
	Novelty in output	Number of new product or service items
	Customer response	Number of suggestions provided by customers about products and services
	Third-party collaboration	Number of innovation projects conducted with external parties
	Innovation projects	Number of innovations proposed per quarter year Nr of innovations implemented per quarter year
	IS development efficiency	Number of change requests, Time spent to repair bugs and fine-tune new applications Time required to develop a standard-sized new application % of Application programming with re-used code
	Relative IT/IS budget	[Total IT/IS budget]/[Total revenue of the organization] * 100
	Budget for buying IT/IS	[Budget of IT/IS bought]/ [Total budget of the organization] * 100
	Budget for IS training	[IS training budget]/[overall IS budget] * 100
	Budget for IS research	[IS research budget]/[overall IS budget] * 100
	Perceived management competence	Qualitative scale on the improvement in project management, organizational capability, and management by objectives
	Perceived relationship between IT management and top management	Qualitative scale on the perceived relationship, time spent in meetings between IT and top management, and satisfaction of top management with the reporting on how emerging technologies may be applicable to the organization

Table 3.4: Process metrics

Metrics	Indicators
Relevance	Identify the Business process relevant to the company
Correctness	The verification of business process models is an important step in the design phase of process–Aware information systems
Clarity	The quality of being coherent and intelligible business process
Completeness	To know how the various model elements can be grouped together to produce a coherent picture of a named business process
Consistency	Process models are consistent in the Way they represent reality
Operation time	Setup time + Run time
Cycle Time	Average time between completion of units
Throughput rate	1 / cycle time
Efficiency	Value-Added Time / cycle time
Productivity	Value-Added Time / cycle time
Utilization of an Operation	Demand / Capacity
Cost	Identification of the quantity structure and exploration of causes (simulation)
Time reference	Detection of variation and exploration of causes
Frequencies	Identification of bottlenecks in total and exploration of causes
Cost	Identification of the quantity structure and exploration of causes (real time)
Quality	Identification of the faulty process results and classification of errors
Efficiency	The performance measures associated with the process with measures related to the process itself.
Adaptability	The ability of the process to adapt to the change
Effectiveness	The performance associated with the process with respect to the change induced to the environment
Quality Criteria	Fitness ,Precision. Generalization, Simplicity
Control flow Perspective	Focuses on the ordering of activities, to find a performant representation (can be expressed in Petri net or some other notation as EPCs, BPMN, UML, Etc.)
Organizational Perspective	Focuses on the information about resources hidden in the log, in order to structure the organization by classifying people
Case Perspective	Focuses on properties of cases
Time perspective	It is concerned with the timing and Frequency of events. It makes possible to discover bottlenecks, measure service levels

Table 3.5: Technology and employee metrics

Metrics	Indicators	Finality or Operationalization
Technology Metrics	System Uptime	According to the company's measures
	Network Uptime	According to the company's measures
	Application Uptime	According to the company's measures
Employee Metrics	Perceived employee satisfaction	Qualitative scale on general satisfaction, possibly indexed as the weighted sum of judgments on satisfaction dimensions
	Average employee saturation, resource utilization for process work	[Time spent daily on working activities]/[total working time] * 100 [Work time]/[available time]
	Resource utilization for (digital innovation)	IS expenses per employee % of Resources devoted to IS development and % of Resources devoted to strategic projects
	Process users	Number of employees involved in a process
	Working time	Actual time a business process instance is being executed by a role
	Workload	Number of products or services handled per employee
	Staff turnover	% of Employees discontinuing to work and replaced, compared to the previous year
	Employee retention	% of Employees continuing to work in the organization, compared to the previous year
	Employee absenteeism	[Total days of absence]/[total and working days for all staff] * 100
	Motivation of employees	Average number of overtime hours per employee

3.3.2 BP quality and perspectives

The quality of a process model can be looked upon from various perspectives. In the context of process mining four major quality dimensions for discovered process models have been proposed [1]:

- **Fitness**, the degree to which the model allows all the observed behaviour; This metric quantifies how much the observed behaviour is captured by the model.
- **Precision**, the degree to which the model only allows the observed behaviour; The model ignores unrelated behaviour that is stored in the log. This metric quantifies how much behaviour exists in the model that are not observed. A high level of generalization model could represent much more behaviour than once presented in the log.
- **Generalization**, the degree to which the model generalizes from the observed behaviour; The model should generalize the behaviour present in the log. This metric quantifies how

well the model explains unobserved behaviours. The main difficulty with the generalization metric is the need of unobserved behaviours treatment.

- **Simplicity**, the degree to which the model is not be needlessly complex. The model should be represented in a simplified structure. This metric quantifies the model complexity.

Table 3.6: Process mining discovery algorithms according to quality criteria

Quality Metrics	**Formulas**	**Definition**
Fitness	$Fitness\,(L,M_2) = 1 - \frac{\delta(\lambda^{M_2}_{opt}(L))}{\delta(\lambda^{M_2}_{worst}(L))}$ With δ is the cost function, $\lambda^{M_2}_{opt}(L)$ is the worst case where there is no possible synchronization between the trace and the model. $\lambda^{M_2}_{opt}(L)$ is the obtained costs for each optimal alignment.	Quantifies how much behaviour is captures the model.
Precision	$Precision\,(L,M) = \frac{1}{\lvert E\rvert}\sum_{e\varepsilon E}\frac{en_T\,(e)}{en_M\,(e)}$ With E is the set of events in she T logs, A is the set of activities, $en_T\,(e) \subseteq A$ is the set of activities presented in the traces and in $M(e) \subseteq A$ the set of activities presented in the model.	Quantifies how much behaviour existed in the model that was not observed.
Generalitazion	Based on the precision results.	Quantifies how well the model explains observed system behaviour.
Simplicity	It is observed and not calculated.	Verifies the simple structure of the process model.
Node detail	$\frac{\sum \mathrm{v} \in \mathrm{V}^{\mathrm{s(v)}}}{\sum \mathrm{n} \in \mathrm{V}^{\mathrm{s(n)}}}$ Where N corresponds to all the primitive nodes (present, aggregated or deleted activities) or the Fuzzy model F, $V \subseteq N$ is the subset of all visible nodes and s is a function that associates each primitive noise F with a measure of importance.	Quantifies activities displayed in the Fuzzy model, relatively to the aggregated or deleted activities.
Conformance	$\frac{\mathrm{M\,(T)} - \mathrm{d} + 1}{\mathrm{M\,(T)} + 1}$ With M the total number of activities present in the logs T and l is the number of deviations identified.	Describes the alignment between the Fuzzy model F and the logs T.

There is a particularity for the fuzzy miner algorithm metrics, the output model is a fuzzy model. To evaluate the fuzzy model, two metrics are available: Node detail and conformance.

- **Node Detail** describes activities displayed in the Fuzzy model, related to the aggregated or deleted activities. Nodes of visible activities are called explicit nodes, while nodes corresponding to an activity are denoted as implicit ones.
- **Compliance** is a measure that describes the alignment between the Fuzzy model F and the logs T. Each activity in the logs that does not exist in the Fuzzy model will be counted as a deviation.

Table 3.6 classifies process mining discovery algorithms, according to quality criteria. This is based on the logic of the final representation, i.e., Algorithms producing Petri net models are suitable with fitness, generalization, and precision metrics, while algorithms producing fuzzy models are adequately evaluated using the node detail and the conformance metrics.
Also, we can define four Considered Perspectives on a BP that are often considered in the literature on BPM, process modelling, and process mining [1, 85]: the control-flow perspective, the resource perspective, the data perspective, the time perspective. This set of perspectives is not comprehensive and there may be other or additional perspectives from which processes can be looked upon, e. g., costs or risks. However, we argue that these four perspectives are significant perspectives for process mining in term of BP improvement.

Control-flow perspective. The control-flow perspective sometimes also called the behaviour or behavioural perspective, of a process describes the order in which its activities should be executed. The overall control-flow of a process corresponds to all the possible sequences of activities. Therefore, the control-flow perspective is, usually, the starting point for a process mining analysis.

Resource perspective. The resource perspective, sometimes also called the organizational perspective, describes the resources required for the execution of a process and how they interact with each other. Resources can be human resources and non-human resources (e. g., machines and materials). Possible artefacts of the resource perspective can be, e. g., social network graphs, assignment rules, and allocation constraints regarding which resources may execute activities.

Data perspective. The data perspective, also denoted as case, object, information, or informational perspective, describes which existing data objects are required as input during the execution of the process, used for control-flow routing decisions, and how data objects are created and updated during the execution of the process. In the process, the recorded value is used to route process instances according to the described rule.

Time perspective. The time perspective focuses on all time-related aspects of the process. In addition to the ordered sequence considered by the control-flow perspectives, activity executions take time and occur at a specific moment in time, e. g., before a predefined deadline. Often, there are rules regarding the timing between activities too.

3.3.3 Synthesis

To conclude, this section demonstrated that existing BP improvement indicators are arbitrarily used and reused during the BPM life-cycle. In this sense, no approach defines adequate improvement indicators to each BPM phase. Therefore, it is required to group them according to specific business conditions and rules, in order to maintain the BP improvement from the design phase to the optimize phase. For the optimize phase, BPM does not clearly introduce improvement metrics to achieve the control-flow perspective (the degree to which the model allows all the observed behaviour or how data objects are created and updated during the execution of the process, etc.) till the introduction of process mining quality metrics and after the business process execution. In this context, no approach maintains the BP improvement from the first phase of the BPM life-cycle.

3.4 Processing and Unstructured Business Process

Business Processes with complex structure are incomprehensible, unreadable, and can be emerged because of human intervention during the BP execution. Indeed, unstructured processes (knowledge-intensive) are loosely defined and rely on access to readily available knowledge. The problematic is how to obtain a simplified and improved representation of an Unstructured BP (UBP) using process mining techniques.
The existence of structured processes (data-intensive) within the unstructured ones make process mining usable for both systems: data-intensive & knowledge intensive.
The early style of data-intensive (BPM) defined processes as the core of the system. This style lets the data-flow through control flows as process instances. This means the process is primary and normally static. In contrast, knowledge-intensive places data in the center which is able to support the surrounding processes to make decisions whenever necessary. The data is considered as a primary. In most cases, processes may not be fully predefined, knowledge workers have to define them on the fly depending on the situation. All in all, SBP contains the predefined process routes, on the other hand, with UBP the case itself is the main focus.
Therefore, UBP still have more issues that are difficult to be analysed and hard to be understood due to its complexity and variability. Moreover, it is required to provide instantiate response in a dynamic manner to these unstructured BPs.
In the following paragraphs, we present existing works, related to those 3 issues, in the process mining context: complexity, variability and dynamicity.

3.4.1 Complexity

In the last decade, many scientific studies have been defined to deal with the process mining issues. More specifically, approaches and techniques that allow reducing the complexity of unstructured BPs to a simplified representation. Through this section, we illustrate the still encountered issues related to process mining applied to UBPs. In this context, the main challenge is how to deal with the event logs complexity [2] at runtime.
Process mining has difficulties to handle complex event logs properly. Fortunately, it is generally agreed that decomposing processes is the best technique to deal with the complexity challenge.

The technique decomposes process mining problems into many smaller semi-problems that can be solved in short time [86]. In the literature, many ways to partition process mining problems exist. However, these approaches of decomposition are not consistent among studies: some authors used the divide-and-conquer approach [87], some used the Single-Entry Single-Exit technique [88], others used the notion of process cubes [89], the four conflicting quality dimensions, especially in conformance checking decomposition [90], some studies answer to issues related to computation and visualization [91] and others have based their decomposition on clustering [92]. Moreover, each study for process mining decomposition has some limitations and need to be investigated in further works. The most relevant problem is choosing the appropriate algorithm to mine the composed event logs provided from an original complex event log. The structuring techniques are based on the following four studies of [93–96]. The first two approaches treat only unstructured acyclic rigid fragments with parallelism. The second two approaches deal only with rigid fragments without parallelism (exclusive gateways). This gap requires a hybrid approach that combines between the two aforementioned rigid fragments categories. To do so, the recent approach of [97] is applied. This later represents a discover-and-structure method for generating a SBP from event logs. This method builds upon the hypothesis: instead of attempting to discover a block-structured process model directly, higher-quality process models can be obtained by discovering abstracted representation of an UBP, then transforming it to a structured one in a best-effort manner. This approach aims at discovering a SBP by operating into the following structuring techniques: 1- gateways structuring (repair unstructured gateways' representation), 2- clones' removal (remove from the process model repeated activities and by necessity actions) and 3- soundness repair (verify soundness of the obtained process model). The approach uses the BPMN as a process model representation language.

Indeed, executing loosely structured processes generate unstructured behaviours at runtime. They are difficult to be analysed and hard to be understood, due to its complex structure. Besides, it is required to treat this complexity problem during the application of the operational support actions. In this context, several approaches have been developed. Since Process Mining Manifesto released at the end of 2011 [85], we focus on research papers published from the beginning of 2012 year:

The [98] develops a concrete implementation of operational support meta-model, based on the work-flow system and the ProM framework. This meta-model treats four types of queries: simple queries, compare queries, predict queries, and recommend queries.

The [99] proposes a global approach that learn discovered behaviours to predict the classes of visible and invisible traces. The discovered signature patterns allow the distinction between various classes of behaviour and by necessity related business conditions. The [100] suggests a method that predicts process risks by applying decision trees to the logs of previous process executions, taking into account multi-perspectives of process mining like: process data, used resources, task durations, and contextual information. To do so, the proposed method helps the process participants to make risk-informed decisions. Likewise, the [101] explores an approach that forecasts the remaining processing time and recommend activities to reduce risks.

The [102] illustrates a method that can prevent the undesirable behaviour from occurring in next executions. This is done based on the Markov Cluster (MCL) algorithm with the ability to detect changes of a process according to the selected perspectives.

The [103] presents framework for predicting dynamic behaviour from event logs. It is capable of correlating and clustering dynamic behaviour. The framework allows the prediction of the executor of a certain activity, the remaining time to the end of the process instance, the next activities to work on, and the outcome of the executions of process instances.
The [104] demonstrates a multi–stage deep learning approach for BP event prediction that aims at predicting the next BP event, considering the execution of log data from the previously completed process instances. This is done to predict the BP events, to initiate timely interventions for undesired deviations from the desired workflow.
The [99] proposes a framework for detecting and analysing, at runtime, BP deviances, which leverages both a novel incremental approach to the discovery of an ensemble–based deviance detection model. Moreover, the [105] establishes a method that addresses the problem aiming at learning the impact of past events on the future events using deep learning methods. It is a deep predictive model for multi–attribute Event Sequence. On one hand, all the cited papers reported on the application of process mining for operational support, do not use the orchestration of the whole existing process mining activities (Discover, Explore, Check, Compare, Promote, Diagnose, Enhance, Detect, Predict and Recommend). On the other hand, some of them only generate operational support to structured BPs.
The ability to know in advance the trend of running process instances, with respect to different features, such as the expected completion time, would allow business managers to timely counteract to undesired situations, in order to prevent losses. Therefore, there is a need for an operational support approach that bridges the gap between complexity and operational support actions (detection violations, predicting events and recommending actions).

3.4.2 Variability

Business process variability is defined as the mechanism that permits users to perform their research according to their objectives in diverse ways. Indeed, different behaviours can be produced. Even users with the same objective may follow different paths and stand different sub-processes denoted as personalized/customizable BPs that vary in terms of structure, objective, and result. This puts forward the difficulty of obtaining and studying user's behaviours [106, 107]. Here, we denote the variant concept that may refer to the activity variant or the process model variant. Moreover, the variation point defines a crucial element (i.e., a node or a sequence flow) of the customizable process model that can be customized via model transformations. Accordingly, the customizable process model encapsulates customization decisions between process variants that need to be made either at design-time or runtime. Customizable process models capture a family of process model variants in a way that the individual variants can be derived via transformations, e.g., adding or deleting fragments.
Therefore, the difficulty of representing some BPs type emerges from their variability concept [108]. This later change according to different contexts and requirement. Even though managing process variability is a non-trivial task because it requires specific standards, methods, and technologies, it still involves many parameters that are not always formally defined. For example, designing the reference process model, which represent the commonalities from the process family,

is a challenge, as well as the necessary adjustments to configure a specific process variant. Thus, each BP variation is applicable to a different situation that affects specified customization criteria in a different way. Such criteria may include high-level qualities or non-functional goals e.g., such as key performance indicators (KPIs) or operational constraints that prescribe patterns to decide which BP must be followed. In this context, two main definitions can be acquired [106, 109, 110]:

- **Variability by restriction** starts with a customizable process model that contains all behaviour of all process variants. Customization is achieved by restricting the behaviour of the customizable process model. For example, activities may be skipped or blocked during customization. In this setting, one can think of the customizable process model as the union or Least Common Multiple (LCM) of all process variants. Customizable process models of this type are sometimes called as configurable process models.
- **Variability by extension** takes the opposite starting point. The customizable process model does not contain all possible behaviour, instead it represents the most common behaviour or the behaviour that is shared by most process variants. At customization time, the model's behaviour needs to be extended to serve a particular situation. For example, one may need to insert new activities in order to create a dedicated variant. In this setting, one can think of a customizable process model as the intersection of all process variants under consideration.

In this regard, process mining algorithms can be used to mine BP variability (variant details) in order to manage and decide the suitable path to execute.

3.4.3 Dynamicity

The concept of dynamicity defines the process that can change some BP activities during runtime under various conditions, which are emerged from real-time variables. It can be adapted according to internal or external environment changes. In this context, we focus on Ad-hoc processes that are characterized by a non-defined workflows design. Indeed, the control-flow between activities cannot be modelled in advance but can simply occur during runtime. Here, users must be able to decide what to do and when. They must also be able to assign work, as sub-process, to other people and create interactions among various users. This puts forward the difficulty of treating dynamically Ad-hoc processes.
The dynamicity can also be nominated, in such research, as the flexibility by design concept [111]. However, our objective is to obtain a dynamic Ad-hoc BP. Indeed, we focus on research that combine between: 1) process mining techniques and Ad-hoc BP. 2) dynamicity and Ad-hoc BP and 3) dynamicity and process mining techniques. To this end, we present a brief literature review of those papers.
In [112] a proposed tool for mining Ad-hoc processes has been demonstrated. Here, the ambiguity concerns the definition of sub-processes with different guiding conditions. Besides, the adaptive process does not respect possible external changes. In [113, 114], authors present two case studies in healthcare domain, especially into the emergency department. Here, the process discovery technique illustrates possible processes that can do a patient in the emergency department. These cases must be generalized into a clear approach. The [115] presents an approach dealing with

dynamicity into Ad-hoc BP using a pre and post sections to subsequently execute actions, based on the set of process goal. In this study, process mining did not match in the Ad-hoc BP definition. Additionally, the paper [116] discussed the context changes (external and internal) and conditions that could influence the BP dynamicity. Some works as [117] treat only the internal context, others do not predefine the automated BP activities and others do not discuss about changes of the context of activities (external to internal Ad-hoc process environment). Further, the [118] demonstrates just external changes. In spite of above, BP dynamicity [119] is treated as agility, adaptability and flexibility, where internal changes have been taken into consideration.
According to the aforementioned studies, we observe that few researchers take up the study of the combination between: Dynamicity, Ad-hoc BP, Process mining techniques, external and internal changes of the BP environment or conditions related to Ad-hoc BP. To this end, we must treat dynamicity into Ad-hoc BP, using process mining techniques and taking into consideration runtime changes.

3.4.4 Synthesis

In this section, we have discussed recent approaches that predict, structure, manage variability and dynamicity for unstructured BPs. This is tackled related to the process mining field. First, we have illustrated approaches that treat event logs complexity. At this stage, it is important to explore structured sub-processes into the unstructured ones. Second, we have discussed the challenge of operational support that can be acquired with complex process models. For instance, it is required to predict the remaining time dimension to execute future activities during the BP execution. Third, we have defined the variability challenge where we have shown over related issues that must be resolved. Fourth, we have discussed the dynamicity challenge and the necessity to predefine processes to select the suitable adaptive sub-process.

3.5 Summary

In this chapter, we firstly discussed the pre-processing event data techniques in the process mining field. From one hand, we have cited existing approaches that consist of filtering out deficiencies like noise, infrequent and completeness. On the other hand, one approach is defined to detect chaotic activities using supervised algorithms. In the case of no ground knowledge about chaotic activities in the business process, this later still none applied to reveal chaotic activities. In this sense, chaotic activities are frequent, distinct from infrequent behaviour and impact the quality [59] of process discovery results (discovered models) likewise its process of improvement. Therefore, **we aim to treat this event logs deficiency for more precise process models results**.
Secondly, we have discussed the processing phase techniques. Here, we can obtain two possible BP structure results:
1-BP with simple structure, which is understandable and readable. Thus, we focus on **how to maintain the BP with simple structure, using improvement indicators in combination with process mining techniques and based on BPM life-cycle**.
2- BP with complex structure is incomprehensible and unreadable. Due to the human interac-

tion giving to knowledge workers the permission to define processes on the fly in unpredictable situations, processes seem to be unstructured and hard to be enhanced. The challenge is **how to improve processes in unstructured conditions**. At this stage, we have observed two main structures of BPs: SBP and UBP (see Table 3.7 and Figure 3.5.1).

Table 3.7: Key difference between data-intensive and knowledge-intensive systems

Data-intensive systems	Knowledge-intensive systems
Able to define an ordered sequence of activities. Example: For cell phone service: it happens thousands of times a day, and the process it essentially fixed.	Able to define an unordered set of activities. Example: Crime investigation, it requires following up on various clues, which are unpredictable. There are various tests and procedure to use, but they will be performed only when needed.
The sequence of activities is stable and hardly changes. The process is predictable and repeatable.	The activities occur unpredictably.
The process determines the events.	The events determine the process
The activities are defined, can be automated and repeatable.	People determine activities and select the best solution for each case.
External documents are not an essential part of the process.	External documents are important to make a decision.

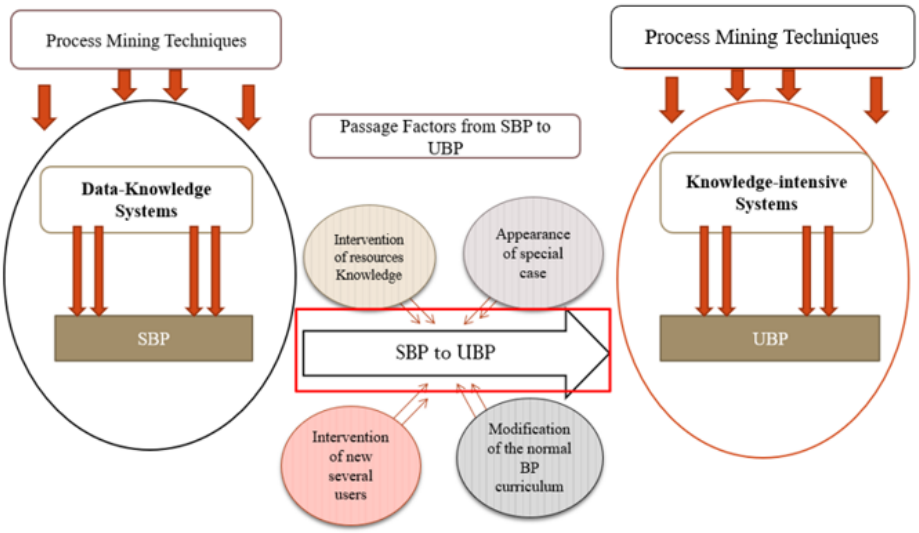


Figure 3.5.1: SBP to UBP transformation

Last, we summarize all the aforementioned challenges in figure 3.5.2. The figure starts with events

logs as input. Then, we try to filter deficiencies, especially chaotic activities (Chapter 4). Next, we can face two BP structures. In the case of SBPs, we try to maintain this structure during the BPM life-cycle (Chapter 5). In the case of UBPs, the variability, dynamicity and complexity challenges must be tackled (Chapter 6).

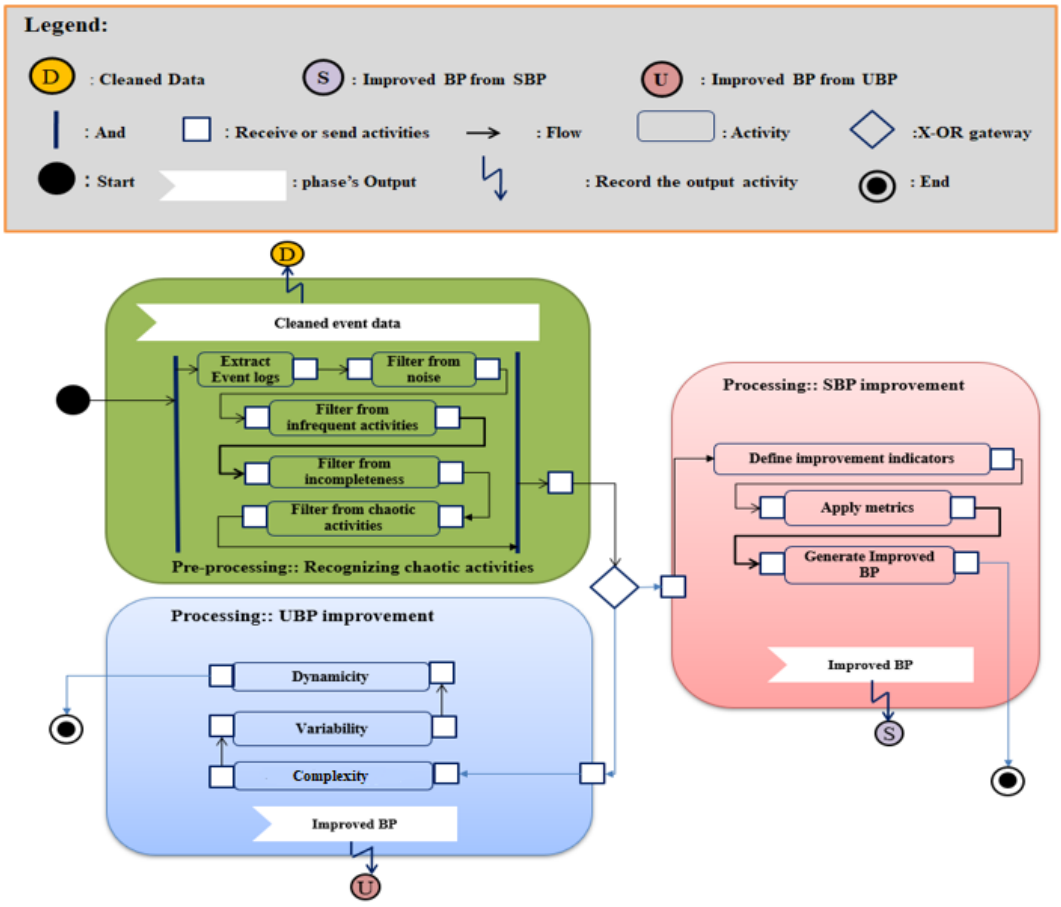


Figure 3.5.2: Thesis challenges

Part II

Contributions

Chapter 4: This chapter illustrates the first contribution of this thesis. It consists of recognizing chaotic activities from event logs. Indeed, we propose a pre-processing method that uses unsupervised machine learning algorithms.
Chapter 5: This chapter aims at processing structured business processes using process mining techniques and a set improvement indicators. In this sense, we propose an integrated approach to improve and maintain the BP simplified structure from the design to the optimize phase.
Chapter 6: This chapter treats unstructured business processes in terms of complexity, variability and dynamicity using process mining techniques. First, we propose an operational support approach to treat the complexity of Unstructured Business Processes at runtime. Secondly, we study the BP variability taking into consideration variants paths. Last, we present an integrated methodology to select suitable sub-process in a dynamic manner by answering to pre-defined business rules.

Chapter 4

Pre-processing: Chaotic Activities Recognizing

This chapter presents our first contribution in this thesis. It consists of recognizing chaotic activities from event logs. Indeed, we propose a pre-processing method that will use unsupervised machine learning algorithms.
The remainder of this chapter is organized as follows: Section 4.1 presents the chapter background. Section 4.2 introduces concepts and notations used throughout the chapter. Section 4.3 illustrates our approach phases that aims at recognizing chaotic activities, based on unsupervised machine learning algorithms. Section 4.4 presents three illutrative examples to evaluate the applicability of our recognition approach. In the first case, we artificially fill in the original event logs file a set of chaotic activities, to analyse whether our approach can reveal out these activities. For the other two cases, we try to detect chaotic activities without having knowledge on which activities are explicitly chaotic. Section 4.5 summarizes this chapter.

4.1 Background

Related to the chaotic activities problematic, an approach [3] has been proposed to filter out this kind of activities. The approach is based on supervised data mining algorithms; this can be simply matched with supervised machine learning algorithms. Its concept requires experimental results and more samples for identifying chaotic activities. This obligates the existence of chaotic activities and none-chaotic activities examples, to train the system. Finding enough examples can be a challenge, as we have no ground truth knowledge on which activities are truly chaotic. This cannot allow detecting new ones. To do so, we develop an approach for recognizing chaotic activities, based on unsupervised machine learning algorithms.
In this context, Machine learning [32, 33] is an application of artificial intelligence (AI) that provides systems with the ability to automatically learn from experience without being globally programmed. In reality, we observe many studies [23, 120, 121] that discuss machine learning techniques in the context of process variant analysis, which is a large subset of the process mining field. Therefore, process mining and machine learning go well together. As we recently observed

the Process Mining for Python (PM4Py) library [41], which integrates process mining techniques with python language into machine learning frameworks (TensorFlow, Anaconda, Keras, etc). It is considered as the only concrete implementation of process mining closed to machine learning with data science libraries, e.g., Pandas, NumPy, Scipy and Scikit-learn. In this respect, process mining is application-oriented discipline. Its objective is to invent approaches to improve business processes, while machine learning aims at developing methods that can resolve related problems and settings. Thus, machine learning algorithms are useful for process mining application.

4.2 Notations and concepts

This section introduces concepts and notations used throughout the chapter. For this purpose, this section will be organized as follows: 1- Clean event logs (Noise, Infrequent, incomplete). 2- Data selection, Encode and Reduce dimensionality. 3- Isolation Forest Algorithm to reveal out chaotic activities.

4.2.1 Clean event logs

We filte L (event logs) from noise, incomplete and infrequent behaviours. This is done by using the existing filtering functions (F1, F2 and F3 of Figure 4.2.1) in the literature. In this respect, we can represent L (1) and $L^{'}$ (2) respectively as a matrix with (N: rows, M: columns) and (n: rows, m: columns), *where* $m \leq M$ *and* $n \leq N$. At this stage, we obtain cleaned data, from which we can select data for specific treatment.

$$L = \begin{bmatrix} a_{11} & \cdots & a_{1N} \\ \vdots & \ddots & \vdots \\ a_{M1} & \cdots & a_{MN} \end{bmatrix} = (a_{ij})_{1 \leq i,j \leq M,N} \ (1)$$

$$L' = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} = (a_{ij})_{1 \leq i,j \leq m,n} \ (2)$$

4.2.2 Data selection, Encode and Reduce dimensionality

Data selection: Captures data from event logs file. For example, $_{cap=}$Ds(.case,.activity)= array([actv$_{01}$, ..., activity$_{0n}$],...,[actiDity$_{m1,...,}$activnty$_{mn}$]). In this regard, we focus on cases and their corresponding activities. We can represent them into a matrix with (nxm) as mentioned in (3). L$_{cap}$=Ds (L) = (a$_{ij}$) with [case:n, activity:m]). The timestamp dimension is implicitly treated, because it organizes the order in which the activities are executed.

$$L'_{\text{cap}} = Ds(L') = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} = (a_{ij})_{1 \leq i,j \leq m,n} \ (3)$$

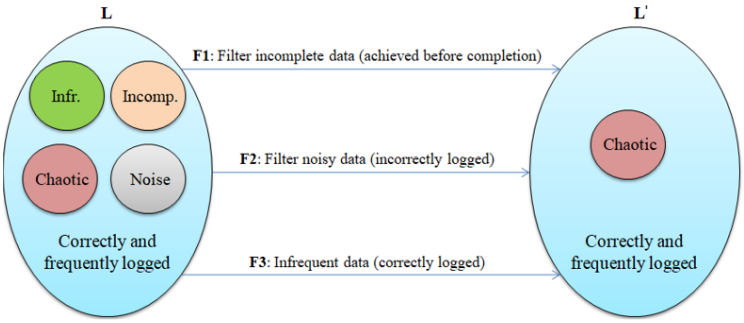


Figure 4.2.1: Filtering out noise, infrequent and incompleteness deficiencies

In the case of three dimensions, we use the tensor representation, while we alternate to ND array for more than three dimensions (cases, activities, resources, costs, etc.).
In reality data encompasses different types of variables: quantitative and qualitative. Due to the machine learning algorithms, which are applied on numeric variables, we use an encoding function, where $(a_{ij})_{\substack{1\le i\le n\\ 1\le j\le m}} \in \mathbb{R}^{m\times n}$

Encode: Encodes categorical (qualitative) features as an integer or quantitative data or array (4). We denote the encode function (E_{lpcap}). Each line of the matrix corresponds to a (case) sequence of recorded activities.

$$E_{lpcap} = \text{Encode}\left(L'_{\text{cap}}\right) = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} = (a_{ij})_{\substack{1\le i\le n\\ 1\le j\le m}} \in \mathbb{R}^{m\times n} \quad (4)$$

n : cases

m : activities

A feature is an attribute which describes the object on which analysis and prediction is to be done. When data is presented in tabular format, it describes instances in rows and attributes in columns. In machine learning, attributes (features) can be used to form a mathematical model to describe that object.

PCA (Reduce dimensionality): Principal Component Analysis is a linear dimensionality reduction using Singular Value Decomposition of the data to project it to a lower dimensional space [122]. The dimension is reduced taking into consideration 99% of event logs. We try to decrease events complexity, for targeting simple events. Therefore, we use the PCA algorithm to define our function (R_{Elpcap}) that aims at reducing event data complexity (5):

$$R_{\text{Elpcap}} = \text{Reduce}\left(E_{\text{lpcap}}\right) = \begin{bmatrix} a_{11} & \cdots & a_{1n'} \\ \vdots & \ddots & \vdots \\ a_{m'1} & \cdots & a_{m'n'} \end{bmatrix} = (a_{ij})_{1\le i\le n'} \in \mathbb{R}^{m'\times n'} \quad (5)$$

n' : reduced dimension of activities (take into consideration 99%)

m' : reduced dimension of cases (take into consideration 99%)

By applying PCA, we do not need to know the labels of data but simply working with all the existing features, which are unsupervised. PCA is not only an essentially dimensionality reduction

algorithm, but also a useful tool for noise filtering, visualization, and feature extraction. It is highly based on Linear algebra, and only works with data in numeric format. When data are fitted with principal components, PCA will find the best possible characteristics of the data and represents them with limited number of PCs. PC is stand for principal components, which are defined as a linear combination of the data's original variables. It represents the directions of data that explain a maximal amount of variance. However, they can be as much as informative compared with real data but much simpler. Most of the information from the original data are squeezed into the first component from the whole set PCs we get.

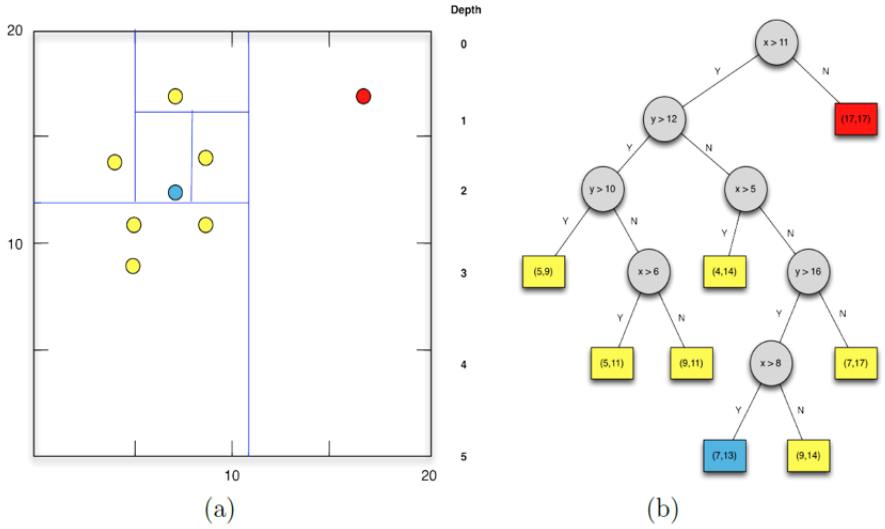


Figure 4.2.2: Isolation Forest application's example

4.2.3 Reveal out chaotic activities

Isolation Forest algorithm (IF): isolates observations by randomly defining a feature and determining a split value between the maximum and the minimum values of the defined feature [123]. This algorithm logic consists of isolating anomaly observations that uses specific conditions to separate cases with anomalies from the normal observations. Besides, isolating normal observations require the use of an anomaly score. This score can be calculated as the number of conditions that allows distinct elements into a given observation. In this respect, the IF algorithm creates isolation trees or random decision trees. Then, the score is calculated as the path length to isolate the observation. This algorithm requires relatively small samples from large datasets to derive an anomaly detection function. This makes it fast and scalable. It does not require example anomalies in the training data set (Cf. Figure 4.2.2). In this example dataset (a), a randomly constructed binary search tree isolates the anomalous value (17,17) (shown in red) in just one division, whereas the medoid (k representative objects that are located in the center of the class they define) value (7,13) (shown in blue) is isolated in five random divisions. (b) The anomaly has a tree depth of 1, compared to 5 for the medoid point. According to the aforementioned IF algorithm definition, we use it in the function (6) to recognize chaotic activities:

$$IF_{\text{REIpcap}} = \text{Isolation\_forest}\left(R_{\text{Elpcap}}\right) = \begin{bmatrix} a_{11} & \cdots & a_{1n\prime} \\ \vdots & \ddots & \vdots \\ a_{m\prime 1} & \cdots & a_{m\prime n\prime} \end{bmatrix} (a_{ij})_{1\leq i\leq n\prime} \in \mathbb{R}^{m\prime\times n\prime} (6)$$

In this sense, we assume that the obtained results, from (6), are in form of graph (O; I; j), where the first dimension X concerns cases and the second dimension Y targets activities. Due to this definition, we can apply $Y\left[IF_{\text{REIpcap}}\right] = 000010010...$, to illustrate the set of outlier's behaviours from event data L', such the 1 determines a chaotic activity and 0 represents cleaned activities (7). Besides, the $X\left[IF_{\text{REIpcap}}\right]$ application aims at defining cases that can handle chaotic activities. In case of simple event logs without revealing complex structure, it is recommended to ignore the reduction dimension function$R_{\text{Elpcap}} = \text{Reduce}\left(E_{\text{lpcap}}\right)$ and directly apply the following function:

$$IF_{\text{Elpcap}} = \text{Isolation\_forest}\left(E_{\text{lpcap}}\right) = \langle 000010010\cdots\rangle (7)$$

4.3 Our chaotic activities recognizing approach

In this section, we propose a new approach for recognizing chaotic activities from simple and complex event data. In this sense, Our approach [17] (see Figure 4.3.1) consists of the following four phases: 1– Data preparation, 2– Data selection, 3–Features selection and 4– Unsupervised learning.

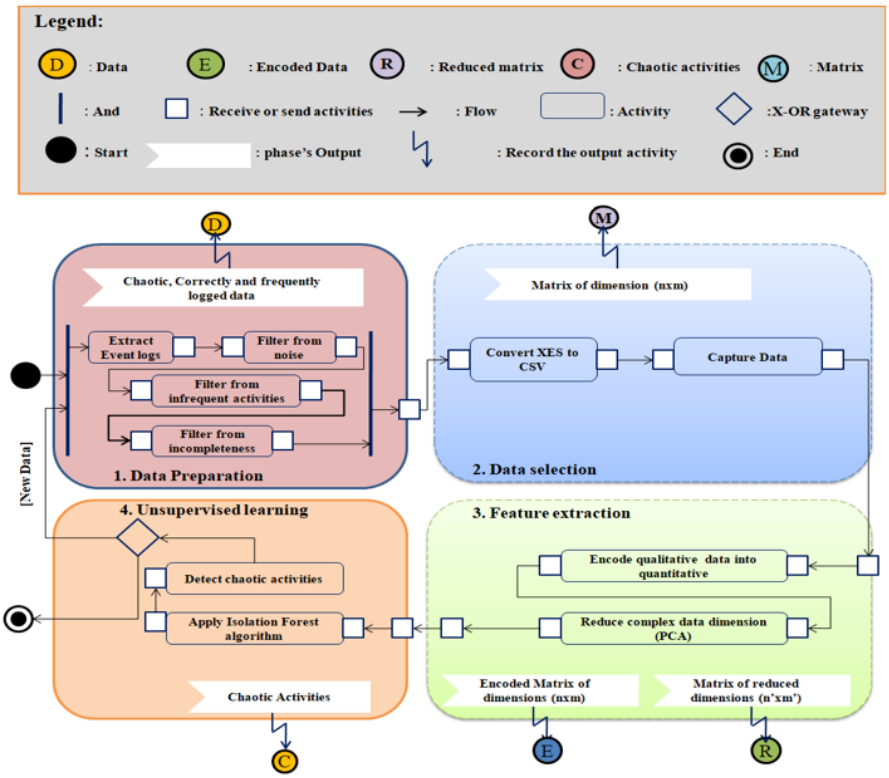


Figure 4.3.1: Our approach's phases overview

4.3.1 Data preparation

We start our approach by the data preparation phase. This phase uses three filtering functions, to clean event data from noise, incompleteness, and infrequent behaviours. This phase's output is a set of cleaned event data, which are expressed in (2) as L', where $m \leq M$ and $n \leq N(MxN$ dimensions of the original event data matrix):

$$L' = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} = (a_{ij})_{1 \leq i,j \leq m,n}$$

It is required to clean event data, to have a high-quality data set. For example, we need to do something with missing data or irrelevant (no–contribution) data as the field has times missing value. In this context, we keep these events and ensure the field is set to Numpy.NaN. In addition, it is quite possible to record events without a case id, keeping them could cause mistakes; as such we delete them. All these operations can be achieved with PM4PY library and python libraries.

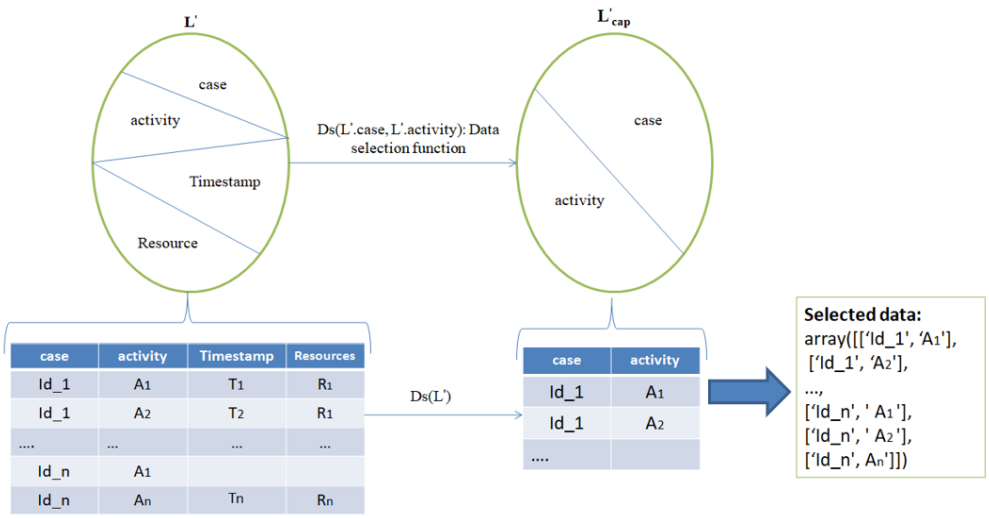
4.3.2 Data selection

After obtaining cleaned data, the next step is to select events on which we try to apply specific treatment. Here, we describe the selection phase by two steps. The first step involves converting the XML based event log format (XES: `https://xes-standard.org/`) to a table/CSV (command–separated values) file format (see Figure 4.3.2) using the PM4Py Python library [41]. In this sense, each row is representing an event instance. Hence, for each case there are multiple rows (events). Each column is an attribute/field.

Figure 4.3.2: Data transformation

This step is required to apply machine learning algorithms. The reason why we choose CSV format is that this format is more widely being accepted and simply programmable by Python, such as panda's library (`https://pandas.pydata.org/`), which is a library in Python that can read CSV file into data frame. Additionally, this choice could also fulfil the further work with machine learning model fitting. In a CSV file, each event is usually stored in a single line, and the attributes of that set of events are separated by a symbol, such as a comma (,).
The second step is to capture data frame. In this sense, we focus on cases and their corresponding activities (see Figure 4.3.3):

$$\begin{aligned} L'\text{cap} &= \text{Ds}(L'.\text{case}, L'.\text{activity}) \\ &= \text{array}\Big([\text{activity}_{01}, \ldots, \text{activity}_{0n}], \ldots, [\text{activity}_{m1,\ldots,} \text{activity}_{mn}] \Big) \end{aligned}$$



case	activity	Timestamp	Resources
Id_1	A1	T1	R1
Id_1	A2	T2	R1
....	...	...	...
Id_n	A1		
Id_n	An	Tn	Rn

case	activity
Id_1	A1
Id_1	A2
....	

Figure 4.3.3: Selecting data frame example

4.3.3 Features' extraction

Generally, this phase aims at finalizing the data source. Firstly, we must encode qualitative to quantitative data. This is done by combining rows per case, since we want a row to represent a case. As a result, the CSV file has N rows. Then, we must apply the PCA algorithm for reducing data dimension, where 90% of data is considered. Consequently, we obtain an informative case-based CSV input data source, having k features of users' cases. An example of quantitative data is shown in Figure 4.3.4. Also, we observe how event data are encoded and how its reduced format is represented.

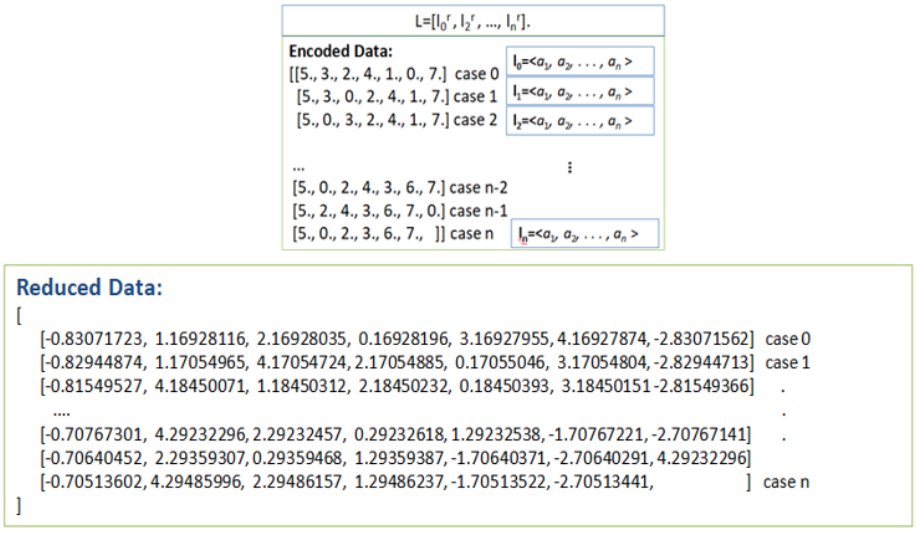


Figure 4.3.4: Features' extraction example

4.3.4 Recognizing chaotic activities using Unsupervised Learning

The fourth phase consists of recognizing chaotic activities from simple or complex event data. To that end, we develop algorithm 2 and 3, respectively for simple and complex events. Algorithm 2 shows how it is possible to detect chaotic activities without reducing the event data dimension. We target, here, a set of simple data. We start with cleaned event data L', we proceed normally in this algorithm core, by capturing data and extracting features (convert and encode event data). Then, we isolate outlier behaviours. This is done by applying the following two algorithms (2 and 3). Algorithm 2 is developed for simple data (see Figure 4.3.5 in part a), while algorithm 3 is proposed to complex events (see Figure 4.3.5 in part b).

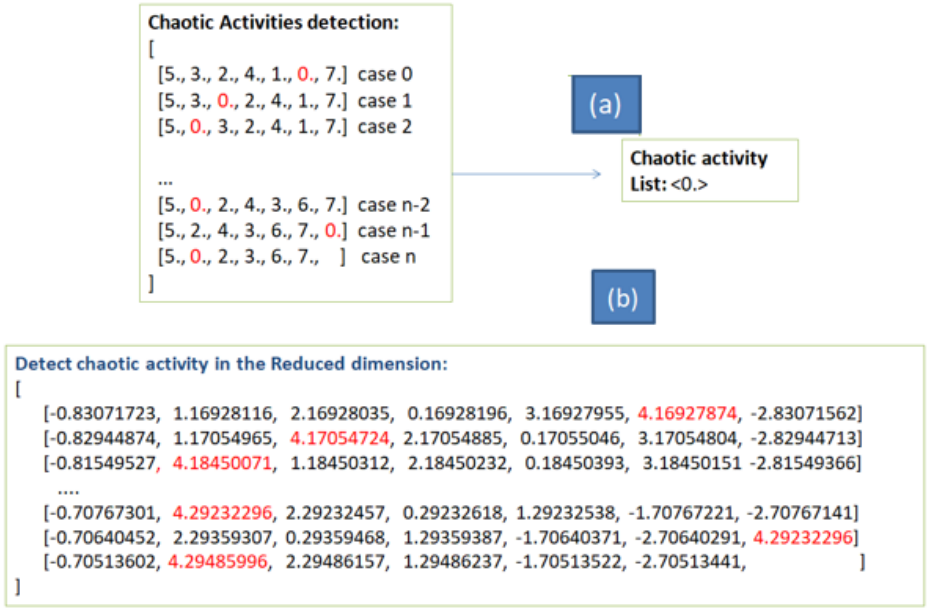


Figure 4.3.5: Recognizing chaotic activities example

To reduce event data complexity, we propose the algorithm 1. This algorithm takes as input complex event logs, which is $\mathbf{C}_L$. This value will be attributed as an array of two dimensions expressed as L', where rows are representing cases and columns mean activities. This algorithm provides a list of simple (reduced) event data $R_L <$ rows $><$ columns $>$. In this algorithm's core, we apply three functions: Capture_Data() for selecting event log, Encode_CD() for encoding captured data and PCA() for reducing complex event data dimension. Additionally, we use the fundamental function data READ (L'), to read event data content.

Algorithm 1: Reduce complex event logs dimension

Input: Complex Event logs filtered from noise and infrequent behaviours $\mathbf{C}_{L'}$

Output: Array with two-dimension $R_{L'}$ <cases><activities> as Simple event logs

Initialization:

$L' = \mathbf{C}_{L'}$

$R_{L'}$ =<><>

Main Procedure:

Data = READ (L')

CD = Capture_Data (data.cases, data.activities)

E = Encode_CD(CD)

$R_{L'}$ = PCA(CD)

return $R_{L'}$

Besides, algorithm 3 is similar to algorithm 2, except in the use of the PCA(x) function that treats event logs with complex dimension.

Algorithm 2: Recognize chaotic activities from simple event logs

Input: Event logs filtered from noise and infrequent behaviours L'

Output: Array with one dimension represents chaotic activities **CH** <activities>

Initialization:

L = L'

CH =<>

rank =<>

Main Procedure:

DA = READ(L)

CD = Capture_Data(DA.cases, DA.activities)

E = Encode_CD(CD)

OB = Isolation_Forest (E)

IA = y[OB]

k = 0

for $i = 0$ **to** N_i **do**

  for $j = 0$ **to** N_j **do**

    if AI_i=E_{ij} **then**

      $rank_k$=j

      $CH_{L'}$ = OB_i

      k=k+1

    end

  end

end

ranks=set(rank)

for $k = 0$ **to** N **do**

  Print($ranks_k$)

end

return CH

Algorithm 3: Recognize chaotic activities for complex event logs

Input: Complex Event logs filtered from noise and infrequent behaviours $\mathbf{C}_{L'}$

Output: Array with one dimension represents chaotic activities $\mathbf{CH}_{L'}$ <activities>

Initialization:

L = $\mathbf{C}_{L'}$

$\mathbf{CH}_{L'}$ =<>

rank =<>

Main Procedure:

DataL = READ (L)

CD = Capture_Data (DataL.cases, DataL.activities)

E = Encode_CD(CD)

$R_{L'}$=PCA (CD)

OB= Isolation_Forest ($R_{L'}$)

IA = y[OB]

k = 0

for $i = 0$ **to** N_i **do**
  for $j = 0$ **to** N_j **do**
    if AI_i=E_{ij} **then**
      $rank_k$=j
      $CH_{L'}$ = OB_i
      k=k+1
    end
  end
end

ranks=set(rank)

for $k = 0$ **to** N **do**
  Print($ranks_k$)
end

return $\mathbf{CH}_{L'}$

4.3.5 Synthesis

Our recognition approach can be summarised in figure 4.3.6:
We start by filtering out deficiencies from event data using PM4PY. In this order, we obtain cleaned events (without noise, infrequent and incomplete behaviours). Then, we proceed to select the data frame on which we will apply machine learning algorithms. We target cases and their corresponding sequence of activities. This consists of converting events from XES format to CSV format. Afterward, we focus on extracting valuable features and discard irrelevant information from the original data set. This is applied in the objective of unifying data and encoding qualitative events to quantitative ones. Thereafter, we reduce the event data complexity. This allows acquiring simple data. Last, we apply the IF algorithm on simplified data for detecting chaotic activities. These activities will be filtered out from events, in the next section, to evaluate process discovery results, conformance checking observations and bring improvement scenarios.

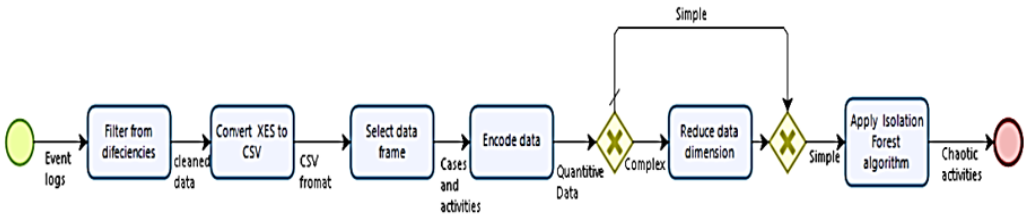


Figure 4.3.6: Process of revealing chaotic activities from event data

4.4 Experiments

In this section, we experiment our recognition approach. Beyond, we demonstrate the ability of our approach in adjusting process models that can be generated with process discovery techniques. In this regard, we aim to reveal chaotic activities, to filter them out of event data and assess the quality of discovered process models.

To that end, we discuss three case studies, which are grouped in two categories. In the first category, we artificially insert chaotic activities and check whether our recognition algorithms can detect the inserted ones. In the second category, we apply our recognition algorithm on event data, in the same objective, without having certainty on which activities are exactly chaotic. In this sense, we present the following case studies: loan Process (Section 4.4.1), Sepsis event (Section 4.4.2) and the purchase order handling process (Section 4.4.3). We end this section by evaluating our method performances, relatively to the process mining techniques (process discovery algorithms, conformance checking and enhancement) in section 4.4.4.

4.4.1 Loan Process

The loan process is a collection of artificial event logs that represent example of financial loan application belonging to a sample bank. The event log is containing a total of 100 event logs. We artificially insert a chaotic activity, and we try to check the applicability of our proposed approach in recognizing the inserted chaotic activity. In this sense, we insert the "random_chaotic" activity that takes different positions (rank from 0 to 7), as mentioned in table 4.1.

For the loan process, we have 8 activities. We encode them with numeric values, from 0 to 7 (Cf. Table 4.1). This is done, to allow machine learning algorithms' application. In this context, we represent the loan encoded version in Figure 4.4.1. Here, each activity is mentioned with specific colour (see part a of Figure 4.4.1).

After applying our recognition algorithm (algorithm 2), we obtain the graph mentioned in part b of figure 4.4.1. Here, we observe that the chaotic activity is the encoded one as 0, because it is executed arbitrarily in the loan process and takes different positions' values (ranks: 5, 2, 1, 6, 3, 4, 0) during the loan process execution. Thus, the chaotic activity and its corresponding cases are represented with the following parameters: Cases array([0., 1., 12., 78., 96., 97., 99.]) Activity

array([0., 0., 0., 0., 0., 0., 0.]).

Table 4.1: Excerpt of loan process Event logs

Case	Activities	Encoding	Rank
Case_0	<register application, check credit, calculate capacity, check system, accept, send decision e–mail, **random_chaotic**>	<5, 3, 2, 4, 1, 7, **0**>	6
Case_1	< register application, check credit, **random_chaotic**, calculate capacity, check system, accept, send decision e–mail>	< 5, 3, **0**, 2, 4, 1, 7 >	2
Case_2	< register application, **random_chaotic**, check credit, calculate capacity, check system, accept, send decision e–mail >	< 5, **0**, 3, 2, 4, 1, 7 >	1
Case_3	< register application, calculate capacity, check credit, reject, send decision e–mail, **random_chaotic**>	< 5, 2, 4, 6, 7, **0**>	5
...	<...	...	...
Case_97	< register application, calculate capacity, check system, **random_chaotic**, check credit, reject, send decision e–mail>	< 5, 2, 4, **0**, 3, 6, 7 >	3
Case_98	< register application, check credit, calculate capacity, check system, **random_chaotic**, accept, send decision e–mail >	< 5, 2, 4, 3, **0**, 6, 7>	4

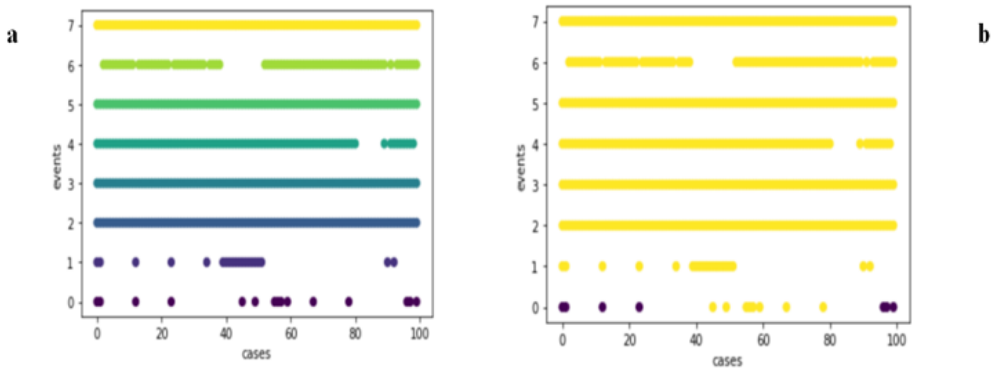


Figure 4.4.1: Recognizing chaotic activities (b) for the loan process event logs (a)

4.4.2 Sepsis Event logs

For the experiments on real-life event logs, we do not artificially insert chaotic activities, but we apply our recognition approach phases on original event logs files. These logs contain chaotic activities that are not known upfront.

Table 4.2: Excerpt of Sepsis event logs

Case	Activities	Encoding	Rank
...	...	...	...
Case_58	< CRP, **Leucocytes**, ER Registration, ER triage, **Leucocytes**, CRP, ER Sepsis Triage, IV Liquid, IV Antibiotics >	<2, **4**, 0, 1, **4**, 2, 6, 7, 8>	1, 4
Case_104	< ER Registration, ER triage, ER Sepsis Triage, **Leucocytes**, LacticAcid, CRP, IV Liquid, IV Antibiotics, Admission NC, CRP, **Leucocytes**, LacticAcid, **Leucocytes**, CRP, Release A, ... >	<0, 1, 5, **4**, 3, 2, 6, 7, 8, 2, **4**, 3, **4**...>	3, 10, 12
Case_207	< ER Registration, ER Sepsis Triage, CRP, **Leucocytes**, LacticAcid, Admission IC, Admission NC, CRP, **Leucocytes**, CRP, Admission NC, Release B >	<0, 5, 2, **4**, 3, 10, 8, 2, **4**, 2, 11>	3, 8
Case_346	<ER Registration, ER triage, ER Sepsis Triage, LacticAcid, CRP, **Leucocytes**, IV Antibiotics, Admission NC, Admission NC, CRP, **Leucocytes**, CRP, Release D, Return ER, ... >	<0, 1, 5, 3, 2, **4**, 7, 8, 8, 2, **4**, 2...>	5, 10
Case_719	< ER Registration, ER triage, ER Sepsis Triage, CRP, CRP, **Leucocytes**, Admission NC, CRP, **Leucocytes**, **Leucocytes**, CRP, Release E >	<0, 1, 5, 2, 2, **4**, 8, 2, **4**, **4**, 2, 15>	5, 8, 9
...	...	...	...

In this case study, we treat the sepsis event logs . These events are conducted in a regional hospital in The Netherlands. The regional hospital has about 700 beds at several locations and is visited by about 50,000 patients per year. They are relating to 16 activities. There are about 1000 cases within total 15,000 events that were recorded with 16 different activities.

This test helps in determining the group of patients could be affected with the sepsis disease. Sepsis is a life-threatening condition that arises when the body's response to infection causes injury to

its own tissues and organs: We focused on patients that do the Leucocytes test. Here, leucocytes (White blood cells) are the cells of the immune system that are involved in protecting the body against both infectious disease and foreign invaders.
This process' events are complex. Therefore, we use the PCA algorithm after the encode function (based on the algorithm 1). This helps in obtaining numeric variables and reducing data complexity.
In this example, we illustrate an excerpt of sepsis event logs (see Table 4.2). The most long executed path has 185 activity executions (0-184). The leucocytes activity takes places 0, 1, 2, 3, 4,...,184, i.e., takes 185 different places. Therefore, the leucocytes activity appears arbitrarily during the process execution. By definition, we consider the leucocytes activity as a chaotic activity that is executed anywhere.
After applying our recognition algorithm (algorithm 3), we obtain an array with one dimension. This array represents the chaotic activity and its positions: Chaotic activities array([4., 4., 4., 4., 4.,. . . , 4., 4., 4., 4.]).

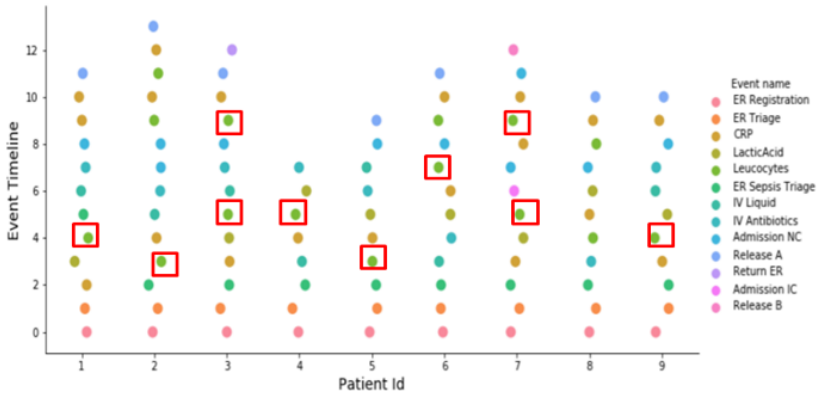


Figure 4.4.2: Patients record.

In the encode function '4' is the numeric value representing the leucocytes activity. In this context, figure 4.4.2 shows the first nine traces of different patients. The leucocytes activity is recorded with different ranks (case1:4, case2:3, case3:5 and 9, case4:5, case5: 4, case6:8, case7:5 and 9, case8:2, case9: 4).

4.4.3 Purchase order handling process

In this case study, we have no background on which of the purchase order handling activities is chaotic or not.
The data source is collected from a large multinational company operating from the Netherlands of coatings and paints. The data source is about the purchase order handling process for some of this company 60 subsidiaries.

Table 4.3: Excerpt of the purchase order handling event logs

Case	Activities	Encoding	Rank
Case_1	<SRM: Created, SRM: Complete, SRM: Awaiting Approval, SRM: Document Completed, SRM: In Transfer to Execution Syst., SRM: Ordered, SRM: Change was Transmitted, Create Purchase Order Item, **Vendor creates invoice**, Record Goods Receipt, Record Invoice Receipt, Clear invoice >	<0, 1, 2, 3, 4, 5, 6, 7, **8**, 9, 10, 11>	8
Case_10	< **Vendor creates invoice**, SRM: Created, SRM: Complete, SRM: Awaiting Approval, SRM: Document Completed, SRM: In Transfer to Execution Syst., SRM: Ordered, Create Purchase Order Item, ... >	< **8**, 0, 1, 2, . . . >	0
Case_27	< Create Purchase Order Item, Receive Order Confirmation, Record Goods Receipt, Record Invoice Receipt, **Vendor creates invoice**, Clear invoice >	< 7, 24, 9, 10, **8**, ... >	4
Case_46	<Create Purchase Order Item, Change Quantity, Record Goods Receipt, **Vendor creates invoice**, Record Invoice Receipt, Clear invoice >	< 7, 20, 9, **8**, ...>	3
Case_71	< Create Purchase Order Item, Record Goods Receipt, **Vendor creates invoice**, Record Invoice Receipt, Clear invoice >	< 7, 9, **8**, ...>	2
Case_71	< Change Approval for Purchase Order, **Vendor creates invoice**, Change Approval for Purchase Order, Create Purchase Order Item, Change Approval for Purchase Order, Record Invoice Receipt, Clear invoice >	< 33, **8**, ... >	1
...	...	...	...

In the data, each purchase order contains one or more items and for each item, there are four main types of flows: (3 ways for matching invoice after goods receipt. 3 ways for matching invoice before goods receipt. 2 ways for matching (no goods receipt needed).

The complexity of the data appeared in the volume of 1,595,923 events relating to 42 activities performed by 627 users. The data contains many purchases, goods receipt messages, and corre-

sponding invoices. Consider for example the process of paying rent. There is a purchase document with one item for paying rent, but a total of 12 goods receipt messages with invoices.
The data refers to many different categories of goods and services and include many different types of vendors. We treat data with [1595923 rows x 2 columns]. We focus on data representing vendor creation of invoices activities. Each line of our matrix represents one trace. To apply machine learning algorithms, we encode the purchase order handling event data. Due to the number of events, we use the PCA algorithm as developed in the algorithm 1, to reduce their complexity. In this example, the longest executed path has 923 activity executions (0–922). It takes 923 different places. Each activity is demonstrated with a specific colour degree (see part a of Figure 4.4.3). As example, we illustrate in table 4.3 an excerpt of cases execution. The "vendor creates invoice" activity takes places 0, 1, 2, 3, 4,..., 922. We can say that the "vendor creates invoice" activity represents a chaotic activity. The vendor creates invoice activity is appeared anywhere in the recorded event logs.

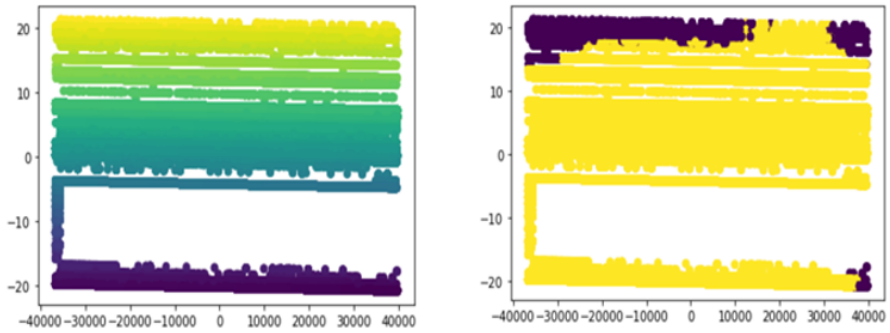


Figure 4.4.3: Recognizing chaotic activities (b) from reduced event data (a) of the purchase order handling challenge.

After applying our recognition algorithm (algorithm 3), we get the following parameters: Chaotic activities Array ([-20.88795379, -20.88795379, -20.88795379... 18.88877908, -20.890254, -20.89 024031]). This can be observed in part b of figure 4.4.3, where chaotic activities are highlighted with the purple colour. This graph shows chaotic activities in event data with reduced dimension.

4.4.4 Results Analysis

At this stage, we discuss results of two different stats: before and after filtering out chaotic activities from events. This is defined based on our recognition approach that aims at detecting chaotic activities, using unsupervised machine learning algorithms. This is an advantage in terms of skipping variable's labels and focusing on the less knowledge of event data.
After filtering out the detected chaotic activities, we try to assess the following quality of discovered process models, which are resulted from the three above case studies. To do so, we use the process model quality metrics: fitness, precision, generalization, and simplicity. Except the LOAN events, the other events are considered as real data that belong to different fields, from healthcare to insurance.
In table 4.4, we compare the four-quality metrics of two different stats related to three process

models, after and before applying the filtering chaotic activities operation. In this context, we use the inductive miner algorithm, to represent process models. This algorithm is one of the few mining algorithms that actually guarantee sound process models. It produces a sound model that does not contain any anomalies and replay most logs. Also, it can detect the starting and ending points of the process and ignore the least important arcs. Comparing to process models generated before the filtering operation, we observe that the fitness value is excellent for the Loan, the Sepsis and the purchase order handling models (more than 90%) after filtering out chaotic activities. In addition, they have high Generalization and Precision values. Therefore, table 4.4 results assure the applicability of our proposed approach in improving process models quality.
For the conformance checking application, we can provide more diagnosis information. This can help in deducing deviations in terms of process models perspectives like timestamp, resources, control-flow, etc. For instance, we can diagnose resources responsible for executing chaotic activities, delays during its executions and the control-flow deficiencies.
To conclude, business processes can be improved using: 1) New sub-set of event data, obtained after filtering out chaotic activities and 2) Information about nonconformities. To enhance discovered models based on the two points.

Table 4.4: Process discovery quality metrics before and after applying our recognition chaotic activities algorithm

	Loan Proecss		Sepsis Case		Purchase order handling challenge	
	Before	After	Before	After	Before	After
Fitness	72%	**93%**	80%	**91%**	77%	**89%**
Precision	63%	**89%**	79%	**88%**	65%	**89%**
Generalization	66%	**87%**	75%	**88%**	58%	**89%**
Simplicity	Relatively No	Yes	Relatively No	Yes	Relatively No	Yes

4.5 Summary

In this chapter, we propose an approach for recognizing chaotic activities from event data (4.5.1). Chaotic activities are executed arbitrarily in the business process. They impact the quality of process discovery results and their process of improvement; especially, in the case of no ground truth knowledge on which activities are exactly chaotic. In this regard, our approach uses unsupervised machine learning algorithms and encompasses the following four phases:
The first phase aims at cleaning event data from well-known deficiencies (Noise, Incompleteness, infrequent). This is done by using PROM plug-ins and the PM4PY library. PM4PY integrates process mining techniques into machines learning frameworks. The second phase consists of two steps. The first step target is to convert XES data to CSV, to make them easy in case of applying machine learning algorithms, which are focusing on arrays format. The third phase aims at defining appropriated features from data. To do so, we encode qualitative data to quantitative ones. This is required for reducing the event data complexity, using linear algebra of PCA. The

fourth, and the last phase consists of revealing chaotic activities based on the IF algorithm. These activities can be filtered out, to evaluate the quality of discovered models using fitness, precision, generalization, and simplicity metrics.
After filtering out chaotic activities from event logs [17], we can obtain structured (Chapter 5) or unstructured business processes (Chapter 6). In this sense, more challenges can be treated.

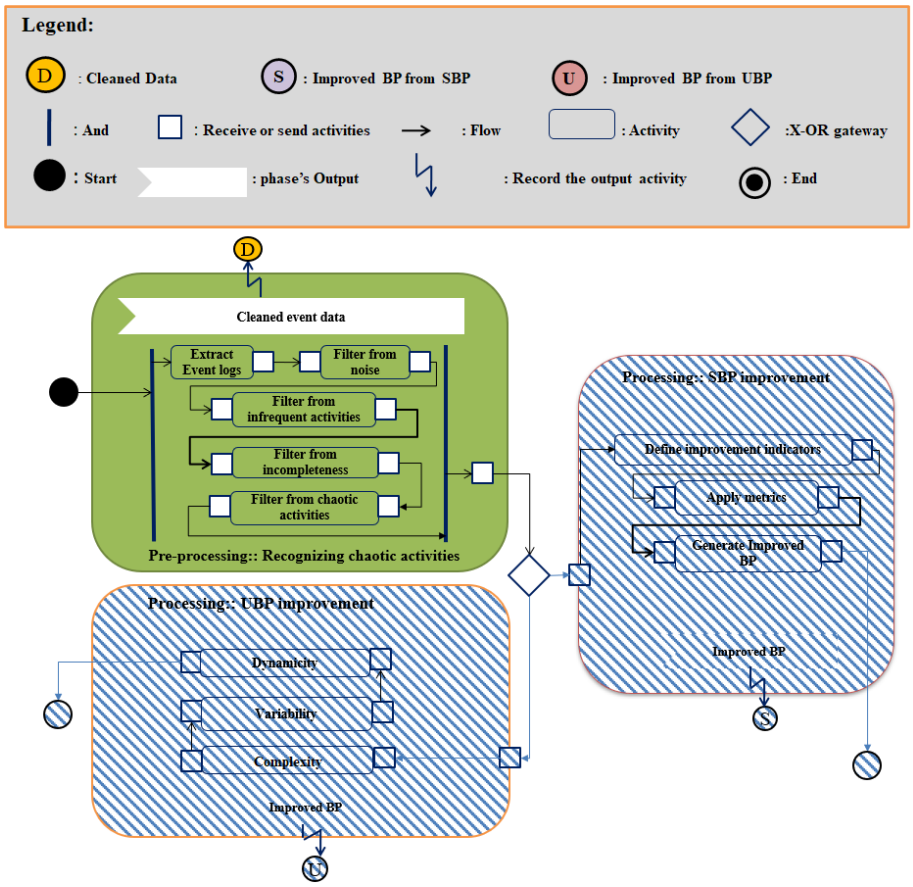


Figure 4.5.1: Pre-processing: Recognizing chaotic activities

Chapter 5

Processing: Structured BP Improvement

This chapter presents an approach that aims at defining a set of improvement indicators for each phase of the BPM life-cycle, in order to improve BPs and maintain their simplified structure. Indeed, the chapter is organised as follows: Section 5.1 introduces our contribution that consists of dealing with the maintainability of the BP simplified structure. Section 5.2 defines improvement metrics during the whole BPM life-cycle. Besides, an overview of our approach is presented. Section 5.3 checks the applicability of our approach. Section 5.4 concludes the chapter and gives summary of the main contribution.

5.1 Background

As a result of the massive development induced by information and communication technologies, organizations are subject to fast changes in all aspects such as technical, organizational, and operational aspects. These changes need systems and procedures to increase the efficiency and performance of BPs. In this order, BPM makes an incremental improvement for BPs, through several methodologies, techniques, and tools, and thus to ensure organization competitive position in the market.

According to the literature review representation in chapter 3, there is currently no integrated method to evaluate BP improvement during the whole BPM life-cycle. Due to, a challenging problem is emerged for companies that are in need to improve continually their BPs. Indeed, it is required to find an objectively way to check the BP improvement during the whole BPM life-cycle phases, in terms of what can be optimized and how can be improved with each passage from one phase to another. Therefore, we propose an integrated approach to fill this gap by defining a set of improvement indicators for each BPM life-cycle phase.

Generally, BPs are characterized by three performance metrics: **time** which stands for the duration that takes the process to run from the start to the end; **cost** that implies the expense of the running BP; **quality** which refers to the disparity level between the projected outcome of the process (product or service) and its actual outcome. Note that metrics of time, cost and quality go hand-by-hand

to improve BPs. Acting on one metric and in one phase cannot improve integrally BPs. Consequently, it will be difficult to guarantee the BP simplified structure (opposite to unstructured BP). In this sense, improving BPs and maintaining acceptable levels of performance are main factors in the success of any organization. Related to BPM approaches, a variety of methodologies and techniques are available. These approaches include BPI methodologies and process mining techniques. Actually, with the BPM life-cycle the improvement operation is only applied on the monitor and the optimize phases.
Therefore, this chapter presents a new vision of the BPM life-cycle whose content differs from the classical one. On each phase, we define a set of improvement indicators. The indicators recast BPs for the best optimization result.

5.2 BPM phases and Improvement Metrics

To improve BPs, we are inspired by the BPM methodology [68]. This methodology involves changes in existing systems, teams, or processes. In this regard, identify the relevant business of a company is the first step, in our new vision, to discover the "as-is" process (defines the current state of the BP in an organization.). Then, the application of the design metrics can generate the "to-be" process (defines the future state of a BP in an organization). In turn, it can be analysed in the second phase, using the analysis metrics to determine suitable execution metrics for the third phase. Next, an executable process is monitored by a set of KPIs (Key Performance Indicators) in the fourth phase. Last, the monitored process must be optimized using the process mining techniques, as enhancement metrics, for the BP improvement.

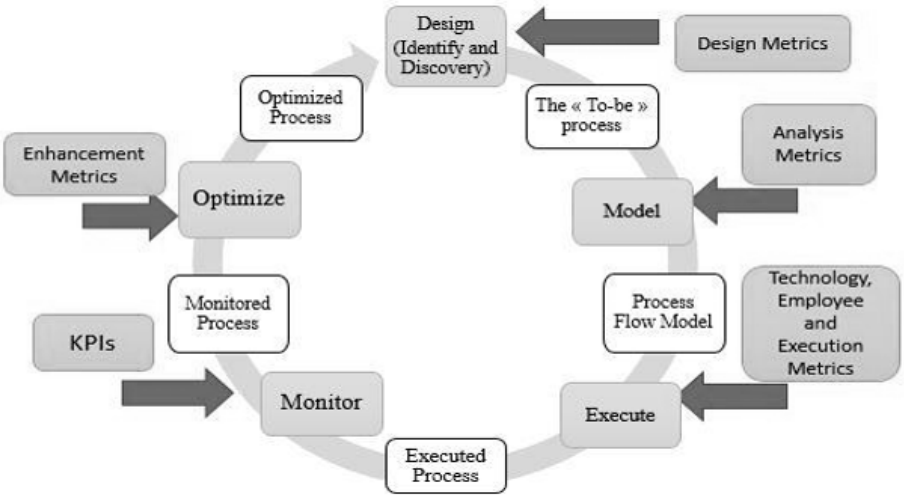


Figure 5.2.1: BPM life-cycle metrics

The advantage of our vision over the classical one (define metrics as KPIs only in the monitor phase) revolves around the improvement of BPs based on the definition of different improvement indicators. These indicators are grouped into metrics (see Figure 5.2.1) that can be applied in each

phase of the BPM life-cycle, to obtain an optimized BP [19]. Indeed, in the following subsections, we will present each phase according to the appropriate metrics.

5.2.1 Design metrics

Design metrics are defined, during the BPM life-cycle, in order to obtain a specific and unique output in each phase.
In this phase, we are faced to two parts: Identify the Business of the company and Discover the "as-is" process. We define the set of indicators related to the design phase according to a specific logic. We start by defining the Business of the company (**Relevance**), that leads to check the syntax of the process (**Correctness**). Once this Correctness is verified, it becomes easy to understand the language that represents the process (**Clarity**). Based on the clarity of the process, we proceed to the process implementation (**Completeness**). Then, we represent it according to the company business context (**Consistency**) to reach the "to-be" process. Beyond this logic, we propose the following order of priority (Relevance – Correctness — Clarity – Completeness – Consistency). The design metrics allow to obtain a good decision diagram: Have correct execution semantics; lay out a clear structure that employs sub-decisions if necessary; provide a complete mapping of all possible input or output combinations; employ data objects that are consistent with the real-life event data (see Figures 5.2.2 and 5.2.3).

Figure 5.2.2: Social Process Model

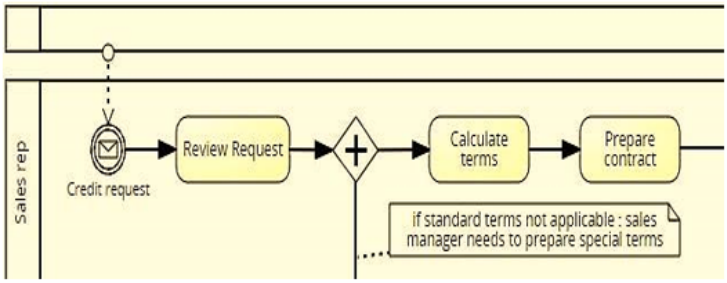


Figure 5.2.3: Deviation in BPMN Diagram

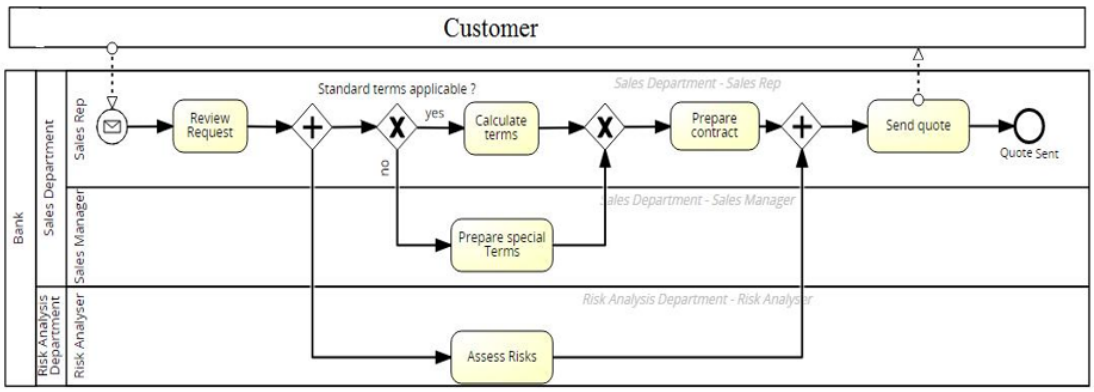


Figure 5.2.4: Applying Correctness, Clarity & Completeness

5.2.2 Analysis metrics

In this phase, we define a set of indicators that are nominated as analysis metrics. These indicators can be applied on the "to-be" process, which is retained from the previous phase of the BPM life-cycle. Also, they conduce to obtain the "process flow model". The outcome of this phase is a pseudo-executable process flow model that can be exported as an executable language format supported by the BPM engine, such as ORACLE BPM, Orchestry, etc. For instance, the mapping modelling language as BPMN can be achieved to a process execution language as BPEL (Business Process Execution Language).
These analysis metrics are based on the logic combined with the order of need for each part of the formulas to another. For instance, we must calculate previously the cycle time to obtain the throughput rate. Indeed, values are comparable to the company's business objectives and verifiable to the outcome process criteria: (if the BP: 1. Valid, 2. Efficient, 3. Effective, 4. Economic, 5. Usable). These criteria are treated according to each indicator requirements. In the literature, we find the following analysis metrics [77, 82, 124–126]:

Operation time: Setup time (the length of time required to changeover from one process or activity to another) + Run Time (is when a process is running (or being executable).

Throughput time: Average time for a unit to move through the system or the amount of time a flow unit spends in a BP from beginning to end. If there is more than one path through the process, the flow time is equivalent to the length of the longest path.

Cycle Time: is the average time between completions of successive units.

Throughput rate: The number of flow units, e.g., customers, money, produced goods or services, going through the BP per unit time, e.g., served customers per hour or produced parts per minute. The flow rate usually is an average rate.

Process Cycle Efficiency: This is a measure of how much waste is present in our process.

Productivity: the ratio between the amount produced and number of resources used in production. Units of Input means resources utilised, and Units of output means anything generated from production(Land: Hectares, Metric: ton, Machinery: machine hours and People: Man Hours, etc.).

Utilization: is the ratio of the time when a resource is activated related to the time when it is available for use.

Capacity: maximum output of a process or resource measured in units or time (a rate).

Quality: Numbers of mistakes in result. Proposition of reworking correction errors.

These analysis metrics allow to evaluate the BPs of a company before their final use or employment. This is done using the simulation technique. Simulation is a method for visualizing and validating the behaviour of a BP diagram with visual indications of all currently simulating activities along with the possible activities that can be executed next. Here, we will easily be able to understand and identify potential issues with the process that we have modelled. This can define several scenarios. For instance, after applying the analysis metrics on the "to-be" process, we may acquire a change in the process model representation, by default related to the resource repartition.

5.2.3 Execution metrics

In this phase, and after the simulation step, we arrive at the execution of the process flow model at runtime. Therefore, we define the Execution metrics. These metrics can be applied in the current system to get an executable process [78, 79, 84, 125]: 1. Measured and 2. Managed.

Execution Metrics: Perform the correspondence between the suitable technology, the Time spent (references & frequencies), the most optimal cost and quality according to the BP at hand.

For both metrics, to obtain an executable process, the most suitable technology must be used, in which this process should be implemented, as well as the execution metrics allow it to be executed in good conditions according to the company objective.

-Technology metrics: The order is giving from the global to specific parts (System –Network– Support application).

-Uptime: The time that a system is "running", and downtime is the inverse of that. Thus, the key indicator to follow it is "availability", which is a measurement of whether a system's, application's or network's target users can access and normally use a system. A system can be "up" but unavailable. Therefore, to define SLAs (Service Level Agreements) or "working agreements" to a specific business, it is required to focus on the availability guarantees indicators and not just on the simple uptime.

-Time Reference (for current or former instances): Through time (min, max, average). Cycle time (min, max, average). Processing time (duration of process step) of all subject carriers (min, max, average).

-Running instances: Number of open process instances per process type in which a subject carrier is involved. Start time of instances per type of process (distribution). Number of instances, which are in a process step. Number of instances that have exceeded a certain period a particular transition).

-Completed instances: Number of process instances per process type and time unit. Number of process instances for which the throughput time is above average. Number of subject changes of

a subject carrier per time unit.

-Cost: Total cost of the process. Cost of a subject carrier in the process. Propositions of subject carriers to the total cost of the process.

-Quality: Numbers of mistakes in result. Proposition of reworking correction errors.

In the fact, the process flow model is executed, and the obtained values are compared with the SLAs and the company business objectives. In the case of a discrepancy between these values, some changes can be suggested to the system's properties or the process model.

5.2.4 Monitoring metrics

After the process has been running for a while, in the execution phase, and all traces have been collected, we periodically diagnose these information to obtain a monitored process model: 1. Used and 2. Reused.
According to [127], the most studied outcomes in the manufacturing field can be defined by their quality, cost, flexibility, and delivery. In this sense, the performance outcomes of each BP can be characterized by reduced cost, reduced time, improved quality, improved work life quality and enhanced learning and empowerment. Also, the contribution in [128] cites eight dimensions of features performance as reliability, conformance, durability, serviceability, and perceived quality. Moreover, five attributes of quality (service as reliability, responsiveness, assurance, empathy, and tangibles) are defined in [129]. Therefore, performance' measures should include how well process is working, how is the process influencing the company's environment to handle every change. Thus, these measures can be defined based on three factors: efficiency, effectiveness, and flexibility.
Therefore, we group the KPIs (monitoring metrics) into process efficiency, process effectiveness and process adaptability. Moreover, we define these KPIs based on the logic combined with the following indicators (calculate time, cost, utility of the BP which leads to check if it is flexible relatively to the user satisfaction).

-Efficiency: The performance measures associated with the process with measures related to the process itself.

-Cost: Material, Equipment, Opportunity cost, Cost of poor quality, Cost added only (rework, waste, duplication).

-Time: Cycle Time, Task Time, Up Time, Down Time, Delay Time.

-Utility: Personal Turnover, System uptime, Network uptime, Support application uptime.

-Adaptability: The ability of a process to be adapted according to different changes.

-Flexibility: Durability, Responsiveness, Timeliness.

-Effectiveness: The performance measures associated with the process with respect to the change induced to the environment. A business process is a set of activities performed by one or more people with the aim of achieving business goals. A company must effective BPs. An effective process is a process that produces the right results consistently. One of the best business process

improvement examples is the continual improvement of process productivity.

-Satisfaction: Requirement of stakeholders are met, Rating of service/product by importance and satisfaction, Employee engagement rating, New customers attracted, Retention rate of current costumers/employees, Customers/ employee complaints.

-Quality: Accuracy, Uniformity, Reliability, Deliverability.

After applying the KPIs, we can reveal out bottlenecks (delay, dissatisfaction) from the obtained BP. For BP improvement, we suggest two scenarios. We may modify the process flow model to adapt these metrics according to bottlenecks that are occurred in the monitor phase. Then, we proceed to the next phase, or we directly skip it to the process mining phase, in order to redesign the process flow model, and to adjust metrics, as a solution of the revealed ambiguities.

5.2.5 Enhancement metrics

In this phase, we focus on how to implement process mining (PM) techniques in our BPM life-cycle.
After the process has been running for a while in the execute phase and the "event logs" collection, we apply "process mining" techniques that consist of a set of analysis techniques that can be used to understand the actual behaviour and the BPs performance by studying recorded events (also known as event logs) based on the following indicators [1, 3, 130]: Quality Criteria, Control flow perspective, organizational perspective, Case perspective and time perspective. To end up with an optimized process that must be: 1. Adaptable, 2. Efficient, and 3. Effective.

The five types of metrics used in the process mining architecture are nominated in this study as "enhancement metrics", these metrics are classified according to the following logic (Discovery – Conformance – Enhancement):

- Quality Criteria: Fitness, Precision, Generalization and Simplicity.

- The control-flow perspective: focuses on the control, i.e., the ordering of activities.

-The organizational perspective: focuses on information about resources hidden in the log (e.g., people, systems, roles and departments).

- The case perspective focuses: on properties of cases, e.g., cases can also be characterized by the values of the corresponding data elements.

- The time perspectives: It is concerned with the timing.

5.2.6 Overview of Our Improved BPM Approach

In the above section, we have classified a set of improvement indicators during the whole BPM life-cycle, in order to obtain an optimized BP. First of all, we have proposed a new vision of the BPM life-cycle with the following elements: indicators, outcome process and outcome process criteria. The indicators present different measures that are attributed to each phase in order to improve the BPs during its passage from one phase to another: Design metrics, analysis metrics, execution metrics, monitoring metrics and enhancement metrics. The phases encompass all the

different phases of our BPM life-cycle: Design, Model, Execute, Monitor and optimize. Outcome process shows the outcome of each phase which is represented in the form of a specific process: "as-is" process, "to-be" process, process flow model, executable process, monitored process, optimized process. The outcome process criteria illustrate a set of criteria that must be checked on each outcome process, at the end of each phase: Valid, relevant, correct, clear, complete, effective, efficient, usable. Secondly, we have distinguished two BPM life-cycle versions (see Figure 5.2.5): Normal & conditional. The original BPM life-cycle is combined with the logic of our new vision by adding: metrics, outcomes processes and outcome processes criteria. The normal is determined by the following path: –Apply the design metrics – Apply the analysis metrics – Apply execution & technology metrics –Apply KPIs –Apply enhancement metrics. While the conditional version summaries how improvement indicators can affect the BP in specific cases.
At the end, we have presented and detailed these improvement indicators that will lead us to improve BPs. Moreover, we have defined different outcome process criteria to obtain an optimized BP.
In the case study section, we will detail our improved BPM approach phases and we will verify its applicability.

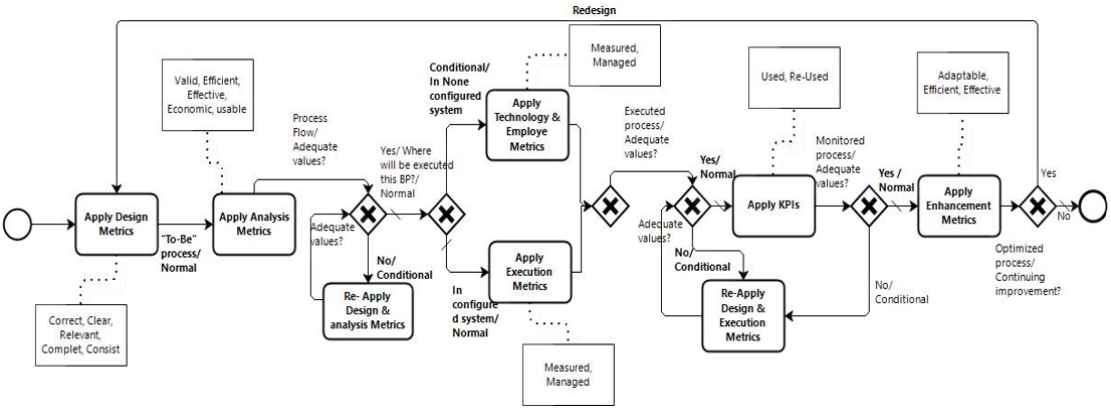


Figure 5.2.5: Our improvement BPM approach scenarios

5.3 Illustrative case study

In this section, we present a concrete case study about an airline company to verify the applicability of our proposed approach, starting from the definition of the company BP objectives. In our case, the BP type is operational. This BP is about a job of handling requests for compensations within an airline company.

5.3.1 Design phase

We start from the company Business definition (Relevance). This will lead us to check the process syntax (Correctness). Once, the Correctness is verified, it become easy to understand the main

process representation (Clarity). Then, we can proceed to the process implementation (Completeness). Also, we must represent it as much as reality (Consistency), which helps in the "to–be" generation. All in all, these four indicators are denoted as design metrics [131] that emerges the following elements:

a- Social Process Model: We use the social process model (see Figure 5.2.2) to define the company business details (Relevance). According to the social process model [132], tree points must be investigated (Impacts, Organization and Assumptions) to determine the main objective of the studied business process.

b- "As-is" process: After the BP definition, we represent its "As-is" process. This process begins with the Register request activity for compensation. An administrative checks to see whether the customer is eligible to issue a request. Thus, the decision is accrued. Then, we can model three possible outcomes: 1- The requested compensation is paid, 2- The request is declined, 3- further processing is needed. Last, the process ends up by the compensation payment or the request rejection. In this example, there is a getaway with one choice. In the process mining context, open decision points can be modelled for future extensions. These points are qualified related to the company's objective. These points are selected (see Figure 5.3.1).

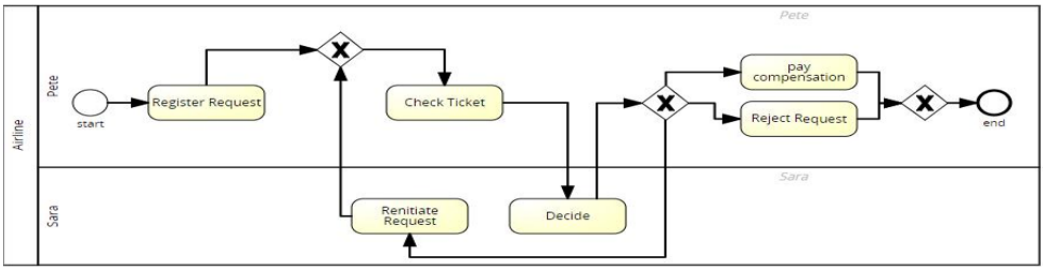


Figure 5.3.1: "As-is" process (handling a request for compensation)

c- "To-be" Process: Once the "as-is" process is created, the "to-be" process model can be derived by projecting improvements or changes, need to be made, on the existing business process. In this regard, two types of additional examination (thoroughly and casually) are added to verify if the register request has been approved. There is a choice between these two activities. The first transition examines thoroughly suspicious or complex requests. The second Straightforward focuses on casual examination. Indeed, we must take into consideration these two ways to improve the current process to better integrate new activities and resources ("To-be" process). Therefore, it is required to insert (see Figure 5.3.2):

- New resource with new activities: Sean [1] the responsible for examining requests.
- Two gateways (Exclusive & Parallel) between the registration activity and the examination activity.
- Two activities that follow the second gateway. The first one denoted as "examine thoroughly" (if suspicious or complex). The second one expressed as casually (if the request is repeated).
- A new gateway between the checked activity and the decide activity must be added.

[1] Resource

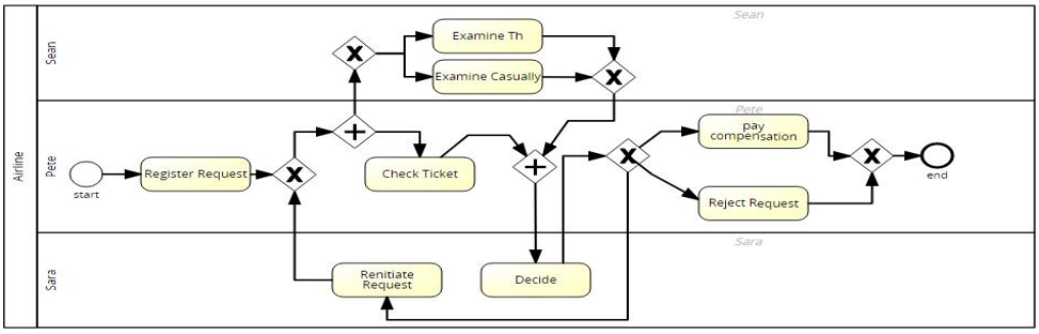


Figure 5.3.2: "To-be" process (handling a request for compensation)

d- Synthesis: The improvement metrics, introduced in this phase, help in defining a diagram of decision with correct execution semantics, lay out a clear structure that employs sub-decisions if necessary, provide complete mapping of all possible input or output combinations and employ data objects that are consistent with the real-life data.

5.3.2 Model phase

The analysis metrics are classified based on the logic combined with the order of need for each part of the formulas to another. For instance, we must calculate previously the cycle time to obtain the throughput rate (see Table 5.1). The values are comparable to the company's business objectives and verifiable to the outcome process criteria: (if the BP: 1. Valid, 2. Efficient, 3. Effective, 4. Economic, 5. Usable). According to [124], the authors focus on what would make such a process a performant process model. In this regard, we aim at the end of each BPM life-cycle phase to check if the process achieved these process model conditions. These conditions are presented according to each phase requirements that we will present in the following.

a- Process flow: This topic describes the simulation method for visualizing and validating the behaviour of our business process diagram (see Figure 5.3.3). With visual indications of all currently simulating activities, we can easily be able to understand and identify potential issues in the process we have modelled.

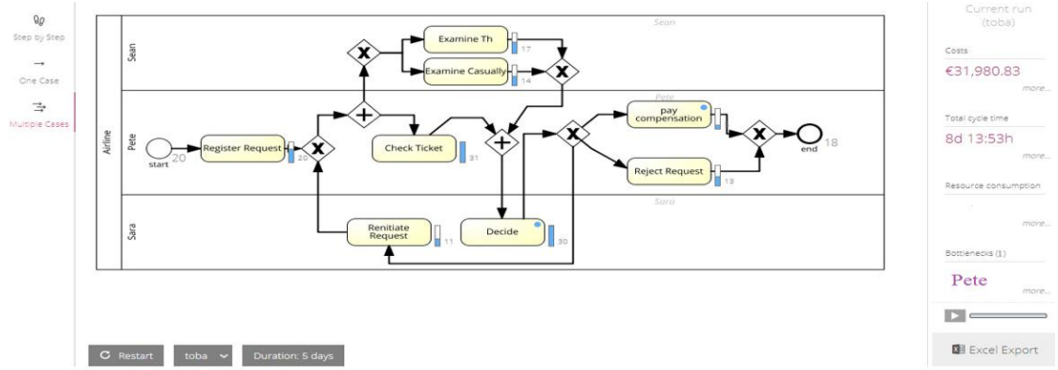


Figure 5.3.3: Screenshot shown 'n cases'

In the table 5.1, we summarize the Signavio [2] report (document that resumes cost, time, bottlenecks for each BP execution during x time, in our case we mentioned five days). We also illustrate the simulation of the verification values, and the company business objectives. Indeed, the table encompasses analysis metrics values throughout five days of simulation.
All cases were programmed in 5 days with a throughput rate of 95% for each one day. The five days must be completed in order to evaluate and guarantee the outcome of the model phase (The process flow model). This is verified by matching new modifications on the "to-be" process and by retesting the simulation values.
According to the Signavio report and after comparing the simulation (The "n case" simulation") values, bottlenecks can be revealed out and different configurations can be edited. Thus, optimization can be determined based on these results. In addition, the obtained values can be compared with the company business objectives (see Table 5.1). In this sense, we can observe a set of non-conformities, where the obtained values are differed negatively from the company objectives values. In addition, we reveal out that the BP is executed slowly. This is referred to the charge attributed to Pete [3]. Moreover, we note that the waiting line for the activity "check ticket" is saturated what caused the slow application. Therefore, there is an intolerant delay in the execution of the "Check Ticket" activity.

Table 5.1: Analysis Metrics Values (Signavio Report)

Analysis Metrics	**The simulation Values**	**Company Objectives**	**The verification Values**
Operation	5d 05:36h (hour)	5d 01: 36h	5d (days)
Cycle time	8d 13:53h	5d	5d 00:36h
Throughput	60%	70%	68%
Efficiency	50%	10%	6%
Productivity	50%	90%	89%
Utilization	2d 17:37h	5d	5d
Cost	31 980,83 €	10 318,83 €	10 320,83 €

In this regard, we propose to attribute the "Check Ticket" activity to the new role Ellen [4], in order to accelerate the execution time. We suggest these modifications on our "to-be" process to end up with a good process flow model (see figures 5.3.4 and 5.3.5).
Once the process model simulated with the proposed modifications, we obtain adequate values related to the company objectives (see Table 5.1). Now, we can proceed to the execution phase.

[2] https://www.signavio.com/
[3] Resource
[4] Resource

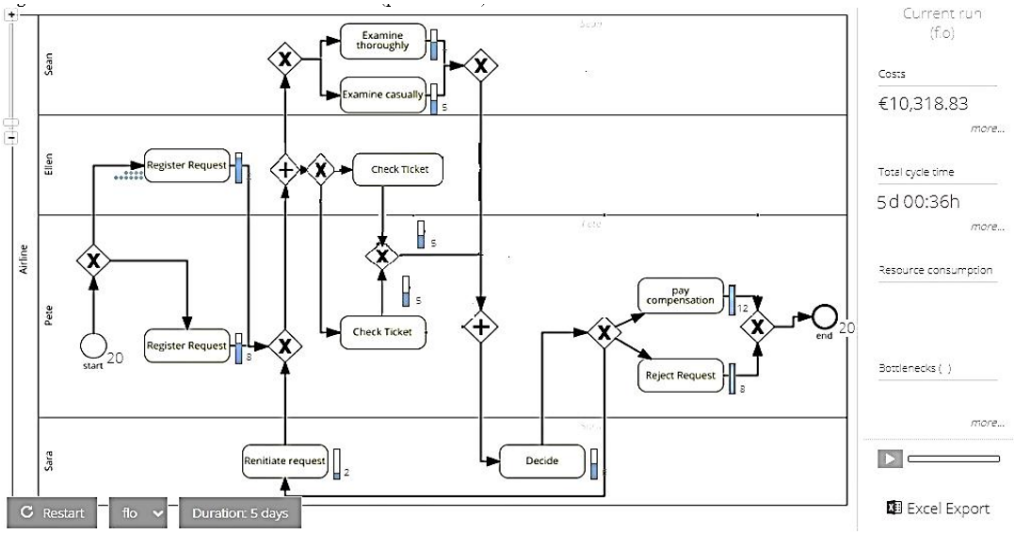


Figure 5.3.4: Screenshot of the verification simulation (process flow)

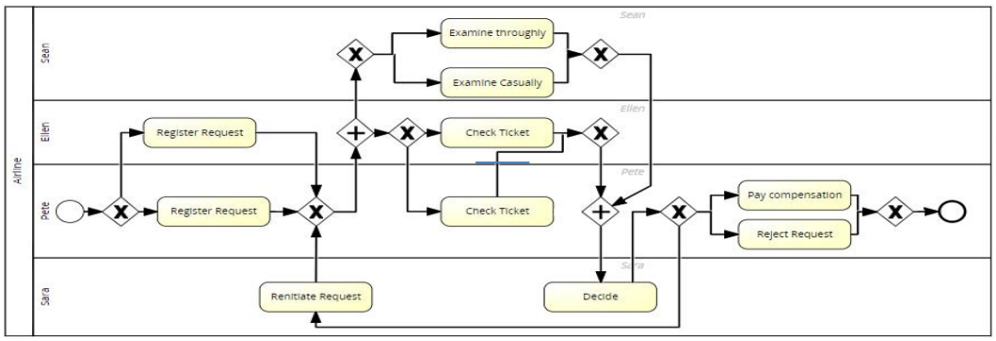


Figure 5.3.5: Process flow (handling a request for compensation)

b– Synthesis: The analysis metrics allow to evaluate the business processes of a company before their final use or employment. After applying the analysis metrics on the "to-be" process at the simulation step, we can easily understand and identify potential issues in the modelled process. Therefore, we can reveal out two scenarios (see Figure5.3.6):

1- If the obtained values are adequate, we proceed to the execute phase (Yes).

2- If not (No) we assume changes in the process model (e.g., the BPMN diagram) and in the attributed resources.

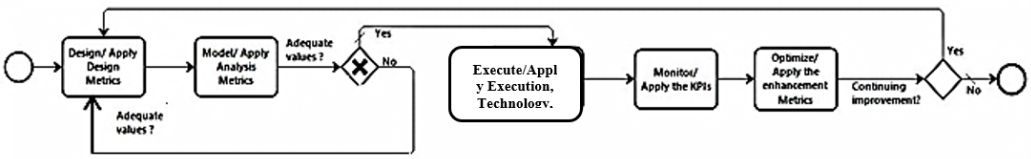


Figure 5.3.6: The model phase's scenarios

5.3.3 Execute phase

We arrive to the execution phase. Here, we apply the execution metrics. At this phase, we aim to obtain a performant executable process model that must be: 1– Measured and 2– Managed.

a- Executable Process Model: For instance, in a configurable system, where all employees (owner, responsible, role, etc.) had definitive activities and all technologies had matched, we need to use the execution metrics to measure the performance of the obtained executable process model.

The execution metrics are organized following this logic: Perform the correspondence between Time spent (references & Frequencies), the most optimal cost and quality according to our business process. Here, the process implementation can change, and the process automation efforts should start once the processes is simulated with positive results. In this context, testing various scenarios with the simulator help ensuring the success of the intended improvements in the studied BP. Also, by given the enormous expense of implementing executable workflow, companies will be sure to apply this step with the highest degree of certainty.

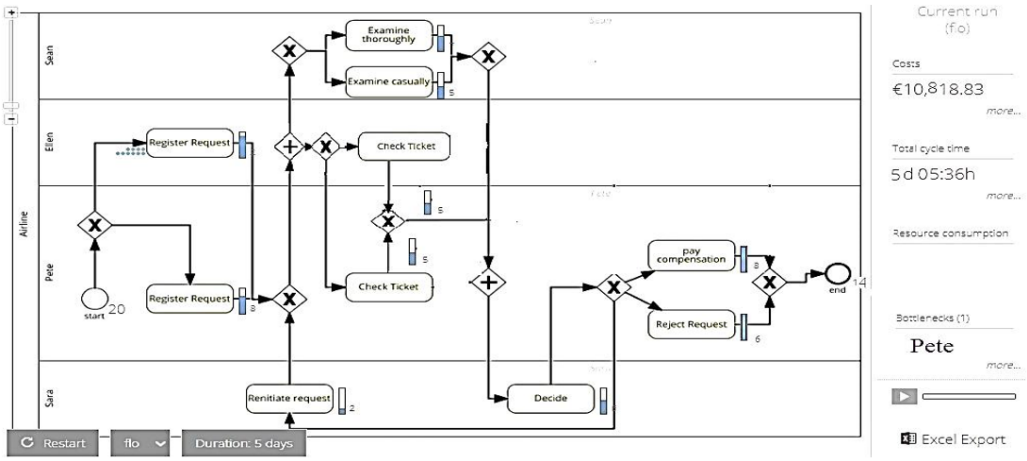


Figure 5.3.7: Execute the process flow

Table 5.2: Execution Metrics Values (Signavio Report)

Analysis Metrics	The simulation Values	Company Objectives	The verification Values
Time reference	5 d 05:36h	5 days	5d 01:36h
Frequencies	60%	70%	68%
Cost	10818,81 €	10318,81 €	10318,81 €
Quality	90%	95%	95%

In the table 5.2, we show the Signavio report with different values: the verification standpoint, the company business objectives and the execution step. Beyond, it represents the execution values throughout 5 days of simulation (see Figure 5.3.7).

We aim to complete all cases, programmed in 5 days with a frequency rate of 70% for each day, in this business process using the execution metrics (Time, cost and quality), in order to evaluate the

execute phase outcome. This can be verified by attributing new modifications in the process flow and by retesting the execution values.

After attributing the Check Ticket activity and the Register request activity to Ellen [5], we obtain optimal results during the simulation step. Therefore, we can implement our model (see Figure 5.3.8).

Later according to the execution results (Given from the Signavio report), we observe deviations between the obtained values and the company objectives. This is referred to delays in the BP execution. Indeed, the Signavio report illustrates bottlenecks that are provided by Pete [6]. Especially, for the Payment and the Rejection activities.

After investigating the Signavio report (see Table 5.2 and see Figure 5.3.7), we note that "Ellen" is the only employee that has completed his job, and he will be available for more time until the new Registration. Therefore, we decide to synchronize these activities between "Pete" and "Ellen" (see Figure 5.3.9).

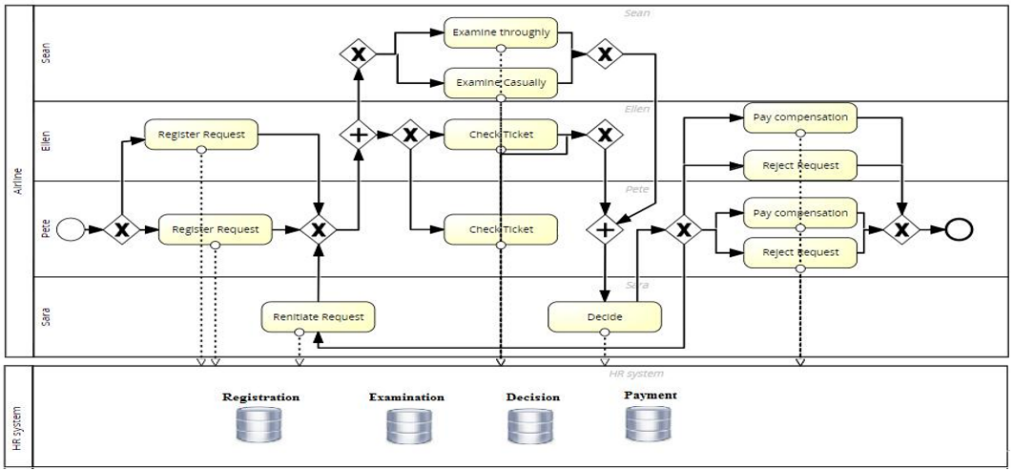


Figure 5.3.8: The executed process

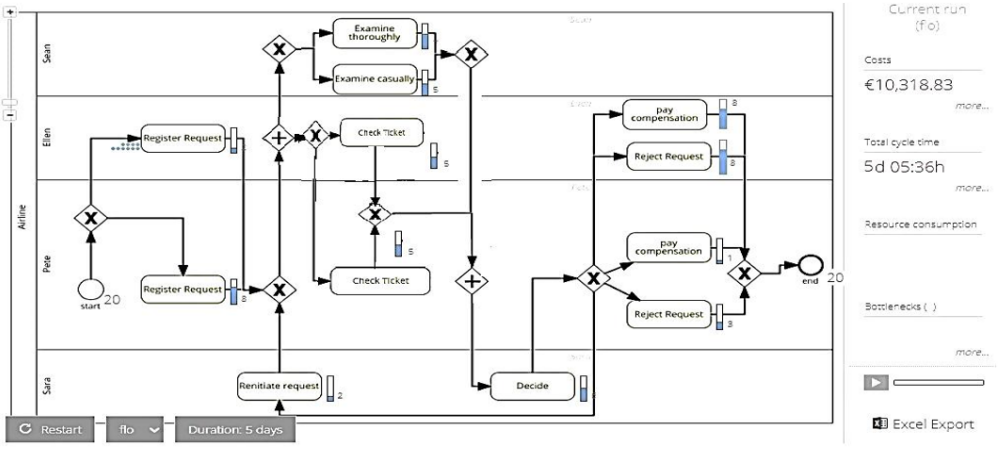


Figure 5.3.9: The verification simulation (executed process)

[5]Resource
[6]Resource

Once, the executable process is established, it will generate suitable data and provide the first opportunity to test the predicted results against the real world (Execute phase). Then, the data are generated, and we can upgrade to the next stage.

b– Synthesis: The process flow model is executed. Then, the obtained values are compared with the SLAs and the company business objectives. In the case of a non-conformity between these values and the business objectives, some changes can be suggested to the system's properties or on the process model (see Figure 5.3.10).

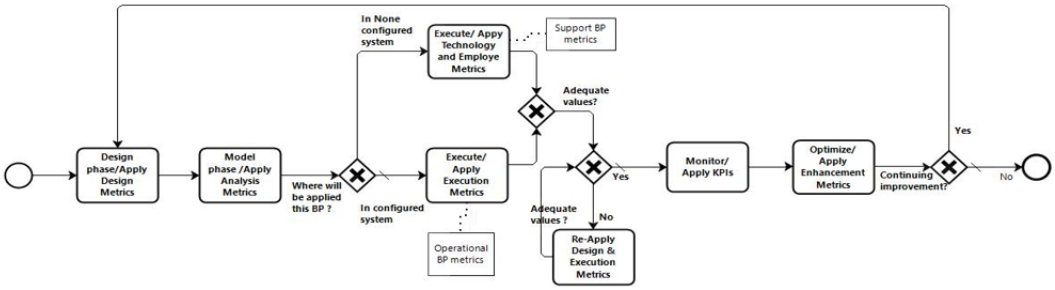


Figure 5.3.10: The execute phase's scenarios

5.3.4 Monitor phase

The process has been running for a while in the execution phase, and all data have been collected, we can periodically monitor the resulted process, in order to obtain a monitored process model: 1. Used and 2. Reused.

Based on [34], we group our KPIs into process efficiency, process effectiveness and process adaptability. We define our KPIs based on the logic combined with: Calculate time, cost, utility of the BP which leads to check if it is flexible relatively to user satisfaction and with suitable quality.

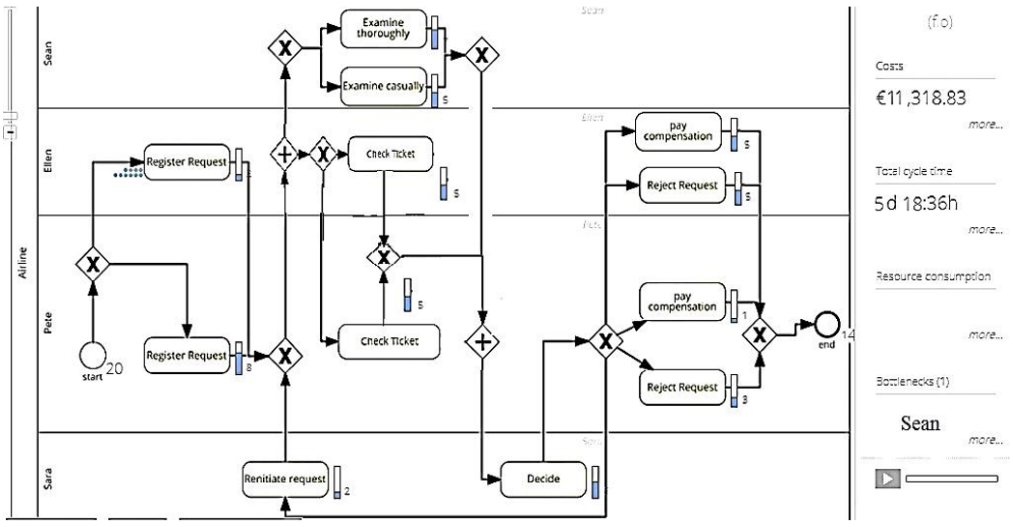


Figure 5.3.11: Monitor the executed process

a- Monitored Process Model: In this step, we monitor the studied business process using the keys performance indicators to obtain real-life results (see Figure 5.3.11). These KPIs also give more information about its Efficiency, Adaptability and Effectiveness. Therefore, BPs must be structurally and semantically corrected, in order to answer to the company objectives. In this sense, we verify if the airline company can answer to strategic and operational questions.
With the same conditions of the two previous phases (Model & execute), we verify the executed process (see Figure 5.3.12) using the Signvaio tool. This is done to measure the resulted process model related values after the aforementioned modifications insert.

Table 5.3: Monitoring Values (Signavio Report)

KPIs		**The monitoring values**	**The company objectives**	**The verification values**
Efficiency	Cost	11318,81€	10318,81€	10318,81€
	Time	5d 18:36h	5d 05:36h	5d 05:36h
	Utility	80%	80%	80%
Adaptability	Flexibility	70%	90%	90%
Effectiveness	Satisfaction	80%	95%	95%
	Quality	95%	95%	95%

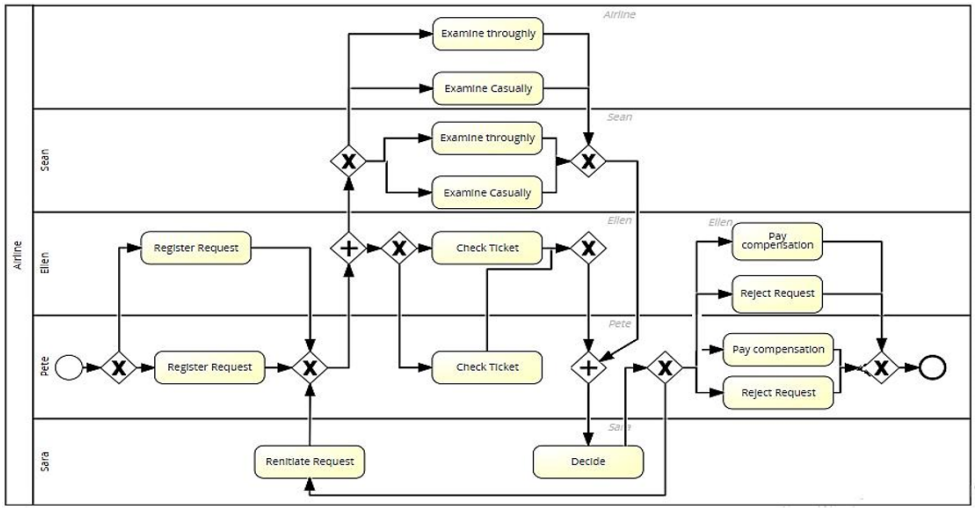


Figure 5.3.12: The monitored process

b- Simulation: In the table 5.3, we illustrate the Signavio report. It shows the monitoring results with the verification values related to the company business objectives. This table is obtained after 5 days of treatment.
We observe that the obtained values are deviated from what the company expectated. These deviations are projected on the Customer satisfaction and the BP adaptability, respectively with the values: 90% and 70%. These values influence the BP time and cost. This implies that some cases of the BP do not respect the deadline date and answer out of time to customers. Indeed, deviations

are referred to delays in the examination activities "Thoroughly & casually", which are attributed previously to the role "Sean".
So, a new resource "Sea" must be nominated as responsible on these activities in order to subdivide the examination activities charge between "Sean" and "Sea".
In order to evaluate and ensure the outcome of the monitor phase (The monitored process (see Figure 5.3.12)), we apply these modifications on the executed process model, and we re-examine monitoring values.
Later, after monitoring the modified process, we obtain verification values. These values adequately answer to the company objectives. This is observed in the table 5.3 by the equality of the verified values (Efficiency, Adaptability and Effectiveness). This will lead to obtain a performant monitored process model (see Figure 5.3.13).

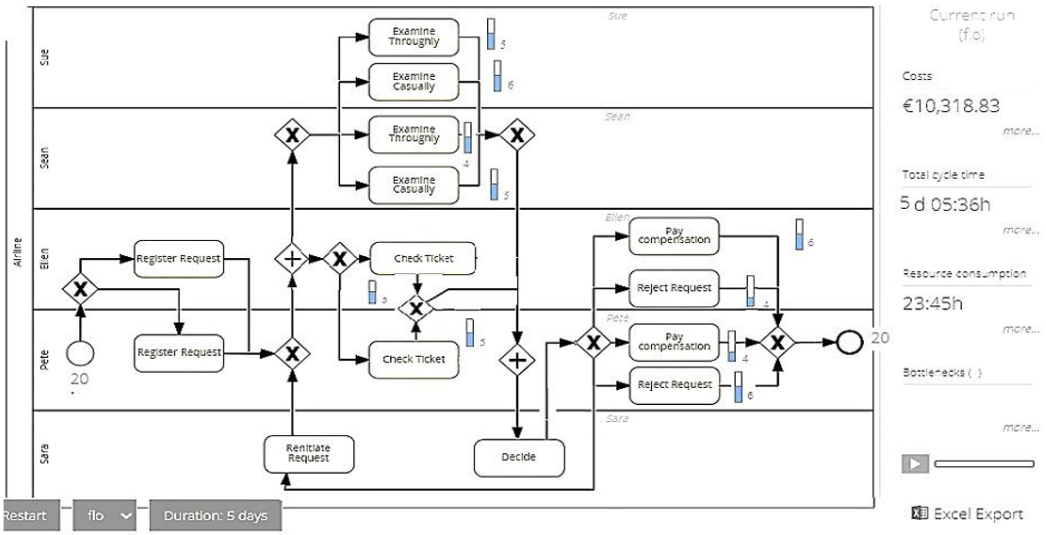


Figure 5.3.13: The verification simulation (monitored process)

c– Synthesis: After applying the KPIs, we detect some bottlenecks (delay, dissatisfaction, etc.). To fix these non-conformities, we suggest two scenarios. We may modify the process flow model (see Figure 5.3.14) or we may proceed directly to the process mining phase to redesign the process flow model and adjust metrics.

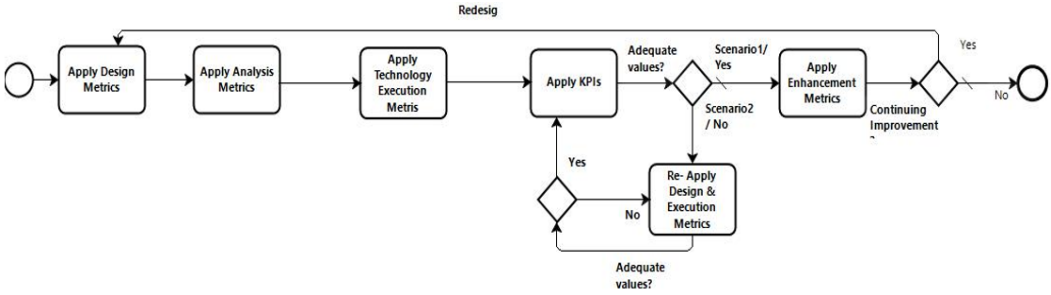


Figure 5.3.14: The monitor phase's process

5.3.5 Optimize phase

After the process has been running for a while in the execute phase, and event logs have been collected. We apply process mining techniques on the collected event logs by measuring: Quality Criteria, Control flow perspective, Organizational perspective, Case perspective and Time perspective. To end up with indications for an optimized process.

	Activity	Costs	Resource	case:concept:name
0	register request	50	Pete	1
1	examine thoroughly	400	Sue	1
2	check ticket	100	Mike	1
3	decide	200	Sara	1
4	reject request	200	Pete	1

Figure 5.3.15: Data source Example

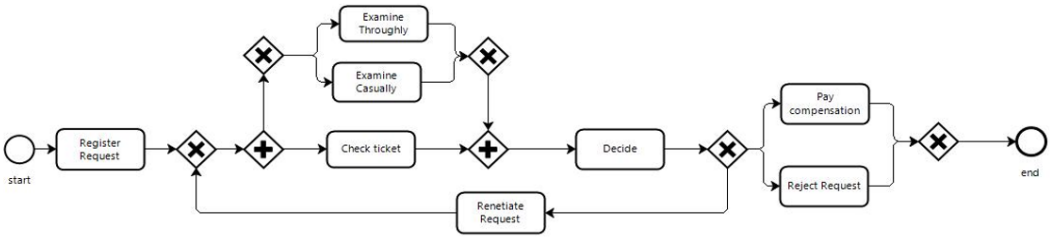


Figure 5.3.16: The Discovered process model

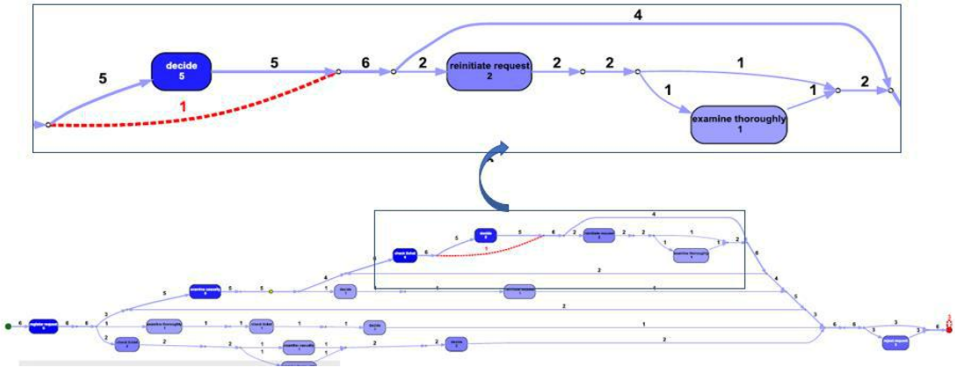


Figure 5.3.17: Analysis phase (conformance checking)

a- Optimized process model: After the process execution and event logs are collected (see Figure 5.3.15). These logs will be used with the process mining techniques. Therefore, to manipulate PM on our case study, we need a PM tool.
According to [40], we decide to use Prom [86] as a framework implementing process mining tools by uploading different plug–ins, that allow to manipulate several mechanisms on the BP. Prom is compatible with several other modelling tools according to the extracted format. In this phase, we

will examine throughout 3 PM types:
The first PM techniques is the discovery process; it takes an event log (see Figure 5.3.15) and uses a discovered algorithm (For instance, Alpha algorithm) to produce the executed process model (see Figure 5.3.16).
The second PM technique is the conformance checking; it uses an existing model to compare it with the extracted event logs. The comparison can deduce bottlenecks (For instance, inductive mining): Event Logs + process model Conformance checking. By applying the conformance checking technique, we observe several deviations. For instance, in our case, we observe that one trace is deviated between the "check ticket" activity and the "reinitiate activity" (see Figure 5.3.17). These two activities respectively attributed to "Ellen", "Pete" and "Sue". We observe also, that Bottlenecks are closely appeared for the activities (examine thoroughly, examine casually, pay compensation and reject request) attributed respectively to roles (we cite: Sue, Sean, Ellen and Pete) [7]. The third PM technique is the enhancement process; aims at optimising and improving existing process models using related information of the event logs (PM perspectives): Event Logs + process model. In order, to obtain an improved process model (see Figure 5.3.18), we combined all PM perspectives (Control-flow, Organizational, Time, Cases). Indeed, we propose the following modifications: - New resource Sea is predicted for the activities (Examine casually, Check ticket, Pay compensation and Reject request), in order to subdivide the charge between Mike [8] and the other accused roles in the conformance step, - Attribute the Reject activity to Pete.

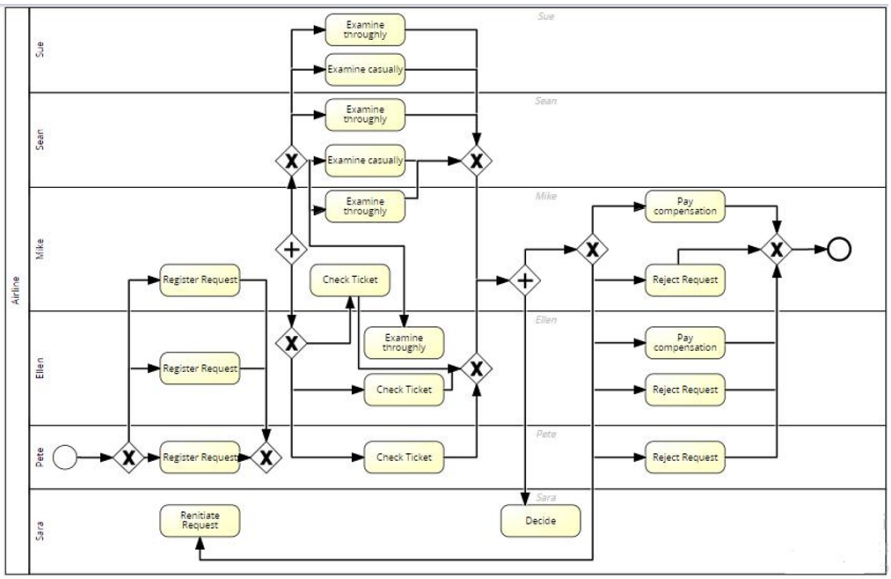


Figure 5.3.18: The Optimized business process

b– Synthesis: The process mining techniques enable organizations to analyse their business processes, diagnose problems and get suggestions for treatment. The optimized process model (in our case to be improved) could be the first step in the BPM life-cycle, as well, the last one. The first

[7] Resources
[8] Resource

step in a continuous improvement cycle for redesigning the BP is the restart of the BPM life-cycle with the proposed improvements indicators. From the long-term standpoint, it is considered as the last step, where the company can achieve it incrementally through the BPM life-cycle phases (see Figure 5.3.18).

5.3.6 Synthesis

Throughout this section, we have applied a set of improvement indicators on a concrete example. This BP is about a job of handling requests for compensations within an airline company. This illustrative example verifies the applicability of our proposed approach. For each phase, we have observed the BP improvement using the defined improvement indicators.

5.4 Summary

This chapter reported our proposed approach that classifies a set of improvement indicators throughout the whole BPM life-cycle. This approach can be applied on structured BPs (see Figure 5.4.1), in order to maintain its simplicity and improve its BP performances.

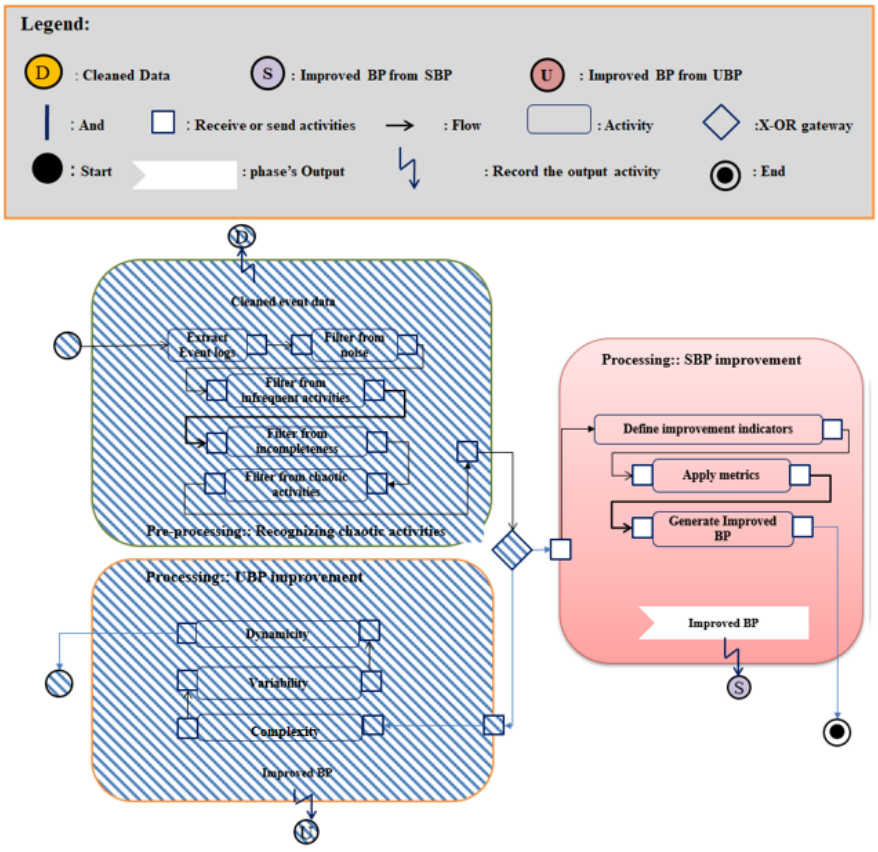


Figure 5.4.1: Processing: SBP improvement

First, we have defined a set of improvement indicators for each BPM life-cycle. Second, we have nominated each BPM life-cycle phase outcome. Third, we have proposed an illustrative example, where these improvement indicators are applied. Last, an improved BP has been obtained.
However, the human intervention during the BP execution may provide more unexpected users' behaviours. In this regard, a BP instruction can be appeared. Therefore, it is important to treat unstructured BP challenges to achieve an improved and performant BP.

Chapter 6

Processing: Unstructured BP Improvement

This chapter is larger than the other chapters because it deals with three issues related to unstructured BPs. Indeed, we propose three approaches to resolve the complexity, the variability and the dynamicity challenges that can be acquired with unstructured BPs. The proposed approaches use process mining techniques. The tackled challenges that are presented in the following order: First, the complexity concerns the possibility of supporting, at runtime, complex BPs by predicting and recommending actions (Section 6.2). Second, the variability consists of managing BPs according to the users' objectives (Section 6.3). Third, the dynamicity introduces the concept of adaptability during the BP execution (Section 6.4). Last, Section 6.5 concludes the chapter and gives summary of the contributions.

6.1 Background

Unstructured BPs (UBPs) are difficult to be analysed and hard to be understood. Especially, during the application of operational support systems for detecting violations, predicting events, and recommending actions. Therefore, it is mandatory to provide an operational support approach for UBPs in order to treat the complexity problematic. Moreover, the difficulty is seen in variants sub-processes are emerged and recorded in the information system. In this sense, we start from the Information Retrieval (IR) system definition. IR is a set of resources and tools that allows users to search for information in a given domain. This system permits users to perform their research according to their objectives in diverse ways producing different behaviours. Even users with the same objective may follow different paths and stand different sub-processes, which are introduced as self-defined BPs (the case of digital libraries, e-commerce websites, cyber-physical systems, Electronic administration services, etc) that vary in terms of structure, objective, and result. This puts forward the difficulty of obtaining and studying user's behaviours.
Beyond, Ad-hoc processes are characterized by a non-defined workflows design. Indeed, the control-flow between activities cannot be modelled in advance but can simply occur during runtime. Thus, users must be able to decide what to do and when. They must also be able to assign

work, as sub-process, to other people and create interactions among various users. This puts forward the difficulty of treating dynamically Ad-hoc processes.
Therefore, it is required to take into consideration situations where the process flow depends on real-time variables. A special interest is given to the use of process mining techniques to deal with the variability and the dynamicity challenges. The main result must be a set of recommendations to end users.
To conclude, the following sections will present three process mining approaches to: 1- Support UBP at runtime in the complexity context, 2- Manage UBP variability in a self-defined BP context, and 3- Provide dynamic manner to treat Ad-hoc processes.

6.2 Complexity

The complex structure of unstructured BPs emerges the difficulty of predicting actions during the BP execution. This requires the use of other process mining techniques. In this sense, we cite the recent refined process mining framework. This framework extends process mining types into three categories with ten activities, which are: Navigation (Discovery, Enhance and Diagnosis), Auditing (Detect, Check, compare and Promote) and Cartography (Explore, Predict and Recommend). These activities link current and historic data to the de jure model (a normative model that specifies how things should be done or handled) and the de facto model (a control-flow model that represents the order in which process model activities must be executed). However, the most challenging task is how current situations benefit from historic data.
To this purpose, operational support systems have been defined, to learn from existing structured models, normative ones, historic and current or running data. Thus, the use of the Detect, the Predict and the Recommend activities is mandatory. Also, the appearance of the predictive (aims at predicting an outcome that can influence next events) and the recommender models (aims at defining the "preference" that can be attributed to an activity or a resource) is crucial. These two models are considered as inference ones. Here, operational support approaches perform well with SBP, while they still a challenging task for UBP. In this respect, the still encountered issues related to the UBP operational support application.
Buijis [59] presents briefly how the process mining activities re-organization can provide an operational support for UBP, based on the structured BP version, i.e., reducing the UBP complexity. This operation necessitates the intervention of other process mining activities as Diagnosis, Check, Promote, etc. Hence, the order of process mining activities stills a questionable task. To this end, our approach objective is to establish an operational support approach that deals with UBP, i.e., detects violations, predicts events and recommends actions for unstructured BPs at runtime. So, we suggest combining, in a specific order, the ten activities of the refined process mining framework.
Tackling the analysis of UBP through the orchestration of existing process mining activities, by necessity techniques, brings new knowledge to the research field in terms of producing a complete approach that can treat the complexity problematic. Sub-section 6.2.1 details our UBP operational support approach phases. In sub-section 6.2.2, a concrete example is depicted to simulate our approach. synthesis is presented in section 6.2.3

In this sense, we detail our UBP operational support approach in terms of work entities and architecture. In this regard, we describe the process mining activities order and we present their workflow through the enterprise work entities.

6.2.1 Overview of Our Operational Support Approach

The complex structure of unstructured BPs requires the use of other process mining techniques. Therefore, we will re-use the refined process mining framework activities [20]. The novelty, here, consists of re–ordering activities, to present a complete scenario of the existing techniques orchestration. In this respect, we arrange the process mining activities according to the following logic (see Figure 6.2.1):

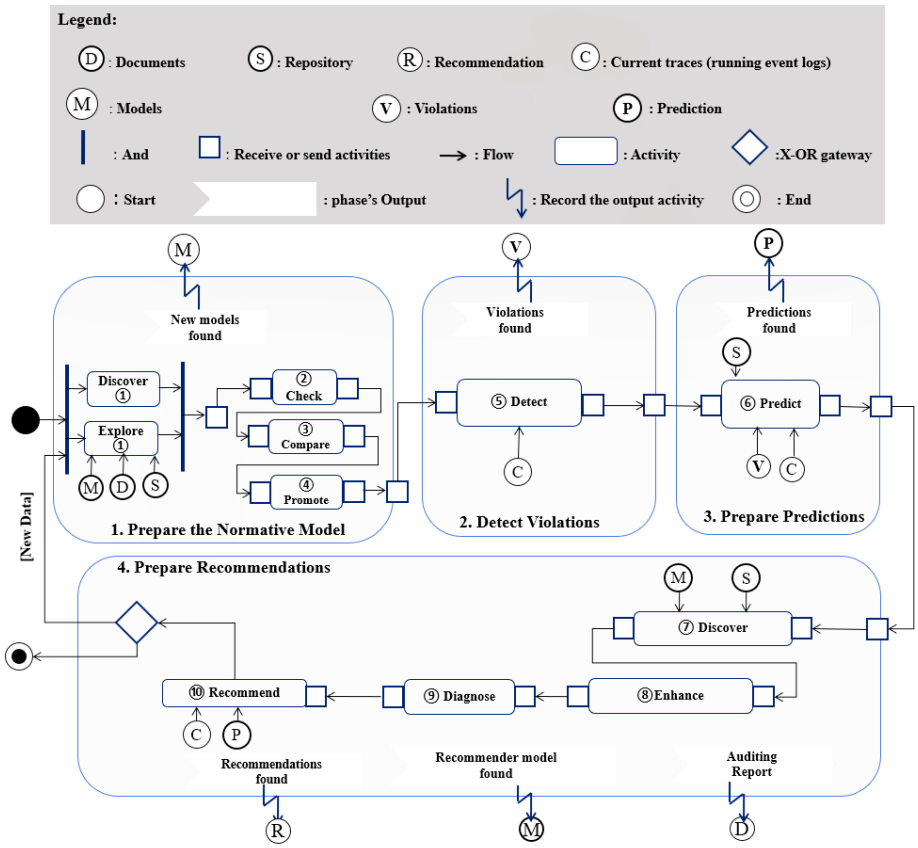


Figure 6.2.1: An overview of our operational support approach

1. To achieve the UBP operational support objective, we require the existence of a SBP, an Initial Normative Model (INM) and a refined or final normative model.

2. To obtain the SBP of an UBP (de facto model), we apply the simplification algorithm combined with the [133] structuring techniques. Here, we select heuristics miner algorithm for the Discover ① activity. In parallel, to define the initial normative model, we Explore ① documentations and recorded models.

3. To obtain the refined normative model (Nominated as the de jure), it is required to use an audit approach by proceeding through these activities (Check ②, Compare ③ and Promote ④). At this stage, we use the SBP and the INM.

4. After obtaining the refined normative model and the SBP, we can detect violations (Detect ⑤). This allows predicting events (Predict⑥) using the predictive model. We can also Recommend ⑩ actions relatively to the obtained diagnosis information (discover ⑦, Enhance ⑧ and Diagnose ⑨) and the recommender model.

In our approach, we denote de facto model as an UBP before applying structuring techniques and as a SBP after the structuring step. As well as we denote a refined normative model as a de jure model or a final model. In this regard, we can present our approach into the following four phases (see Figure 6.2.2): We start by the phase of preparing the normative model using five activities: Discover, Explore, Check, Compare and Promote. This phase's output is a refined normative model. Then, we proceed to the second phase that aims at detecting violations using the Detect activity. According to the revealed violation and historic data, we can predict events in the third phase. Last, we use diagnosis information, to generate a recommender model and acquiring suitable recommendation using the Discover, the Enhance, the Diagnosis and the Recommend activities.
In the company environment, we determine different enterprise work entities (part a in figure 6.2.2); each one releases a specific business. In the literature [134, 135], we distinguish:

1. The operational support system: allows supporting processes at runtime, in terms of detecting violations, predicting events and recommending actions by using normative or De jure, predictive and recommender models.

2. The management system: allows defining normative models and business rules.

3. The information system: allows executing BP, recording events and communicating messages between users and other entities.

Additionally, our approach defines two new entities, which are the DECoD and the construction systems:

1. The DECoD (Discover, Enhancement, Conformance, and Diagnosis) system: allows generating diagnosis information that are required in the definition of predictions and recommendations.

2. The construction system: allows producing predictive and recommender models from historic event logs. These historic events are considered as a set of training data that aims at enriching predictive models.

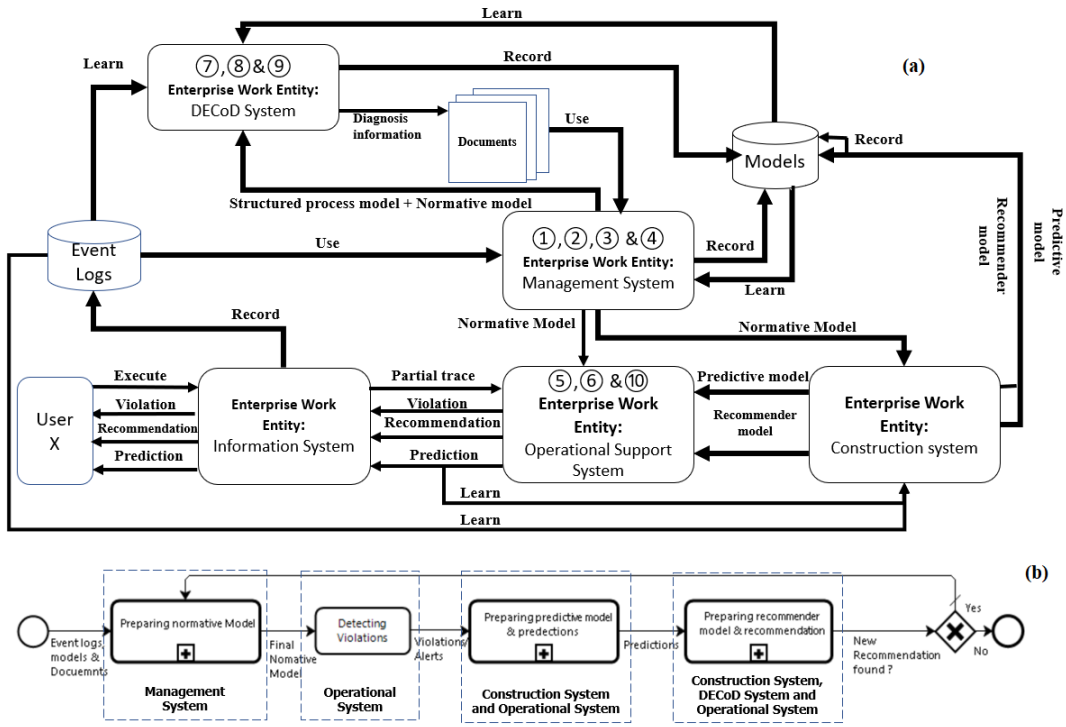


Figure 6.2.2: Our UBP approach architecture including work entities

To do so, the flow passes between and throughout these five entities performs as follows (part b in figure 6.2.2): At the beginning, we prepare a refined normative model (Promoted version of de jure model). Then, we use it to detect current traces violations, i.e., processes that deviate from business rules. Next, we prepare a predictive model and by necessity we predict events according to detected violations, historic traces and current events. Consequently, we can construct the recommender model and recommendations. At the end, we record this latter and its corresponding violation alert. This operation is often attributed to the information system, i.e., it records predictions, models and auditing reports. This later broadcast informative messages.

According to different phases' outputs, we propose to match each phase to a specific work entity (Approach phase → Enterprise work entity), which are: Preparing normative model → Management System; Detecting violations → Operational support system; Preparing the predictive model and predictions (sub–subsection → Construction system and Operational support system; Preparing the recommender model and recommendations → Construction system, DECoD system and the Operational support system.

In this sense, table 6.1 illustrates inputs and outputs data of each work entity. For instance, the management work entity uses documents, logs and models for producing a refined normative model.

In this order, we will explain each phase, the corresponding enterprise work entities and the process mining activities.

Table 6.1: Enterprise work entities inputs and outputs

Work Entities	**Management System**	**Operational Support system**	**DeCOD system**	**Construction system**	**Information system**
Input	Event logs, models, documents	Historic and current traces, Predictive and recommender models	Documents, event logs and models	Historic event logs	User execution, event logs
Output	Refined or Final normative model	Violations, predictions, recommendations	Diagnosis information	Predictive model, Recommender model	Informative message, Alert

6.2.2 Illustrative case study

In this section, we present an illustrative example for simulating our UBP operational support approach. The example is about the road [1] traffic fine management process. The control-flow process model consists of 11 activities and 561470 events grouped into 150370 cases. By definition, this process model is considered as an UBP (see Figure 6.2.3).

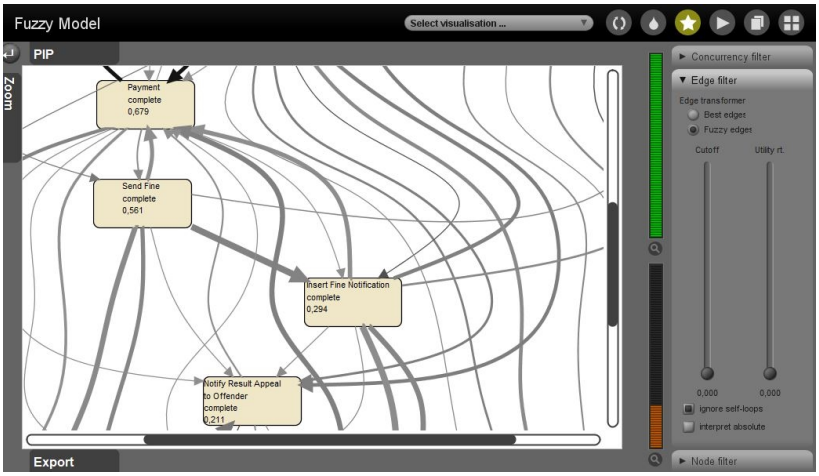


Figure 6.2.3: An unstructured process: Excerpt showing 20% of road traffic fine management process

a- Preparing the normative model step:

Throughout this section, we apply the four phases of our UBP operational support approach (Cf. Figure 6.2.2), in order to treat a real–life UBP example. To do so, we demonstrate how workflows are passed between and through our UBP operational support approach work entities, respectively to the following order: preparing normative model in the management entity, detecting violations

[1] https://data.4tu.nl/repository/uuid:270fd440-1057-4fb9-89a9-b699b47990f5

in the operational support entity, predicting and recommending events at runtime into construction and operational support entities.
This sub-section applies the first phase of our UBP operational support approach. It aims at preparing the refined normative model. This model allows detecting deviations according to specifics business boundaries. It also nominated as the final normative model. For this purpose, it is important to use simplification algorithms, structuring, checking, comparing and promoting techniques.

1- Simplifying and structuring

Our process, with excerpt illustrated in figure 6.2.3, starts with a fine being created (i.e., Create Fine). After, a fine has been created, it is sent to the offender's place of residence (i.e., Send Fine). When the offender receives the fine, the date of reception of such notification is also registered (i.e., Insert Fine Notification). After this action, the fine should be paid within 60 days (i.e., Payment). However, in case the fine was physically handed over to the offender, e.g., by means of a parking ticket, the offender is able to immediately pay the fine. In such case, the fine will not be sent and there will be no registration of the notification. This exception, i.e., direct payment after ticket creation saves the offender administration costs. In total, the offender has 60 days to either pay the fine or appeal against it. After this period, a penalty is added to the fine amount (i.e., Add Penalty).
After obtaining the SBP, we look for its related refined normative model. This helps in analysing the SBP as an auditing action. In this regard, we promote the INM with the SBP to obtain the final normative. All in all, we propose a framework for structuring the UBP and refining the normative model (see Figure 6.2.4).

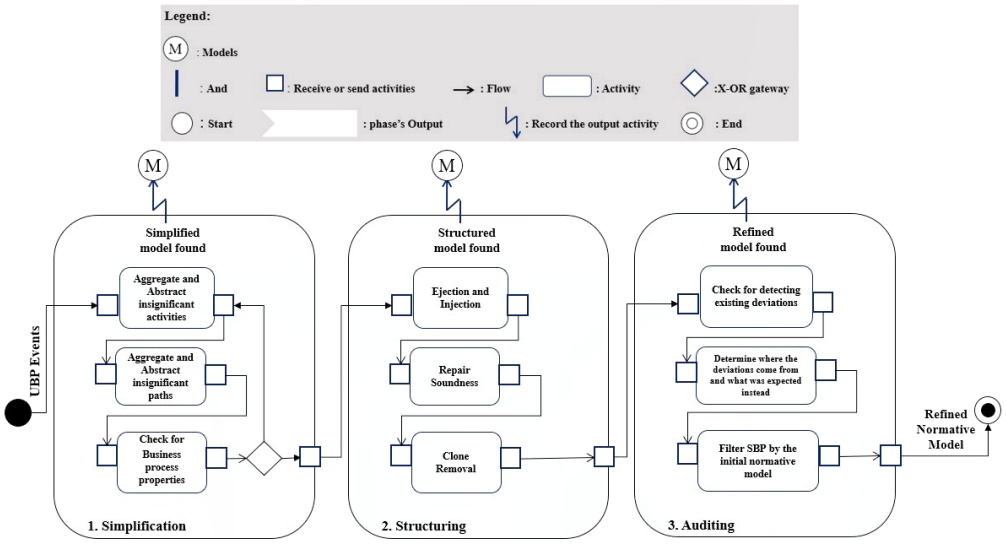


Figure 6.2.4: Framework for producing structured and refined normative models

Our framework starts by the extraction of UBP event logs and their simplification, using heuristics mining technique. This algorithm abstracts a process model based on the frequency activity parameter, which can extract main behaviours of an unstructured process (here, it produces a de

facto model). The current step is repeated until obtaining a concrete simplified BP with fitness value at least equal to 80% (i.e., at least 80% of behaviours should match between model and the event log).

After obtaining the simplified process model, we apply the structuring techniques for generating a SBP (latest or improved version of de facto model). Then, we proceed to the audit phase, which includes the Check, the Compare, and the Promote activities. Here, we obtain a refined normative model, which aims at filtering the SBP with an INM (INM is often nominated as de jure model), in order to extract the control–flow related features. These features serve mainly to identify frequent execution paths with an abstracted representation (see Figure 6.2.4).

In this example, we consider works treating the process at hand, as data sources of the INM. For instance, we use these two works [136, 137] for obtaining the control–flow INM.

In our example, we apply the simplification phase using the heuristics miner algorithm (see Figure 6.2.5). This aims at discovering the de facto process model that can be converted to other modelling languages like: BPMN, in order to use structuring techniques (gateways structuring, soundness repair, clone removal) as the second phase of our framework.

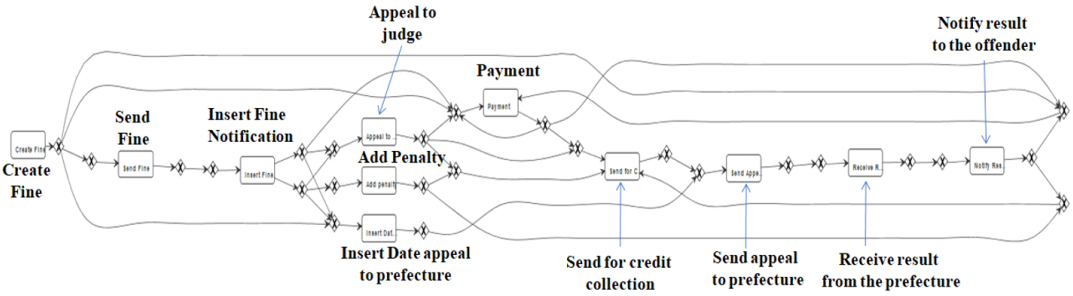


Figure 6.2.5: Simplified process model discovered using heuristics miner algorithm

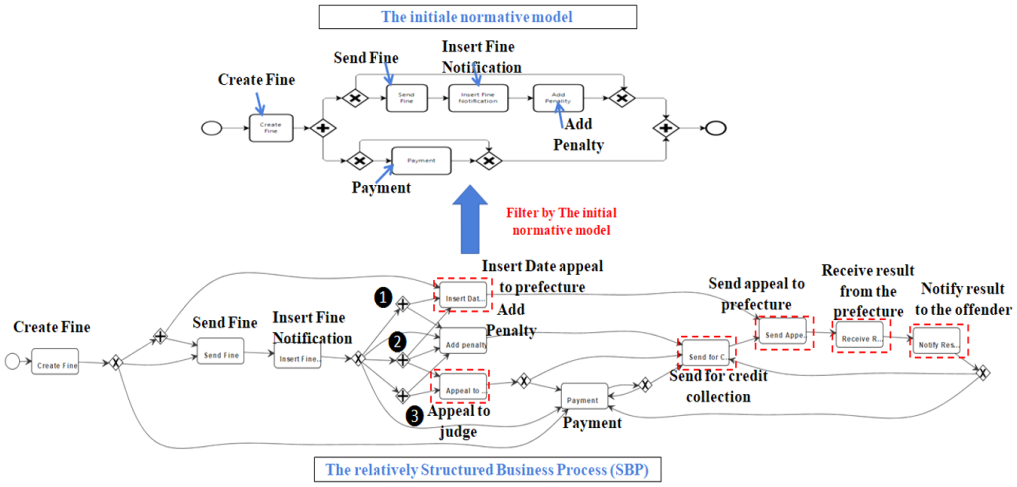


Figure 6.2.6: Filtering the SBP by the initial normative model (Promote)

In figure 6.2.6, we observe that the converted model does not respect the [133] structuring techniques. For instance, XOR gateways are unstructured. Therefore, we must use structuring techniques.
The structuring techniques application results the process model illustrated in figure 6.2.7. This model is relatively structured; we can use it to reach specifically the next step of our simplification framework (see Figure 6.2.4), i.e., the auditing step and generally next phases of our UBP operational support approach (Cf. Figure 6.2.1), i.e., detect violations.
For further process model structuration, we suggest the adjustment of the three parts (❶,❷ and ❸) of the figure 6.2.7.

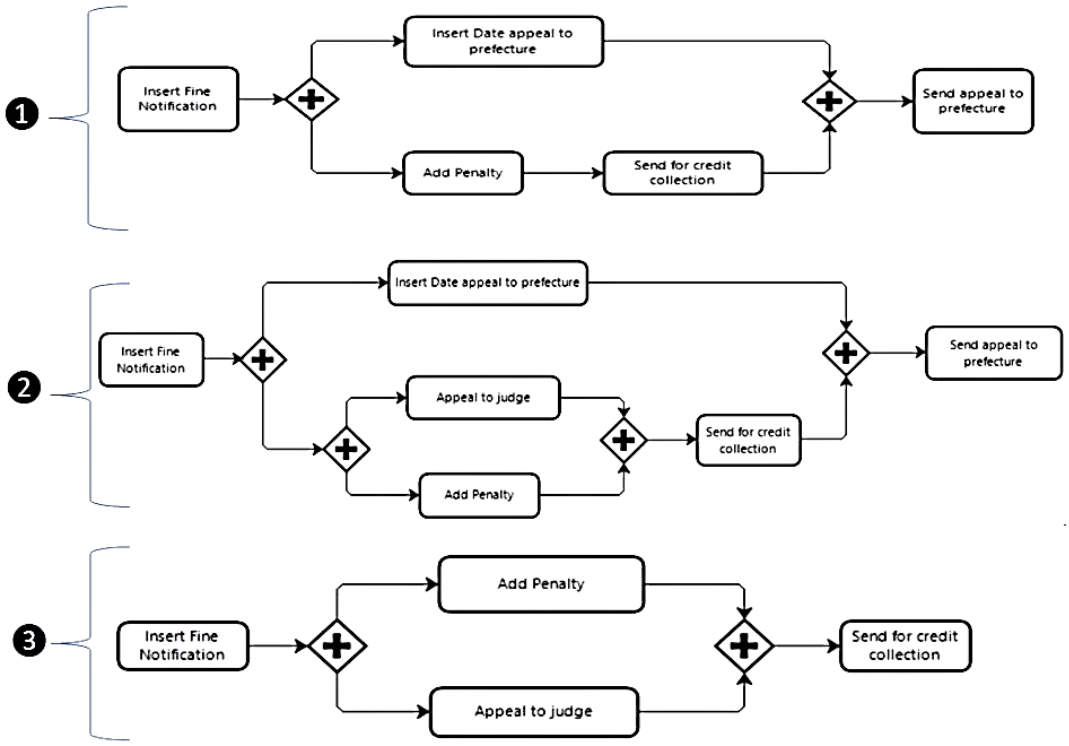


Figure 6.2.7: Re–structuring the de facto process model parts ❶,❷ and ❸

2- Auditing

We start this phase by checking where and why the INM deviates from historic events. Then, we compare the INM to the de facto model (SBP), in order to analyse these deviations. Consequently, we observe the appearance of new activities. This is representing deviations causes between the INM and de facto model (see Figure 6.2.6): 1-Insert date appeal to prefecture, 2-Send for credit, 3-Send appeal to prefecture, 4-Receive result from prefecture, 5-Notify result to the offender and 6-Appeal to judge. Last, the INM will be promoted based on the SBP model, in order to obtain the refined normative model.
The frequency matrix (shows percentage of the direct flows between activities) is absolutely integrated into the heuristics miner algorithm application. Moreover, deviations emerged between the INM and the SBP are detected with the comparing and the checking activities. Therefore, we

proceed directly to the refine action by filtering the SBP with the INM into the Promote activity. This is producing a new frequency matrix that can generate a refined normative model (see Figure 6.2.8). In this sense, we use two values (1 and 0) for defining relations between the finding activities (solid or not exist). The activities illustrated in table 6.2 are resulted after applying the filtering step (Promote according to observed deviations), i.e., the entire log activity named CF[2] has a solid relation with the SF[3] activity. For instance, CF as x is directly followed by SF as y (x>y).

Table 6.2: frequency matrix for applying the Promote activity (x followed by y)

x>y	CF	SF	I	A	P	SP	ID	RP	NO	AJ	SC
CF	0	1	0	0	1	0	0	0	0	0	0
SF	0	0	1	0	0	0	0	0	0	0	0
I	0	0	0	1	1	1	0	0	0	0	0
A	0	0	0	0	1	0	0	0	0	0	1
P	0	0	0	0	0	0	0	0	0	0	0
SP	0	0	0	0	0	0	1	0	0	0	0
ID	0	0	0	0	0	0	0	1	0	0	0
RP	0	0	0	0	0	0	0	0	1	0	0
NO	0	0	0	0	0	0	0	0	0	1	0
AJ	0	0	0	0	1	0	0	0	0	0	1
SC	0	0	0	0	0	0	0	0	0	0	0

According to table 6.2, new activities are observed and must be handled by the refined normative model (see Figure 6.2.8), which are: CF (Create Fine), SF (Send Fine), I (Insert Fine Notification), A (Add penalty), P (Payment), SP (Send appeal to Prefecture), ID (Insert Date appeal to prefecture), RP (Receive result from the Prefecture), NO (Notify result to the Offender), AJ (Appeal to Judge), SC (Send for credit Collection). Therefore, the new process is described as follows: If the offender appealed against the fine within 60 days, the appeal is sent to the corresponding prefecture (Send Appeal to Prefecture), which is registered when it is received (Insert Date Appeal to Prefecture). The results of the appeal are sent back to the municipality (Receive Result Appeal from Prefecture) and they are notified to the offender (Notify Result Appeal to Offender), which can appeal against the result (Appeal to Judge). If the offender does not pay (possibly after a denied appeal), the fine is sent for credit collection (Send for Credit Collection). The refined normative model is represented within Petri Net notation. This notation is suitable to next verifications. The model, illustrated in figure 6.2.8, does not capture any form of parallelism. It is merely a sequence of activities combined with possible choices.

To conclude, the input data (event log) was organized to promote the discovery task by clustering the log into deviated and none-deviated activities, to obtain different sub-logs. These sub-logs are used to carry out the activities presented in the workflow.

[2]Create Fine

[3]Send Fine

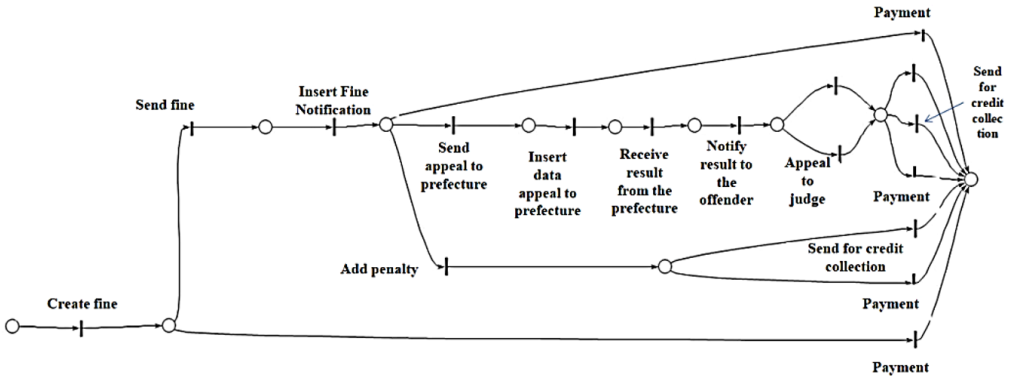


Figure 6.2.8: The refined normative model with Petri Net notation (Excerpt)

Therefore, the efficiency of the transformation, UBP to SBP, is demonstrated in the simplicity and the structuring of the resulted road traffic process. As appeared in figure 6.2.3), the control-flow was with a complex structure. After the transformation, it appears in a readable form with reduced number of activities and paths (see Figure 6.2.8).

b- Detect violations:

In this step, we apply the second phase of our UBP operational support approach. We try to detect violations at runtime. Indeed, we replay the following trace [4] (see Table 6.3) on the refined normative model:
$<CF_{start(10)}, CF_{complete(20)}, SF_{start(25)}, SF_{complete(30)}, I_{start(31)}, \mathbf{A}_{start(32)}>$.

Table 6.3: Historic traces

CaseID	Trace
1	$CF_{start(12)}$, $CF_{complete(19)}$, $SF_{start(25)}$, $SF_{complete(35)}$, $I_{start(36)}$, $I_{complete(40)}$, $A_{start(45)}$,$A_{complete(50)}$, $P_{start(61)}$, $P_{complete(70)}$
2	$CF_{start(25)}$, $CF_{complete(30)}$, $SF_{start(32)}$, $SF_{complete(26)}$, $I_{start(27)}$, $I_{complete(33)}$, $A_{start(35)}$,$A_{complete(40)}$, $P_{start(41)}$, $P_{complete(46)}$
3	$CF_{start(17)}$, $CF_{complete(23)}$, $SF_{start(28)}$, $SF_{complete(32)}$, $I_{start(33)}$, $I_{complete(38)}$, $A_{start(50)}$,$A_{complete(59)}$, $P_{start(70)}$, $P_{complete(75)}$
...	...

At time 10, after executing the first event $CF_{start(10)}$ no deviation is found because trace the $CF_{start(10)}$ can be replayed without missing tokens. The next two events can also be replayed, i.e., $CF10_{start}$, $CF20_{complete}$. $SF25_{start}$ is a possible firing sequence of the workflow net, in which each activity is refined into a start and complete transition. The next event, i.e., $A_{start(32)}$ is not possible in this state. Hence, an alert is generated at time 28. The alert signals that activity A was

[4] CF (Create Fine), SF (Send Fine), I (Insert Fine Notification), A (Add penalty), P (Payment), SP (Send appeal to Prefecture), ID (Insert Date appeal to prefecture), RP (Receive result from the Prefecture), NO (Notify result to the Offender), AJ (Appeal to Judge), SC (Send for credit Collection)

started without being enabled. Therefore, a violation, in the activity Add Penalty, is detected because it is started while the insert fine notification is not released yet: $< CF_{start(10)}, CF_{complete(20)}, SF_{start(25)}, SF_{complete(30)}, I_{start(31)}, \mathbf{A}_{start(32)} >$.
Therefore, it is important to predict when the insert fine notification activity should be achieved to start the A activity. This puts forward the need of a predictive model.

c- Preparing the predictive model and predict events:

In this sub-section, we apply the third phase of our UBP operational support approach (see Figure 6.2.2). We aim to prepare the predictive model and predictions. In this context, the predictive model learns from historic events, in order to collect possible predictions. To do so, we use an example of historic traces, mentioned in table 6.3, for constructing the predictive model. In this sense, it is possible to make predictions about future events, e.g., the remaining flow time and the probability of success. This is done by combining information about running cases with discovered models as SBP or hand-made as de jure model. For example, we use the following parameters, to predict the completion time of the insert fine notification activity: the remaining flow time of the targeted case, the sojourn time (Time spent in a state before passing to the next) of each case and the elapsed time at a specific state.
To this end, figure 6.2.9 demonstrates the replaying operation of three cases. In this figure, the initial state [start] has no annotations since no events have occurred when visiting this state. The final state [P4] has no sojourn time because there is no next event, when visiting this state.
Let us consider the first case $< CF_{start(12)}, CF_{complete(19)}, SF_{start(25)}, SFcomplete(26), I_{start(27)}, I_{complete(33)}, A_{start(35)}, A_{complete(40)}, P_{start(41)}, P_{complete(46)} >$. This case started at time 12 and ended at time 46. Hence, its flow time was 34 time units. States visited by this case are annotated with a tag (t, e, r and s) where t is the time of the state is visited, e is the elapsed time since the start when visiting the state, r is the remaining flow time, and s is the sojourn time. State [CF] is tagged with the annotation (t = 12, e = 0, r = 34, s = 7) because this state was visited by the case directly after the first event $_{CFStart12}$ occurred. t = 12 because event CF12 start occurred at time 12. e = 12-12=0 because no time elapsed after executing just one event. r=46-12=34 is the remaining time until the end of the case after CF was started at time 12. s = 19 - 12 = 7 because the next event occurred 7 time units later. State [p1, CF] is tagged with annotation (t = 19, e = 7, r = 27, s = 6) because CF completed at time t = 19. e = 19 - 12 = 7 because CF completed 7 time units after the case started. r= 46 - 19 = 27 because the case ended at time 46. s = 25 - 19 = 6 because the next event occurred 6 time units later.
At this stage, we use this method [43], for predicting the remaining time of the ambiguous activity (insert fine notification activity). Due to the availability of data (road traffic fine event), tool (Prom plug-ins) and data context [3]. These two methods are considered as completed approaches that every researcher can rely on, to start developing new predictive techniques (related to the public administration domain) and to test impacts on other predictive business monitoring dimensions. For instance, we can demonstrate how the predicted remaining time can influence the next activity in terms of performance (Add penalty) and how this can lead us to select suitable resource to the insert fine notification.
After replaying historic events, we can now predict future ones. In our example, we conclude the

following method for calculating the remaining flow time, the elapsed time and the sojourn time. The average remaining flow time is calculated by applying $\frac{R}{C} = \frac{\sum_{i=1}^{3} r_i}{\sum_{i=1}^{3} c_i}$, where R is the sum of all remaining time in the concerned state (in our case at the state I start(27), we aim to calculate the remaining time:R= $\sum_{i=1}^{3} r_i$) and C is the number of cases replayed (In our example, we have three cases $\sum_{i=1}^{3} c_i$). The average sojourn time is obtained by applying ($\frac{S}{C} = \frac{\sum_{i=1}^{3} s_i}{\sum_{i=1}^{3} c_i}$) where S is the value of sojourn time of each case at a specific state: S=$\sum_{i} s_i$) and the average elapsed time ($\frac{E}{C} = \frac{\sum_{i=1}^{3} e_i}{\sum_{i=1}^{3} c_i}$) where E is the value of elapsed time of each case at a specific state: E=$\sum_{i=1}^{3} e_i$).
Therefore, predictions are:
$< C_{Fstart(10)}, CF_{complete(20)}, SF_{start(25)}, SF_{complete(30)}, I_{start(31)}, \mathbf{I}_{complete(36)}, \mathbf{A}_{start(42)} >$. $I_{complete(36)}$, denotes the time required, to achieve the insert fine notification activity. $\mathbf{A}_{start(42)}$ denotes the time to start the add penalty activity. Last, we record all predictions.

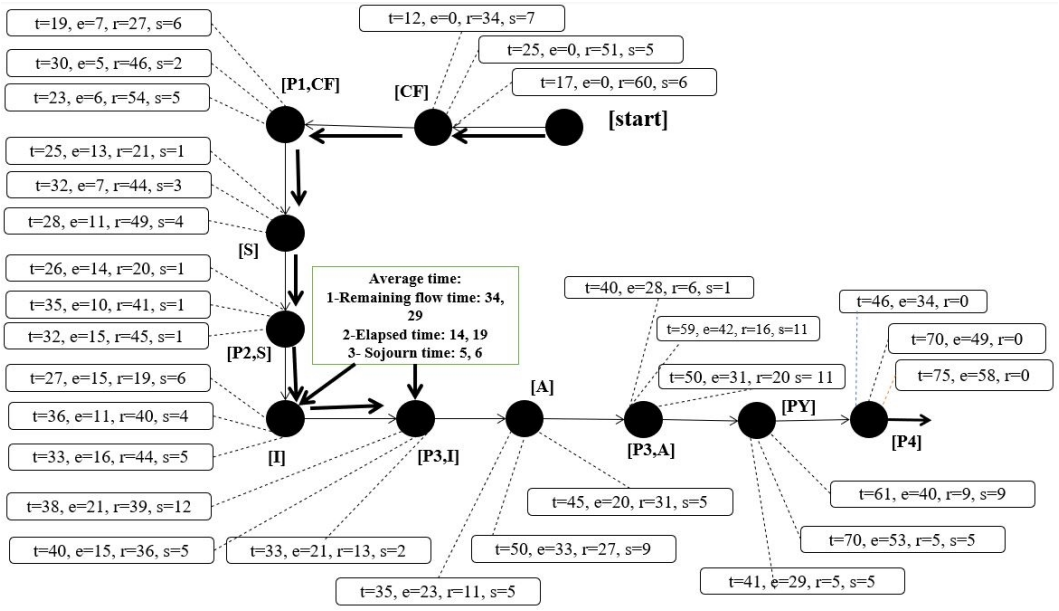


Figure 6.2.9: Statistics while replaying the historic cases (t: the time the state is visited, e: the elapsed time from the start when visiting the state, r: the remaining flow time, s: the sojourn time)

In our example, we look for a sample application to prove our concept. Therefore, we use the simplification of an UBP to a SBP that makes the resolution time predictor seems too simple, instead of its complexity in real-life events. In this sense, we focus on three specific cases, to avoid application errors, while focusing on multiple dimensions can provide complex applications.
d- Preparing the recommendation model and recommend actions:

In this sub-section, we apply the fourth phase of our UBP operational support approach (see Figure 6.2.2). We aim to prepare the recommender model and recommendations. In this order, we use predictions report, to discover the latest version of the de facto model. Then, we treat its diagnosis information. This information aims at enhancing the de facto model and obtaining the recommender model, through the Discover, the Enhance and the Diagnosis activities.

1- Diagnosis information:

Analysing the de facto model (SBP) can give general insights, of the whole process model performance, and specific ones of each activity's performance. In our example, we choose to diagnose information from the social network investigation. Appropriately, we present the road traffic fine management discovered process model according to the social network resources (see Figure 6.2.10). Theoretically, each activity is attached to a specific cluster (resources executing the same activity). At this stage, we focus on the insert fine activity. We observe that cluster C resources are handling this activity.

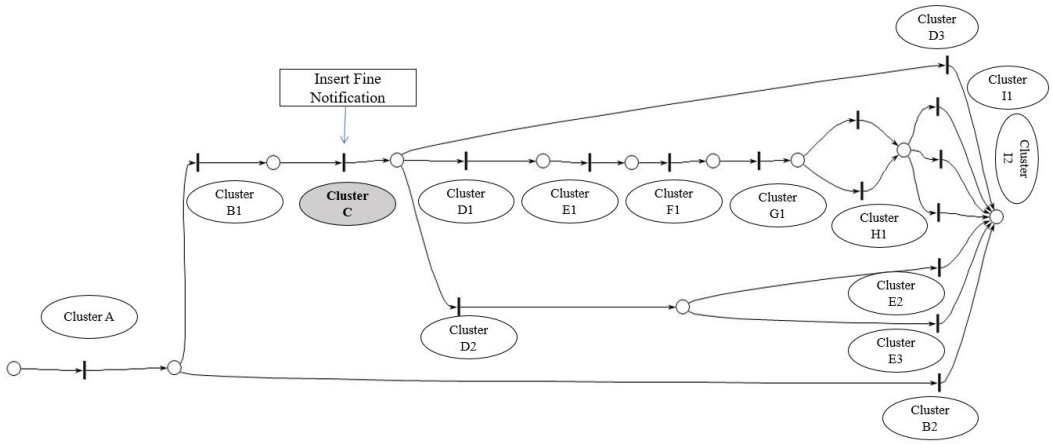


Figure 6.2.10: The de facto model with clusters using the social network plug-in

Based on the discovered social network information, we proceed to the Enhance activity application. In this sense, process model improvement can be performed from different process mining perspectives [1]. Among these perspectives, we find those dealing with control-flow perspective, organizational perspective, data-flow perspective, etc. To do so, the de facto model uses historic data, i.e., recorded predictions, for potential improvements. This is done by applying the conformance checking technique. Thus, if we observe that the de facto model deviates from the de jure model, we apply the Enhance activity.

Table 6.4: Business rules

Case_ID	Activity	Timestamp of the de facto model (minutes)	Timestamp of the de Jure model Rules (minutes)
Trace1	Insert Fine notification	06:00	**05:00**
Trace2	Insert Fine notification	07:12	**05:00**
Trace3	Insert Fine notification	**05:00**	**05:00**
Trace4	Insert Fine notification	**05:00**	**05:00**
Trace5	Insert Fine notification	05:35	**05:00**

According to table 6.4, we focus on the time perspective. In this sense, we observe delays in the insert fine notification activity. This activity must be achieved in 5 minutes, but it is attained between 5 and 7 minutes. The boundary rule has been not respected. Based on the selected BP predictive method, cluster C resources are responsible for these emerged delays. At this stage, we can enhance the de facto model according to detected deviations.
For advanced models' investigation, we apply the Diagnosis activity on the de facto model and the predicting events. In our case example, the de facto model will be the same in terms of control-flow representation, because modifications are influencing only the timestamp proprieties of the Insert fine Notification activity and the add penalty activity.
In reality, a de facto model differs from a historic one because of major modifications that aims at improving this latter to a normative model. This progressive transformation complies with a set of predefined rules. Indeed, the de facto model, under ideal conditions, can be similar to the normative model.

2- Prepare the recommender model:

After obtaining the diagnosis information, we can prepare the recommender model. Here, our objective is to find out the suitable resource to execute the insert fine notification activity, respectively to the predicted time (sojourn time: 5 units and completion time at 36). To do so, the existence of a recommender model is required. This model defines possible actions from which one decision can be chosen. This decision is learned from historic data (events represent predictions) and diagnosis information. In the present example, the recommender model should illustrate appropriate resources to the insert fine notification activity and sojourn time corresponding to each resource (see Table 6.5). In this regard, we analyse clusters resources and their corresponding activities. Mainly, we focus on the cluster C. This cluster is responsible for the execution of the insert fine notification activity (see Figure 6.2.10). It contains 8 resources. We mentioned, in table 6.5, the recorded sojourn time of each resource in order to decide which one is the most responsive.
In reality, a resource cluster may execute attached activities to other clusters. For instance, cluster B resources can execute, in a specific time and in a specific situation, the payment activity which is attached to cluster E3 resources. In this context, more parameters can be used to obtain suitable and refined cluster classification.

3- Recommend:

To provide a recommendation, we replay current partial traces on the recommender model. This allows defining possible actions from which we choose one. In our illustrative example, we recommend R35 as a suitable resource to execute the insert fine notification activity. This decision is based on calculating the activity executing certainty, taking into consideration the average sojourn time (TimeR35= 5+6+4, R35=5 units). Thus, the resource R35 executes the insert fine notification activity in an optimal execution time. For instance, we can minimize the business process remaining time by attributing the insert fine notification activity to the resource R35. In this regard, we observe the relationship between predicting events and recommending actions. To do so, recommendations answer to predictions, i.e., we predict that the insert fine notification must be executed in 5 units and by necessity we recommend a resource that will adequately execute this activity in the predicted time boundaries (see Table 6.5).
To conclude, this example demonstrated the impact of numerical prediction on categorical prediction. In this sense, we used numerical prediction to recommend the suitable resource for executing an activity [138].
Besides, we urge using more applicative examples. Likewise, more experiments should be designed to test the efficiency and the effectiveness of our proposed approach, through a user evaluation and an assessment of quality based on process mining metrics [1]. However, it is important to apply our approach on event logs known by their variability such as the Sepsis [5] Event Log.

Table 6.5: Performance of Cluster C resources

Resources	R23	**R35**	R17	R20	R301	R10	R668	R53
Sojourn Time	11, 13, 12	**5, 6, 4**	13,12, 6	7,1 0, 8	8, 11, 15	6, 7, 19	9, 8,10	13,11, 17
Average sojourn time	12	**5**	10	8	11	10	9	13

6.2.3 Synthesis

In this section, we have proposed an UBP operational support approach. To this end, we used the refined process mining framework activities. Each activity can be related to a specific work entity. The correspondence between process mining activities and work entities is denoted as follows (System → Activities):

1. The management system → Discovery, Check, Compare and Promote.
2. The operational support system → Detect, Predict and Recommend
3. The DECoD system → Discover, Enhance, Diagnosis
4. The construction system → Produce the predictive and recommender models.

[5] https://data.4tu.nl/repository/uuid:915d2bfb-7e84-49ad-a286-dc35f063a460

We observed that each work entity outputted a specific model or data forms. For instance, after applying the Enhance activity, we obtained an enhanced process model and after applying the Predict activity, we obtained events as predictions. Therefore, the diversity of these outputs and inputs presents more challenging tasks in terms of communicating results and delivering decisions.
From a technical perspective, we suggest the use of these two research works [135] and [139] respectively published in 2016 and 2018. They propose different methods, to consolidate discovered process models.
All in all, the approach proposed here has never been considered elsewhere, to be more specific, it brings new knowledge to the research field in terms of providing a complete scenario of the process mining activities.

6.3 Variability

The contribution presented in this section is about mining self-defined BPs [21]. Indeed, a suitable technique for representing users' behaviour in their information seeking processes is required. In this sense, process mining seems to be an interesting option to deal with. Also, managing self–defined BP variability is essential (Configurable Process Model). Furthermore, it is important to handle semantic content (Ontologies based on semantic reasoning) to recommend one process in the case of variation points.

a- Configurable Process Model (CPM): appeared with the objective of integrating different process variants into one model [140]. Thus, the CPM enables extracting a process variant, which is a process model different from the original one, but that fits better in the application environment. This approach enables to represent the commonalities of the process variants. By sharing the particularities among multiple variants, this approach also promotes the model reuse [141]. Several aspects related to the BP variability have been discussed, such as: management, (re)design, modelling, and configuration. Furthermore, most of the proposed approaches present a low level of automation. Indeed, the configuration of the process variant requires the verification of a syntactical and semantically levels of resulted models, where existing approaches do not differ between planned execution and real process execution, i.e., what happens during the process execution may be not planned to happen. Therefore, the use of process mining techniques is mandatory because they enable the extraction of information from event logs. Thus, by analysing the generated process model, process variants can be discovered, and problems can be corrected. For this purpose, a process mining technique, called Decision Miner, is selected to analyse decision points that enable detailing variation points, alternatives, and rules. The benefits provided by the semantic enrichment of the BP include the improvement of its representation and understanding; the automation of tasks related to the modelling, configuration, evolution, and the adaptability of the BP according to different requirements. Therefore, it is possible to analyse the CPM in a semantic manner.

b- Ontologies and semantic reasoning: The ontology enables to capture, represent, re (use), share and exchange common understanding in a domain [37]. The ontology is composed by commonly agreed terms, thus describing the domain of interest. However, knowledge shared and

reused among applications and agents are only possible through the semantic annotation. Semantic annotation enables to reasoning over the ontology, thus ensuring the quality of the ontology by deducting new knowledge [142].
The semantic enrichment of the BP was proposed to increase the level of BPM lifecycle [143] and to compliance checking [144]. Regarding to the CPM, semantic technologies have been applied for semantic enrichment [107] and for semantic validation [145].

c- Self-defined BP challenge: The difficulty of representing self-defined BP emerges from its variability [108]. The later changes according to different contexts and requirement. Even though managing process variability is a non–trivial task because it requires specific standards, methods, and technologies, it still involves many parameters that are not always formally defined. For example, designing the reference process model, which represent the commonalities from the process family, is a challenge, as well as the necessary adjustments to configure a specific process variant. To overcome these challenges, it would be useful to represent users' behaviour (information–seeking processes), i.e., to define the generic process model, in order to study the self–defined BP variability and recommend the suitable path to each user. Also, it is useful to manage the process variants through ontologies based on semantic reasoning and Configurable Process Model [140], i.e., to select the appropriate process variant according to the combination between different self-defined BP ontologies.
To achieve these objectives, this study uses process mining algorithms to mine user self–defined BPs. Seven process mining algorithms are applied on two real datasets of users' traces interactions with the e-administration. The first step is to select the most preferment algorithm, based on process model quality criteria, to discover the generic process model. The second step aims at managing existing process variants to recommend the suitable path according to user's objective, requirements, and engine knowledge. This is done by using decision miner algorithm to obtain the CPM (process model with variants details) and by employing related semantic reasoning of the self–defined BP ontologies.
Therefore, this contribution is organized in order to evaluate the performance of process mining algorithms in representing self-defined processes and their ability to generate user's behaviours and identify variation points, alternatives, and rules. Besides, it illustrates the applicability of semantic reasoning as a decision task that can be combined with process mining, to recommend one path instead another, in the case of interactions between users and e–administration services. In section 6.3.1, we detail the methodology to mine and manage variability of self-defined BPs. Also, this section details how it is possible to select the suitable process disocvery algorithm to the self-defined BP type. In section 6.3.2, a concrete case study is depicted to simulate the proposed approach. The case study is about interactions between users and electronic services of e-administration with self-provided processes, where citizens' service requests can proceed through self–defined processes. The section 6.3.3 summarizes the main section and introduces future research.

6.3.1 Overview of Our Mining Self-defined BP Approach

Self-defined BPs are characterized by variability. This puts forward the difficulty of representing this BP type. Therefore, we propose an approach for representing and managing self-defined BP. Indeed, the applicability of process mining and semantic ontologies, to represent users' behaviour in their information-seeking process is required. In this context, the mining self-defined BP approach consists of three main steps (see Figure 6.3.1).

a-Data preparation: The self-defined BP approach starts by the event logs preparation step (part a of figure 6.3.1). In this step, it is important to pass through the extraction and the filtering operations, to obtain cleaned event data. In this sense, we focus on two main filters: The category filtering [146] and the deficiencies filtering [47]. The first ignores data with forbidden access, while the second ignores noise (ignores data incorrectly logged) and chaotic (ignores data logged by the system even though it is not part of the main process flow) behaviours. Accordingly, we can proceed to the discovery algorithm selection step (part b of figure 6.3.1). This step aims at evaluating the capacity of each process discovery algorithm for representing self–defined processes. This evaluation can be released based on process models quality criteria. This phase's output can be used in the managing variability step (part c of figure 6.3.1). To do so, we define ontologies and apply the decision miner algorithm. These two operations are required for semantic reasoning. At this stage, it is crucial to detail paths variants (CPM) using events gathering knowledge and ontologies, to determine the cluster of paths (processes with the same user objectives and requirements) on which the refinement activity can be released to choose one path instead another. This is done by the reasoning engine activity that can combine the resulted CPM with the level of knowledge ontology (semantic reasoning). This recommends a unique process for each user according to the inputs' data. Thus, the unicity of the resulted process is checked. In the case of multi–processes result, the reasoning step restarts. All these approach steps will be detailed in the following sections.

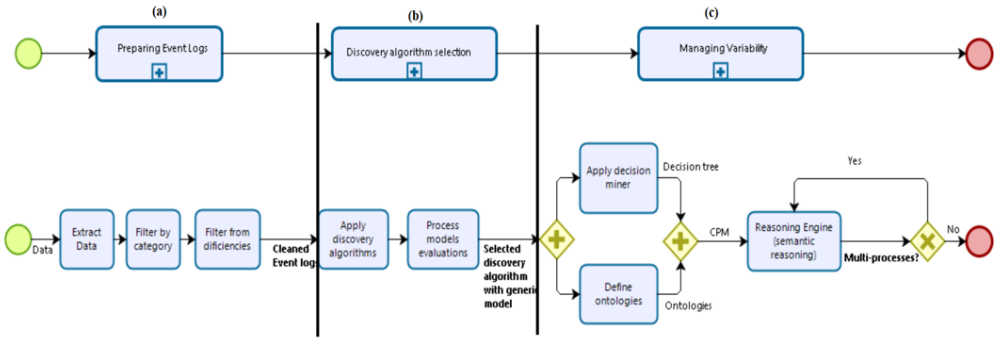


Figure 6.3.1: Approach for mining self–defined BP

b- Discovery algorithm selection: In our approach, a process discovery algorithm constructs a generic process model based on event logs. Indeed, the generic model is an abstracted and general representation of real event logs. Several discovery algorithms are described with basic representation of process models, like alpha algorithm. Other algorithms are representing differ-

ent abstraction levels combined with clustering and classification techniques, to model processes from unstructured and complex events. In this regard, we conduct a comparative study to nominate the most preferment discovery algorithm to represent self-defined processes. In this regard, we are inspired by [133, 147] to list the following process discovery algorithms: Alpha ++ algorithm, Heuristic Miner (HM), Inductive Miner (IM), Genetic Miner (GM), Fuzzy Miner (FM), State Based Regions (SBR), Language Based Regions (LBR).

On one hand, the Alpha ++ detects non–free choice relation by describing activities of the selected relation that depends on other activities. It cannot detect invisible tasks. Therefore, this algorithm gives unsound results. In this sense, an extended version of the alpha algorithm has been created, to take into consideration the patterns' frequency. Indeed, the HM algorithm can discover main behaviours and abstract exceptional and noisy ones leaving out less important activities. This later cannot group traces with sub-logs representation. Accordingly, the IM algorithm has been developed to treat events by grouping them into sub-logs. For each sub–log, a sub-process is generated. Then, a combination between the resulted sub–processes are released to obtain the generic process model. In this respect, the IM algorithm produces sound models, i.e., less none–conformities detected and it fits with the majority of present logs. Besides, it cannot identify complex and non–local process control patterns.

On the other hand, new algorithms have been developed to treat event logs in their uncertainty, for example the GM algorithm. This algorithm uses the genetic concept in creating process models from logs. This is done randomly. For each process, the precision metric is calculated. Then, sound models are combined based on the mutation operation. The main limitation of this approach is their complexity in discovering and representing process models from real data sets. From the same complexity standpoint, the FM deals with unstructured processes. In this sense, FM simplifies unstructured processes by preserving significant behaviour, while less significant but highly correlated behaviours are aggregated into clusters, and less significant or less correlated behaviours are abstracted.

Furthermore, the SBR algorithm generates a Petri net from a Transitions System (ST) based on specific abstractions, such as: Set, Multi-Set, Sequence and other types of abstractions, in which each state of the ST can be represented by a complete or partial trace. This algorithm ensures the fitness metric, as well as the identification of complex control structures. Besides, SBR is unable to process incomplete and noisy logs, while the LBR algorithm can find process model places based on the language process. Indeed, the LBR algorithm uses properties derived from logs (causal relationships), to determine the final model by describing different places. Unfortunately, this algorithm is unable to process incomplete and noisy logs.

c- Managing variability: To overcome the variability challenge, this work manages the process variants through ontologies based on semantic reasoning (see Figure 6.3.2). we propose to select the appropriate process variant according to the user's objective, the user's requirements, and the engine level of knowledge. To this end, the semantic annotation aims at reasoning over ontologies, to recommend final decision to each user.

Analysing event logs by process mining techniques provides all process instances. The instances properties can be used in the process model generation. At this stage, it is important to detail the

generic model into a CPM emerged with the objective of integrating different process variants into one model.

The CPM can be obtained by applying the decision miner algorithm. In this sense, decision point analysis allows identifying variation points (parts of the model that are subjects to variation), alternatives (available for the variation points), and rules enabling to choose one path instead another. By identifying these aspects, the process variants can be extracted.

From a technical standpoint, the user provides information related to daily objectives and requirements. Indeed, the questionnaire-model approach [106] is applied to guide the configuration process. Here, each variation point is associated to a question, whose alternatives determine the path selection. Thus, by selecting an alternative related to a question, the system configures a process variant. In the proposed approach, the questionnaire is developed using the decision point analysis.

After discovering process variants, it is required to combine them with ontologies. Here, a focus is given on self-defined BP factors, to define ontologies, which are: 1. Objective ontology, 2. Requirements' ontology and 3. Level of knowledge ontology. Respectively, variation points, available alternatives and rules related to citizen and e–administration interaction must be formalized. Therefore, process mining can enrich ontologies definition by extracting knowledge from event logs in order to recommend one variant to each user. In this context, figure 6.3.2 illustrates the proposed approach for managing variability into self-defined BP.

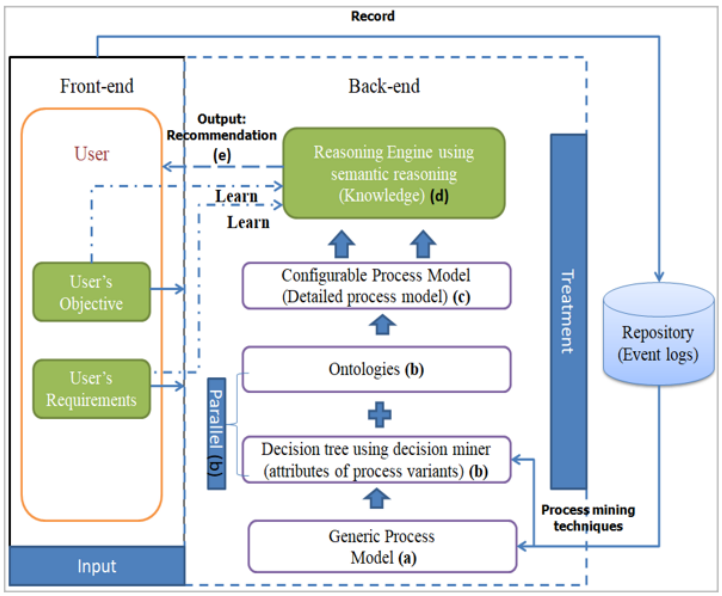


Figure 6.3.2: Approach for managing self-defined business process variability

The figure 6.3.2 presents two aspects: the front–end aspect and the back-end aspect. The first aspect concerns the user's inputs, which are the user's objective and requirements. These two points will lead, later, to obtain the final decision process. The second aspect encompasses the analysis and the engine parts. The analysis part details the generic model and the CPM, based on historic data, ontologies, and the decision miner algorithm (Decision tree). Consequently, the

self-defined BP ontologies, guidelines and process variants attributes are emerged. In this context, the use of semantic reasoning [109] is required, to recommend the suitable process that the user must be achieved, based on the engine knowledge, and learned from the user's objective and requirements.
In this way, when the user (Front-end) provides an information related to the user's objective and requirements, a cluster of activities is performed. In addition, the knowledge of the reasoning engine will be used to recommend the unique process to achieve.
For example, when the user asks for a document, usually the first operation is the evaluation of the document type. Thus, some required information must be performed, such as, identity, reason of the request, etc. The objective and requirements presented by the user determine multi-processes. Their refinement must be released with ontologies based on semantic reasoning. In this regard, the unique result must be enriched by the level of knowledge ontology (Reasoning Engine), according to these steps (a), (b), (c), (d) and (e) of figure 6.3.2.

6.3.2 Illustrative case study

In this section, an example about citizens in interaction with the Electronic Administration is presented, to illustrate the applicability of the mining self-defined BP approach (see Figure 6.3.1).

a- Preparing event logs:

The definition of e-administration or electronic administration refers to any of the mechanisms, which are meant to transform what in a traditional office, based on paper processes, into a paperless office based on electronic ones. This is an ICT (Information and Communications Technology) tool intended to improve productivity and performance.

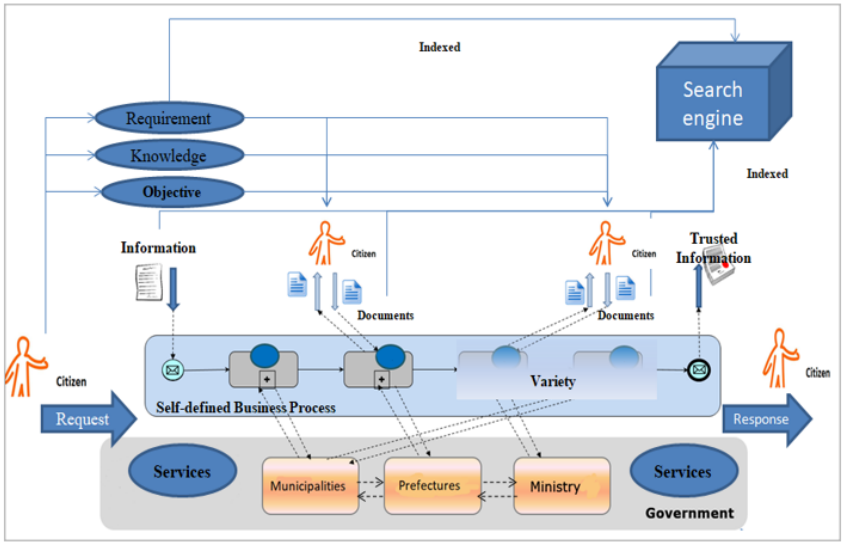


Figure 6.3.3: Interactions between e-administration and citizens (example of e–government)

Administration processes are characterized by the fact that several organizational units (community, municipalities, etc.) can be involved in the process treatment of citizens. These organizational units often have their own specific IT applications; it becomes clear that getting data, which is related to e-administration processes, is not an easy task because the latter are characterized by variability. Indeed, this sub-section aims at extracting and filtering events, in order to prepare data (part a of figure 6.3.1) for next steps.
The process between citizen and government is a particular example of electronic administration. In this sense, figure 6.3.3 illustrates interaction process between the citizen and the government that can be partially predefined since this interaction between them depends on specific objective, level of knowledge and requirements. This can vary the process and produce different user behaviours that cannot be predefined. These processes need advanced analysis to understand user's behaviour in their interaction with the e-administration services.
As shown in figure 6.3.3, the e–administration is a system that allows navigating through large volumes of administrative documents created in interaction with other parties. These documents are often heterogeneous. The information search activity carried out by users is a process [148] that relies mainly on search engine queries, filters and consulted documents. These elements depend on users' requirements, types of information handling strategies, as well as the way how data are indexed. Indeed, the use of a user–guided system, in relation to electronic administration, produces many processes with different variations and self–identification. These two characteristics describe self–defined BPs.
Generally, as mentioned in [149], electronic services related to e-administration are defined in three categories: public, voluntary, and private. The private events are with forbidden access (accessible only by the data owner) like services between the citizen and the police entity. The public and voluntary services are accessible to the data analyst. Therefore, this case study focuses on two event logs categories: public and voluntary.

Table 6.6: An example of event logs

Case_ID	**User**	**Date**	**Activity**	**Category**
1	Citizen1	2018-05-17|14:23:25	Trustprofil	Public
2	Citizen2	2018-05-17|14:24:25	Trustprofil	Public
1	Citizen1	2018-05-17|14:24:26	Scroll	Public
1	Citizen1	2018-05-17|14:24:28	ResourceAccess	Public
3	Citizen3	2018-05-17|14:26:26	request	Voluntary

According to the data volume, some filters must be applied to obtain clean and specific data (see Table 6.6). Indeed, we use two filters: The first filter aims at ignoring forbidden [146, 150] noise and chaotic data. The second filter aims at filtering events on specific period. A focus is given to the period defined by behaviours with high level of variability. This level distinguishes self-defined BPs structure (see Figure 6.3.4).

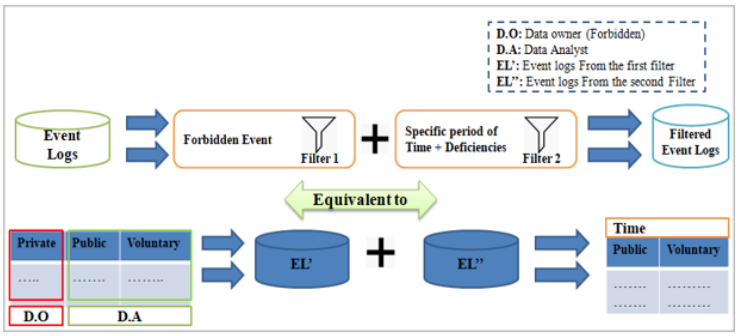


Figure 6.3.4: Event logs preparation

In this work, we use raw data collected by the IR of electronic services. The data contain information about a group of interaction between citizens and the e-administration (municipalities, prefectures, and ministries) treated in 2018, for which all diagnostic and treatment activities have been recorded. The data distinguish two events logs categories that encompass a set of activities for both public and voluntary categories. The public category contains 110 traces, 1755 events, while the voluntary category contains 39 traces, 1570 events. Indeed, the logs in the public category contain 110 process instances (traces) of 80 variants. 10 traces appear more than once while 70 are unique. In the logs of the voluntary category, there are 39 unique traces. This is an important property of self-defined processes, where recurrent traces are rare and not very redundant. Process discovery algorithms have the advantage of finding generic models from non-redundant traces. Another important aspect of user self–defined process is the repetition of an activity within the same trace. Therefore, long traces can be generated.

b- Discovery algorithm selection and process evaluation:

After the data preparation step (part a of figure 6.3.5), we proceed to the process discovery algorithms application (part b of figure 6.3.5), in order to select the suitable algorithm for the self–defined BP representation. Therefore, in this sub-section, we compare and evaluate each resulted process model with the aforementioned quality criteria (part c of Figure 6.3.5), to deduct the most generic, representative, and performing one, by necessity defining the suitable discovery algorithm related to the e-administration citizen's behaviours (part b of Figure 6.3.1).

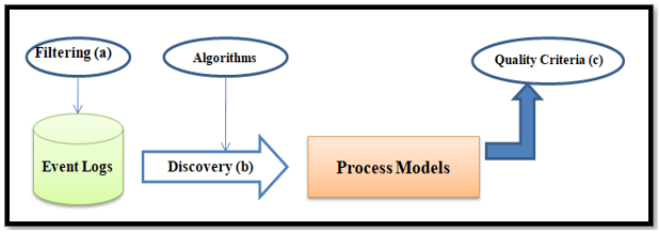


Figure 6.3.5: The Processing steps

To discover the generic process model and to evaluate its quality, we use the ProM tool. In this sense, the fitness is calculated using the "PNetReplayer" package, while the Precision and the

Generalization metrics are calculated using the "PNetAlignementAnalysis" package.
All discovered models, using HM, IM, GM, LBR, SBR and Alpha++ algorithms, must be transformed into a Petri net form, to check model's conformance. The simplicity of these process models is not checked in this chapter. To this end, Petri net representations are evaluated with three process model quality metrics: fitness, precision, and generalization, while the fuzzy miner representation based on two metrics: Node detail and conformance.

Table 6.7: Fitness, Precision and Generalization for all models

	Public			**Voluntary**		
	Fitness	**Precision**	**Generaliz.**	**Fitness**	**Precision**	**Generaliz.**
HM	0.00	0.00	0.00	0.00	0.00	0.00
IM	98%	24%	99%	93%	15%	99%
GM	99%	18%	99%	62%	80%	99%
LBR	62%	38%	97%	78%	19%	96%
SBR	90%	42%	99%	96%	29%	99%
Alpha++	0.00	0.00	0.00	0.00	0.00	0.00
FM	—	—	—	—	—	—

The Petri net discovered with the Alpha++ algorithm is an unsound model because it contains deadlocks and insignificant traces (few traces are generated and none of them correspond to those presented in the logs). The same limitations are observed for the models generated by the HM algorithm (see Figure 6.3.6 and Table 6.7). The heuristic model has no initial and final marking, i.e., the model has no departure or arrival points, but it is readable. However, it helps understanding links between different stages of the studied process.

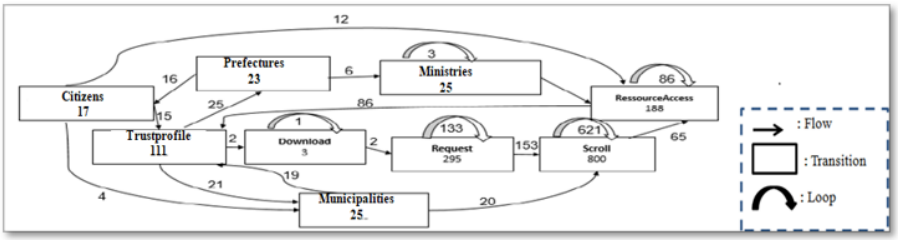


Figure 6.3.6: Heuristic Miner algorithm applied on the voluntary category

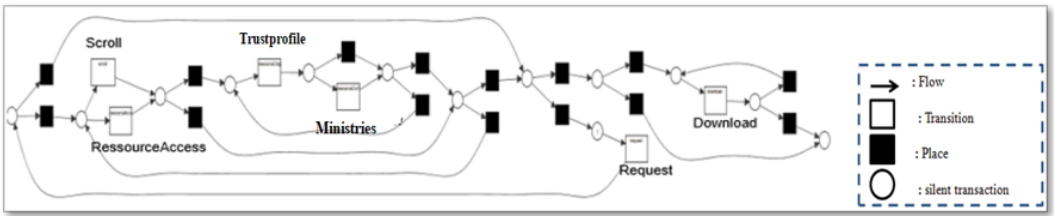


Figure 6.3.7: Inductive Miner algorithm applied on the public category

Figure 6.3.7 illustrates the process model discovered from the public logs, using the inductive miner algorithm. In this respect, the fitness value is excellent on both public and voluntary logs

(more than 90%). In addition, the obtained models have a high Generalization and a low precision (see Table 6.7). These models can detect the process start and end points. They also ignore insignificant arcs. These arcs show different users processes to accomplish their information-seeking process. This inductive algorithm gives a very clear representation of the process that is followed by citizens using e-administration services.

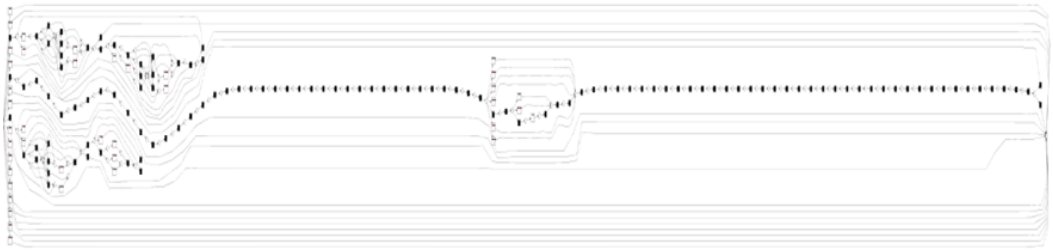


Figure 6.3.8: Genetic Miner algorithm applied on the public category

The genetic miner algorithm discovers two models, with high level of generalization. The public process model, in Figure 6.3.8, covers more logs than the voluntary process model. On the other hand, the fitness for public logs reaches 99%, while it is equal to 62% for the voluntary category. Consequently, the model discovered from the public logs are underfitting (precision=18%) and equal to 80% (for the voluntary logs). Therefore, the genetic model has a high level of precision and generalization. The inconvenient of these models is the complex representation. They do not allow determining the most important links between different activities. Thus, different user's processes, to achieve their objectives, cannot be determined. The resulted model is unreadable.

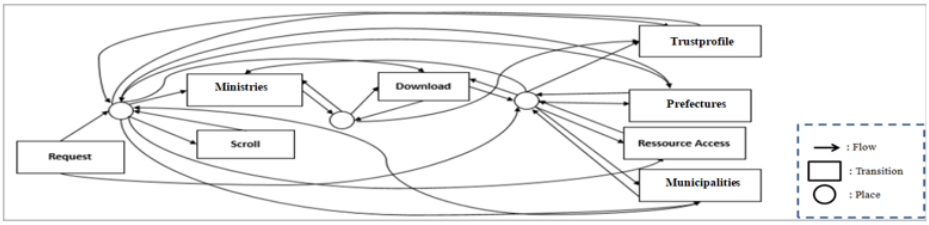


Figure 6.3.9: LBR algorithm applied on the public category

The LBR models are similar to the inductive models, in terms of the absence of frequency and different choices made by users. Also, these models are similar to the flower form, i.e., all activities are accessible from other states. Moreover, it does not represent initial marking. For both categories, the LBR algorithm produces models with high generalization and fitness rates (both cases equal to 70%). Therefore, the LBR model (see Figure 6.3.9) provides an interesting generalization. The different activities are well represented and links between them are very detailed.
The obtained models with the SBR algorithm are discovered based on the Multiset abstraction option and the choice of partial traces. The two discovered Petri nets have initial marking. The SBR algorithm also produces two models with a high level of generalization. In addition, replaying logs on the obtained models produces sound results for both public and voluntary events (Fitness=90%). The SBR models allow visualizing, very precisely, all different users' choices, in order to navigate from one activity to another. However, the SBR models are failed to represent the suitable generalization metric.

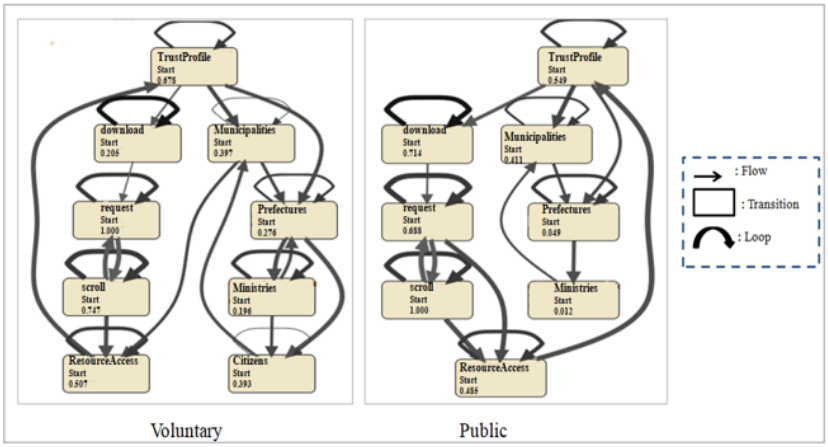


Figure 6.3.10: Fuzzy Miner algorithm applied on the voluntary and the public categories

The Fuzzy algorithm is applied on the public and voluntary event logs categories, using the Fuzzy Miner package. The Table 6.8 shows that the visible nodes, found in the two graphs, are important and significant. Moreover, replaying logs on obtained models gives sound results (conformance= 80%). Therefore, these models represent a clear and interesting generalization (see Figure 6.3.10). The frequency of users' choices is well presented and provides useful overview of the studied processes.

Table 6.8: Measures for the fuzzy miner algorithm

	Node details	**Conformance**
Public	100%	80%
Volunraty	100%	84%

To conclude, the suitable process discovery algorithm to represent self-defined BPs is the Inductive Miner. Also, the Fuzzy Miner algorithm gives sound results. For more simplification, the generic model (public and voluntary categories) is converted to the BPMN form. The process illustrates 8 activities (see Figure 6.3.11). The user has two possible ways to consult the e-administration services: via scroll mode (search for information) or via the resource access mode (search for resource). In addition, the user can be connected using the trust profile (officially generated for each citizen). According to the trust profile (citizen ID and objective), the citizen can directly contact municipalities, prefectures, or ministries. Then, the citizen can apply for administrative certifications. If the request requires the contact of other resources, citizen must go back to the trust profile activity (the citizen's profile against the requested document: owner, participant, intermediate, etc.), to get a final decision. Next, the citizen can download the paper for a limited period. Last, the citizen can log out.
In this abstracted representation, four XOR gateways are observed, while the deep abstraction level generates more process variants (see Figure 6.3.8). This puts forward the difficulty of managing self–defined BP variability.

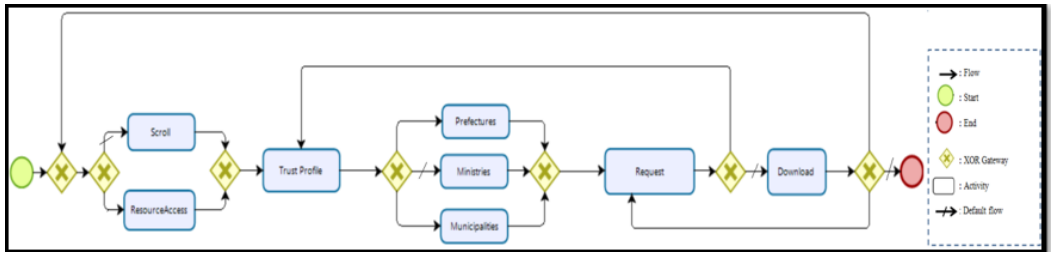


Figure 6.3.11: The generic process model with BPMN representation

c- Managing variability:

After defining the generic self-defined BP, it is important to meet the challenge of the BP variability. So, in this sub-section, an application of the proposed approach to manage self-defined BP variability (part c of Figure 6.3.1) is presented. In this context, process variant configurations through ontologies have been proposed by some authors, such as Huang et al. [151]. However, the approach proposed, in this research, enables to identify process variants, its characteristics and ontologies from event logs. In this way, the process model can be correctly individualized by meeting the requirements of the context application. Moreover, event logs reflect what is happening during the citizen and e–administration interaction and enable the process variants improvement.
To this end, figure 6.3.12 illustrates how the management approach could treat this variability. Indeed, a focus is given on the fragment of three XOR gateways, as decisions points' example of the generic process model (see Figure 6.3.8).
First, the decision miner algorithm is applied on the generic model to obtain the decision tree of these gateways and to define user's objectives and requirements ontologies. Based on these two elements, process variants are obtained. For instance, a citizen request could be a demand for simple or complex documents. The simple request is immediately accepted, while the complex one requires the trust profile status verification (owner, intermediate or participant).
According to this status, e-administration services (ministries, prefectures, or municipalities) could respond. Side by side, objectives and requirements ontologies examples are defined. The objective ontology contains trust profile class (citizen ID), status (owner or participant or intermediate) and information as sub-classes (personal information). The requirements ontology encompasses Requests, Resources as classes and documents as sub-class. In this order, the CPM is obtained. This will add three additional activities: Activity of comparing the citizen-ID (verify if citizen origins; rural or urban area in order to contact the suitable area to its situation) Activity of collecting status (take into consideration the status verification) and the activity of verifying the document type (classify the request). Next, the CPM is combined with the existing ontologies. This allows the emergence of an advanced ontology entitled: level of engine knowledge. Therefore, the correspondence between decisions points and different classes and sub–classes of ontologies is observed. In this order, the request class calls the document sub–class. This later has two choices called the trust profile sub-class or the resource sub-class, etc. Last, the semantic reasoning based on the CPM and the advanced ontology is applied. This operation takes into consideration the inputs information (request of citizen), to recommend unique process. For instance, the citizen consults the website via the ResourceAccess option. He is connecting via trust profile with a rural

ID. The citizen requests a simple document. The citizen can download the requested document. Therefore, the recommended path is: ResourceAccess →Trust Profile → Municipalities Request → Download.

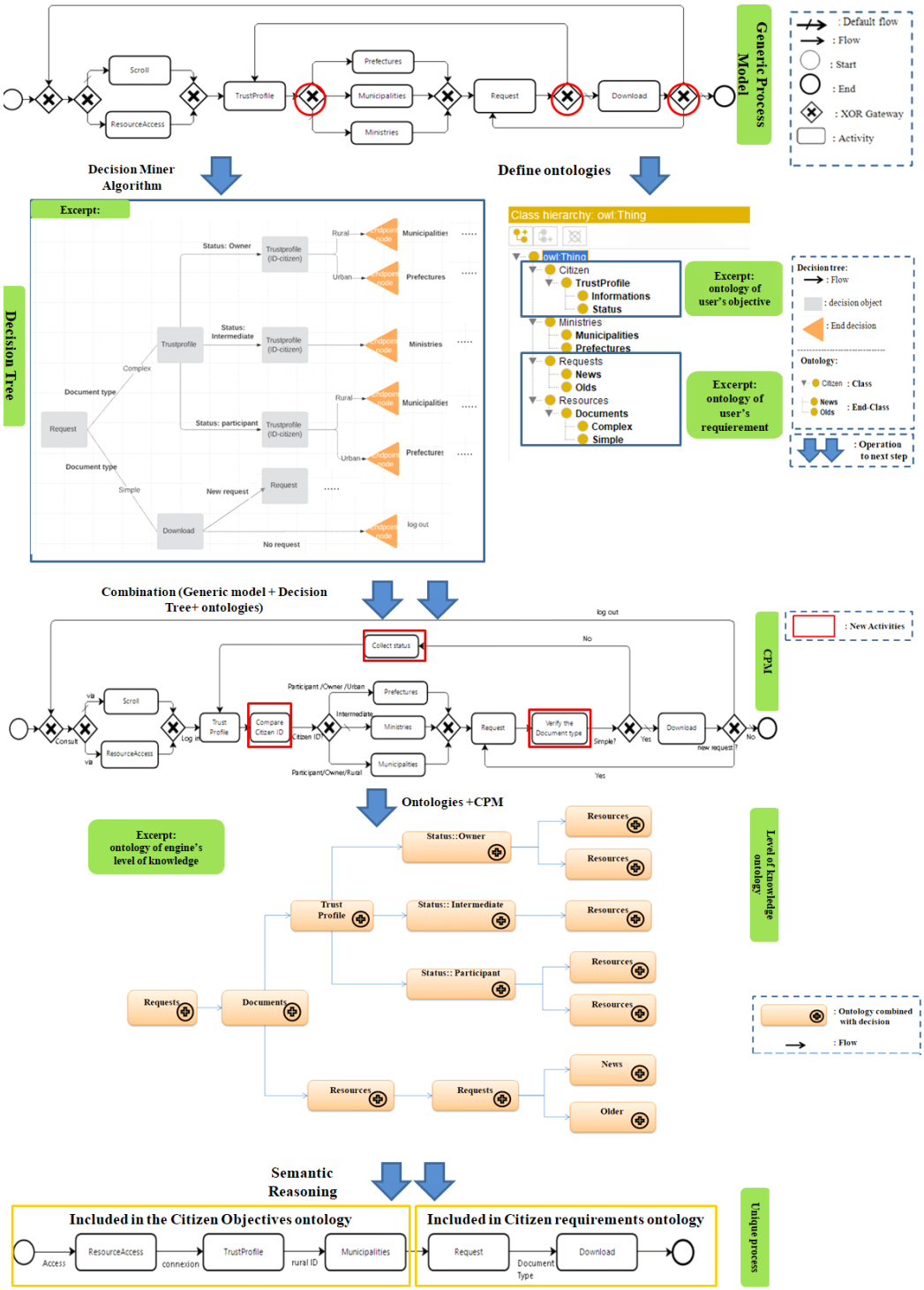


Figure 6.3.12: Selection of the appropriate process variant according to the user's objective, the user's requirements, and the engine level of knowledge

6.3.3 Synthesis

In this work, the approach for mining self-defined BPs hands over three main results. These results provide a decision–making support for the user and enable to individualize a process model that respects the user's requirements and the internal or external regulations.
The first result represents the suitable process discovery algorithm to generate self-defined BP. We applied several process discovery algorithms in order to decide between their effectiveness and their limits. In this approach, we looked at self-defined processes in the context of electronic services, to be more specific, in e-administration.
Based on the comparative study, the inductive miner achieves the best performance in terms of fitness and precision, while other algorithms cannot guarantee soundness. In case of complex events, it is necessary to use a filtering method prior to applying existing automated process discovery algorithms. The second result is about detailing the variation points of the generic process model. In the literature, the application of decision miner algorithm is used with variable, flexible and dynamic BPs as the Ad-hoc BPs example. The novelty, here, is to introduce decision miner with self–defined BP. The third result gives rise to the ontologies definition, to apply semantic reasoning and recommend suitable path to users. We extract knowledge from event logs to define ontologies, then combine them in semantic reasoning based on training samples. The added value over existing studies is the application in self-defined BP context, respecting its different properties, such as: variability, unique paths, different ontologies, etc.

6.4 Dynamicity

The concept of dynamicity defines the process that can change some BP activities during runtime under various conditions, which are emerged from real-time variables. It can be adapted according to internal or external environment changes. In this context, we focus on Ad-hoc processes that are characterized by a non-defined workflows design. Indeed, the control-flow between activities cannot be modelled in advance but can simply occur during runtime. Here, users must be able to decide what to do and when. They must also be able to assign work, as sub-process, to other people and create interactions among various users. This puts forward the difficulty of treating dynamically Ad-hoc processes.
In this way, our work aims at introducing dynamicity concept into Ad-hoc BP, using process mining techniques. We try to demonstrate how it is possible to extract in-formation from event logs (Off-line), using the process discovery technique, in order to model and configure Ad-hoc BP. Then, we demonstrate how to select suitable Ad-hoc sub-processes adaptively to specific conditions at run-time (On-line), using the conformance checking technique. The enhancement technique can be applied for re-modelling and re-configuring the Ad-hoc BP.
For this purpose, this section is organized as follows: In section 6.4.1, we detail our approach of selecting dynamically an adaptive sub-process using process mining techniques. In section 6.4.2, a concrete case study is depicted to simulate our approach. The case study is about an insurance claims handler system in a smart city environment.

6.4.1 Overview of our BP Dynamicity Approach

This contribution consists of treating dynamically Ad-hoc BP, using process mining techniques. On this subject, we try to model Ad-hoc BP sub-processes based on historic events knowledge and to select an adaptive sub-process using running events. In this respect, we present our approach in two views: Off-line and On-line (see Figure 6.4.1). In the Off-line view, we analyse event logs in order to discover the generic process model. We can also determine the full Ad-hoc process model by attributing business conditions and rules to the generic process model. In the On-line view, we try to adapt our Ad-hoc BP, i.e., dynamic selection of an adaptive Ad-hoc sub-process according to specifics conditions. These conditions can be defined as a cross-environmental variable that includes business conditions and rules. In this context, **business conditions** are considered as decision points for each X-OR gateway. **Business rules** are expressed as a set of rules prescribed as a guide for actions or business behaviours. **Cross-environmental variable** passes external conditions to internal Ad-hoc BP environment; it may combine business rules, business conditions and optimal performances as a resource, time, etc. To approve these three points, we apply the conformance checking technique.

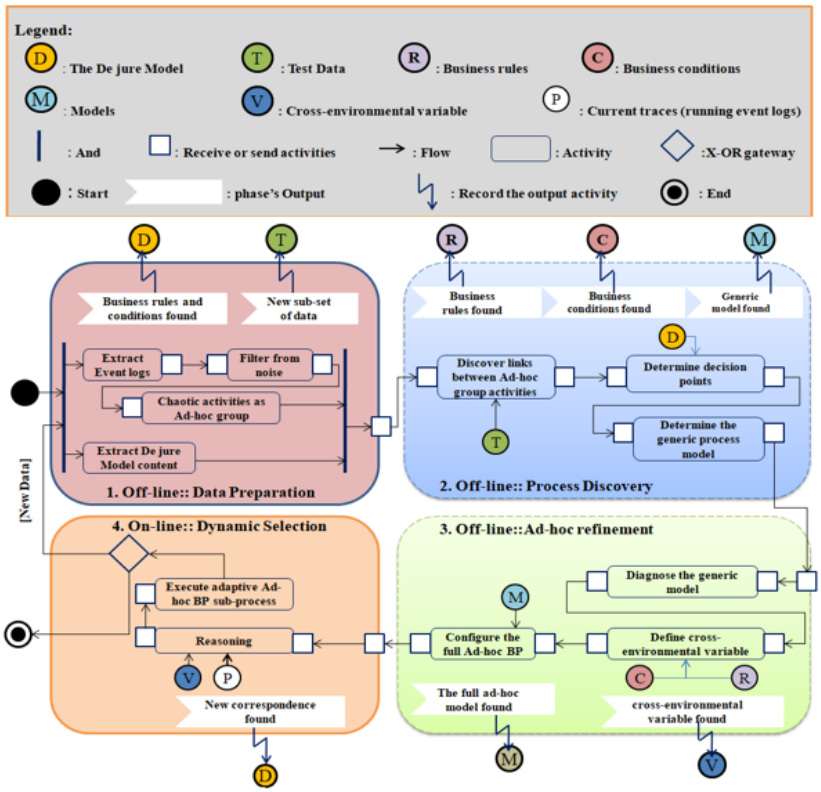


Figure 6.4.1: Different views of our proposed approach

a- Off-line:

We start our approach by the data preparation phase. In this regard, the initial input of our approach is an event log.

Generally, an event log is generated, while the process execution history is stored in the database. Adequately, the classic ETL (Extraction, Transformation and Loading) process is employed for event log preparation. In this work, Extensible Event Stream (XES) and Comma Separated Values (CSV) are used. Indeed, database is presented as a CSV file or with XES format. Now it is ready to be filtered and reorganized. Parallel, we explore the De jure model content. This step aims at determining pre–defined business boundaries. Indeed, this phase's outputs are a set of event logs for test and a document (De jure) that compasses business rules. In this context, business rules are responsible on workflow orchestration. Besides, data test contains filtered data from noise [47] and the Ad–hoc group. This group includes chaotic activities, i.e., they can be executed anywhere in the process [3].

The second phase is nominated as a process discovery operation. This phase aims at discovering the generic process model. The generic model defines all possible behaviours (sub-processes) that can handle an Ad-hoc BP (see Figure 6.4.2). Here, the selected discovery algorithm is heuristics miner because it focuses on the frequency of patterns. Thus, the main pattern of behaviours can be extracted, and its output can be converted to BPMN. In this regard, the most often applied notation for representing dynamic behaviours is BPMN. Indeed, heuristics miner stills a suitable algorithm to discover Ad–hoc processes.

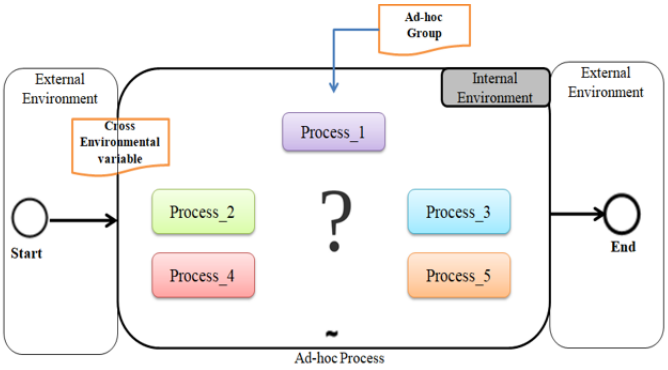


Figure 6.4.2: Dynamic Ad-hoc Business Process

In this phase, we introduce the frequency concept, which aims at determining the representative BP for each frequency degree. The frequency values are implicitly changed when the produced BP is changed. For instance, the values 80% and 70% are producing the same process model. This process model defines a specific category A (sub-process). If the produced model is contently changed, we define a new category B (sub-process). In this example, we assume that we have two process model variations.

To refine the generic model with business conditions and rules, we use the decision miner algorithm [6]. For the business rules' representation, we propose three possibilities which are correct pattern, incorrect pattern and indirect flow. This will be related to the resource, the time and the antecedent executed activity. This approach is inspired by the Oryx editor process [152]. The Correct and Incorrect pattern icons can be attached to sequence flows of a process model to define rules, whether the behaviours must be done (correct pattern) or must not be done (incorrect

[6]http://www.processmining.org/prom/decisionmining

pattern). Indirect flow is created to define an additional sequence flow, which does not exist in an imported process model. Indirect flow gives the meaning of indirect following. If an indirect flow connects activity AA to BB, then BB has to follow AA but BB does not need to immediately follow after AA. For example, AA can be followed by CC and then BB, and it still complies with the rule. This phase's outputs are full-ad hoc BP, business conditions, and rules.
Third, we proceed to the Ad-hoc refinement phase. This phase's outputs are cross-environmental variable and a categorized model. We define cross-environmental variable values. These values are used to determine, later, the adaptive Ad-hoc sub-process for execution. This variable contains different elements: available resource, optimized duration to execute a specific sub-process, the previously executed activity (antecedent of the fixed part or the external environment) and a specific business goal for achieving each Ad–hoc BP part or goal. For instance, category A is described with a resource Rs1, who is executed the category A of part B at the time (t1-t3), with a specific goal entitled Sp1. In this work, we define category as the group that often-assigned similarity in the defined activities or criteria.
Last, we aim to combine each frequency degree with a specific set of business conditions and rules, to define properly different Ad–hoc BP sub–processes and objectives.
After the cross–environmental variable values definition, we obtain the full Ad–hoc.

b- On-line:

After obtaining the full Ad-hoc BP with the cross-environmental variable, we can determine which sub-process must adaptively be executed during the dynamic selection phase. This is done by respecting certain boundaries of the cross-environmental variable values.
For this purpose, we use the most famous mechanism to check business rules, which is the Complex Event Processing (CEP) mechanism. This mechanism is implemented in the conformance checking technique. The CEP [153] can detect whether there is a violation with certain rules or conditions (cross–environmental variable values) by analysing the logs of BPs execution. Generally, the CEP mechanism is used to monitor generated rules in real–time. Interestingly, the CEP engine can be used in an On–line setting for backward compliance checking. In this work, CEP is selected due to its ability in analysing high volume of event series and for al-lowing customized triggers for detecting violations among events, to ignore no matched Ad–hoc BP sub–processes. After checking conditions, we can select a suitable Ad–hoc BP category. For instance, executing category A with Rs1 resource during t1–t3 gives insights on the selected category during the On-line view.

6.4.2 Illustrative case study

In this section, we present a BP of an insurance claims' handler system. This system makes sure that claims are handled efficiently and that payment for valid claims is made. Thus, this process consists of making decisions on the extent and validity of a claim. It can also check for any potential fraudulent activity. At the beginning, the claim is received, via an alert message, from the smart house [154] security system. Then, the alert message notifies different destinations as SOS and insurance systems about an emerged alert. At the end, systems treat the alert and respond to the notification alert.

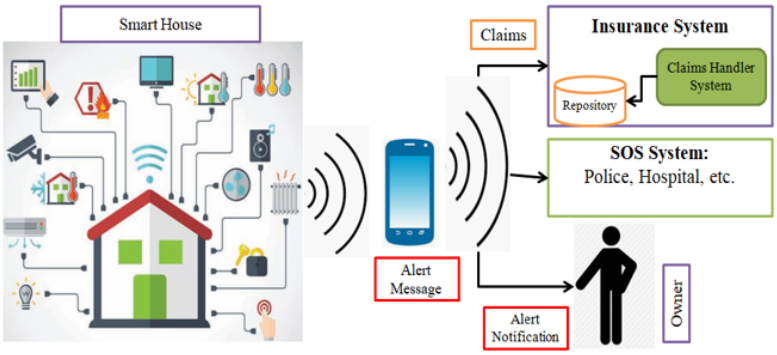


Figure 6.4.3: Claims handler system

In this case study, we use a dataset that comprises event logs regarding the activities of daily living performed by several individuals. The event logs are derived from smart home sensor data, which is collected in different scenarios and represented activities of daily digital claims customer (claimant). These scenarios include fire claims, water leak claims, etc. In this sense, event logs represent claimant' behaviours.
According to our proposed approach (see Figure 6.4.1), the Off–line view aims at preparing event logs by filtering them from deficiencies as noisy data. Then, it allows constructing the generic BP and the full Ad-hoc BP (all claimants' behaviours) based on:

- Applying the process discovery technique in combination with the frequency concept on events.
- Refining the Ad-hoc BP by attributing business conditions and rules to it. Business conditions are established to distinguish different categories (sub–processes) of the full Ad–hoc BP, to decide in dynamic manner the adaptive Ad–hoc sub–process and to ensure its modelling phase in next stages.

a- Data preparation:

To prepare event logs, we should extract logs from database and fetch them into the transformation function. Due to the difference of event logs format, a converter is needed for the purpose of transforming XML logs into CSV format. Event logs consist of various attributes for each case. However, the needed attributes involve case identification, event name and timestamp. After extracting the required attributes and transforming logs in CSV format, the CSV format is loaded into the PROM tool and converted into XES format for evaluating the proposed approach.
In our case, event logs are already prepared; we download logs from the source `(http://www.processmining.org/event_logs_and_models_used_in_book)`, then we import them in PROM for manipulation. The claims log has 8 different activities (check if sufficient information is available, AC: Assess claim, DLC: Determine Likelihood of Claim, RC: Register Claim, IP: Initiate Payment, CC: Close Claim, ACR: Advice Claimant on Reimbursement and E: End), 132 process instances and 1642 executed events. It means that over a specific period, PROM has investigated 132 cases of claims and executed 1642 tasks.
We observe that the 7 activities: RC, DLC, ACR, CC, IP, E are considered as chaotic activities (see

Table 6.9). These activities are executed anywhere in the process. For instance, RC= 0, 5, 6, ... According to the Ad–hoc definition, they are arranged into an Ad-hoc group, in order to discover the relationship bringing together these Ad-hoc activities into the process discovery phase.

Table 6.9: Excerpt of claims handler system Log summary

Case_ID	***Activities***	***The activity index (chaotic activities or Ad–hoc group)***
1	*<CC,DLR,AC,ACR,CC,IP, E>*	*RC=0, DLC=1, AC=2, ACR=3, CC=4, IP=5, E=6*
2	*<E,CC,AC,ACR,DLC,RC,IP>*	*RC=5, DLC=4, RC=2, ACA=3, CC=1, IP=6, E=0*
3	*<AC,ACR,DLC,E,IP,CC,RC>*	*RC=6, DLC=2, AC=0, ACR=5, CC=3, IP=4, E=3*
...	...	...

b- Process Discovery:

Heuristic miner is applied to discover a process model from event logs. The configurable dependency parameter of this algorithm is used to measure relationship between activities. A high value denotes a strong dependency, while a low value denotes a weak dependency. In this case, the main behaviours of the given real–life log are focused, therefore high dependency values are considered. They are experimented to review the output process models. Heuristic miner discovers first a heuristic net, and then the heuristic net is converted into Petri net. Lastly, Petri net is transformed into BPMN.

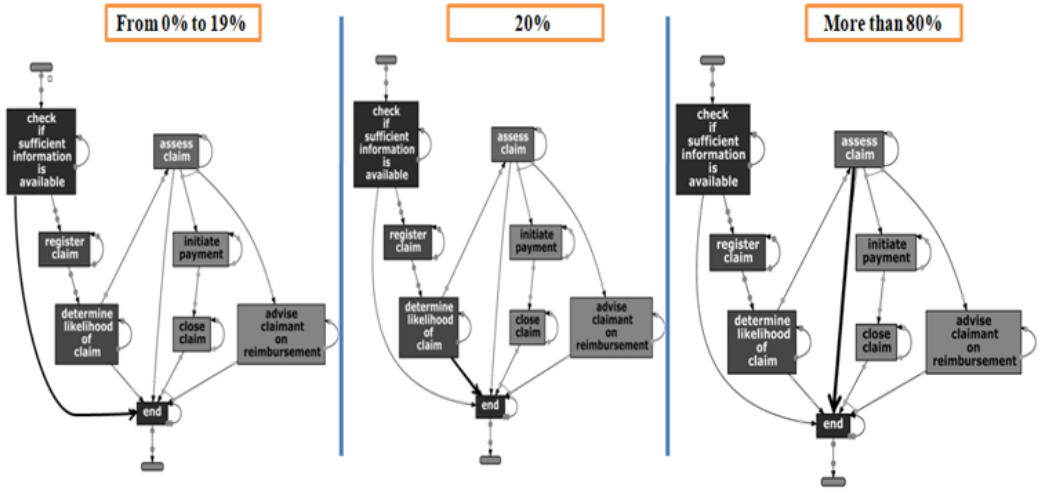


Figure 6.4.4: Discovered process models according to the frequency degree.

For our claims' handler process, we are based on the following logic (see Figures 6.4.4 and 6.4.5): First, we apply the heuristics miner plug-in. Second, we try to discover all possible process models from these event logs by balancing the frequency scroll. Hence, we determine three intervals,

where process models are unchangeable (bold arrow). In this context, we obtain three models: the first model represents more than 80% of users' behaviours. The second model represents 20% of users' behaviours. The third model represents less than 20% of users' behaviours.
For our claims' handler process, we have three different degrees of frequency, which are described respectively to two main processes: an Ad–hoc BP (instead of Ad–hoc group) and a fixed activity (check if sufficient information is available).

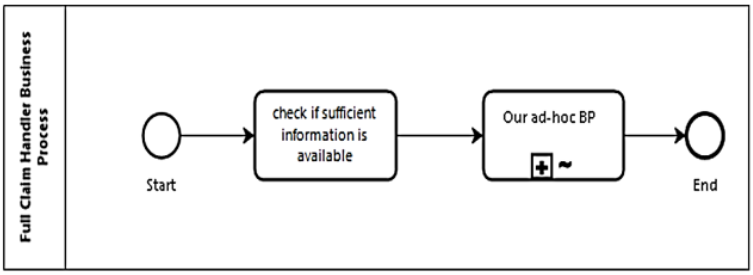


Figure 6.4.5: Generic process model representation

The Ad-hoc sub-processes are detailed in figures 6.4.6:

- The model with the frequency equal or more than 80%: we can finish the BP from the Assess Claim Activity.
- The model with the frequency equal to 20%: we can finish the BP from the Determine Likelihood for Claim Activity.
- The model with the frequency varies between 19% and 0%: we can finish the BP after applying the fixed activity (check if sufficient information is available).

Next, we can apply the decision miner algorithm. This is done for defining decision points of this process model.

c- Ad-hoc Refinement:

After obtaining the generic process model (fixed part + Ad–hoc BP), we define categories specifications for the Ad–hoc BP part (see Table 6.8). It can be nominated as the full Ad-hoc BP (see Figure 6.4.6). These categories are defined relatively to the resource condition (who is the responsible for executing this category?), the time condition (when the sequence of activities is executed?) and the antecedent condition (what is the previous executed activity?).
In addition to the generic BP goal, we define a cross–environmental variable (set of conditions) with a specific business goal is mentioned, i.e., what is the reason to execute this category?
In table 6.10, we define three categories: A, B and C with the following conditions:

- Rs1 nominated as responsible for executing the category A at the instance T1, when the Ad-hoc BP can be achieved after the Assess Claim activity.
- Rs2 nominated as responsible for executing the category B at the instance T2, when the Ad-hoc BP can be achieved after the Determine Likelihood for Claim activity.

- Rs3 nominated as responsible for executing the category C at the instance T3, when the Ad–hoc BP can be achieved after the check activity.

After defining the Ad–hoc BP categories and its business conditions, we proceed to define business rules for each category. To do so, we insist on correct pattern, incorrect pattern and direct flow notations (see Figure 6.4.5).

Table 6.10: Excerpt of the cross-environmental variable values

Categories	Conditions			Model Frequency (Fq)	Specific goal (Sp)
	Resource	Time	Antecedent		
Category A	Rs1	t1 to t3	A	Fq1 (0,9)	Sp1
Category B	Rs2	t4 to t5	A	Fg2 differs from Fq1 (0,2)	Sp2

In this example, we determine 14 rules: There are 6 rules for the category A, 7 rules for the category B and 1 rule for the category C. There are 9 correct rule patterns and 5 incorrect rule patterns. The process model with defined graphical rules is demonstrated in Figure 6.4.6.

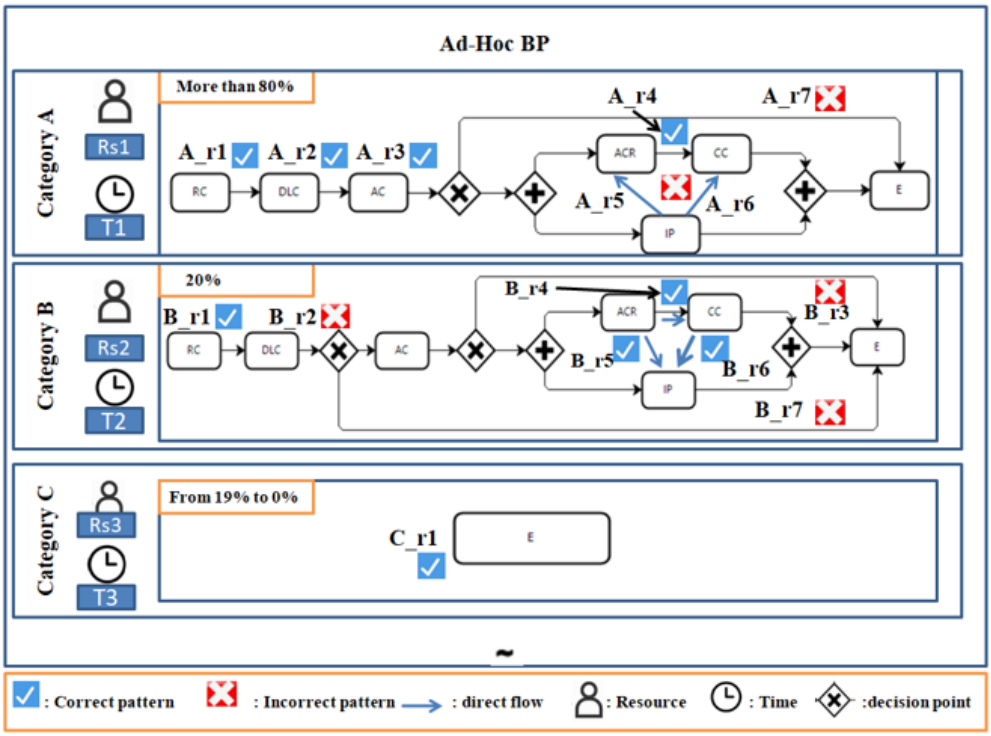


Figure 6.4.6: The full Ad-hoc BP.

At this stage, we can translate graphical rules into a rule language. In this work, Esper rule is selected because it is an open-source Java-based framework, and it is commonly used for CEP to analyse event series for detecting situations among events, which can be used for compliance

checking. Esper is expressed in Event Process Language (EPL), which is a SQL-like language with for example SELECT, FROM, and WHERE clauses.

Table 6.11: Excerpt of rules definition and creation

Business rules	**Definition**	**Business rules creation**	**Categories**
A_r1	DLC must immediately follow RC	INSERT INTO A_r1 SELECT m.event AS mEvent, m.processId AS mProcessId, n.event AS nEvent, n.processId AS nProcessId FROM pattern [every m= RuleCheck(event = 'RC') ->(endEvent =RuleCheck(event = 'End', processId = m.processId)) and not n=RuleCheck (event = 'DLC', processId = m.processId)]	**A**
B_r2	AC must immediately follow DLC	INSERT INTO B_r2 SELECT m.event AS mEvent, m.processId AS mProcessId, n.event AS nEvent, n.processId AS nProcessId FROM pattern [every m= RuleCheck(event = 'DLC') -> (endEvent =RuleCheck(event = 'End', processId = m.processId)) and not n= RuleCheck(event = 'AC', processId = m.processId)]	**B**
C_r1	E must immediately follow check activity	INSERT INTO C_r1 SELECT m.event AS mEvent, m.processId AS mProcessId, n.event AS nEvent, n.processId AS nProcessId FROM pattern [every m = RuleCheck(event = 'check') -> (endEvent= RuleCheck(event = 'End', processId = m.processId)) and not n= RuleCheck(event = 'E', processId = m.processId)]	**C**

The EPL syntax described as follows:
The INSERT INTO clause is recast as a means of forwarding events to other streams for further downstream processing. A PATTERN may appear anywhere in the from clause of an EPL statement. The notation of "->" indicates a PATTERN of event ordering.
Table 6.11 demonstrates an excerpt of three possibilities of rule definitions and their rule creations into the rule language. This table also demonstrates rule statements (A_r1, B_r2 and C_r1), which are interpreted from the graphical defined rules by the rule creation function for categories: A, B and C.
The Rule creation function interprets graphical rule into rule statements, which are expressed in Event Process Language (EPL). The rule statements are deployed in CEP engine, for the compliance checking in the next step.
After defining the three categories of our Ad–hoc BP and attaching their appropriate business conditions and rules, we obtain the cross-environmental variable values, the generic BP representation with the fixed part and the full Ad–hoc BP (see Figure 6.4.7).

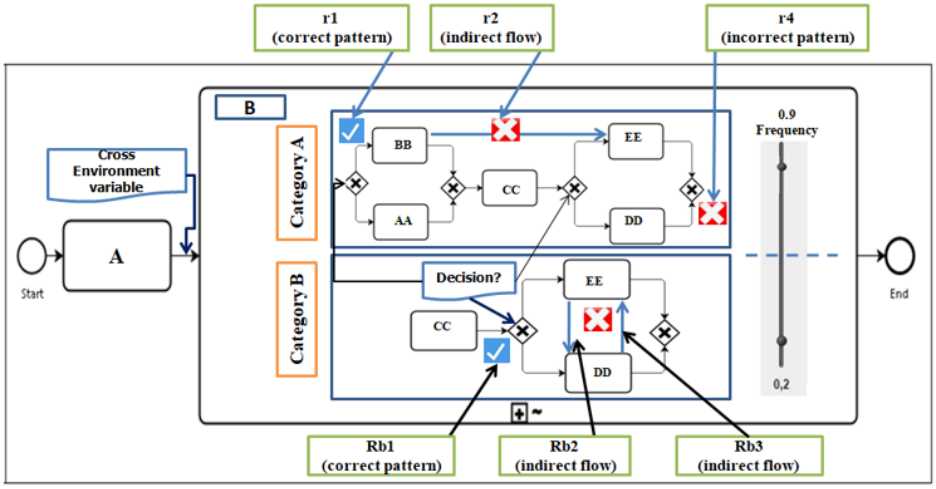


Figure 6.4.7: The generic BP with the fixed part (A) and the full Ad–hoc BP part (B: frequency degree, graphical rules, and decision points).

Table 6.12: Our case study cross-environmental variable values

Time	Resource	Antecedent	Category	Specific goal	Frequency
T1= From t1 to t3	Rs1	check	A	Possible End after the AC activity	More than 80%
T2= From t4 to t5	Rs2	check	B	Possible End after teh DLC activity	20%
T3= From t6 to t7	Rs3	check	C	End after the check activity	From 19% to 0%

d- Dynamic selection:

According to our proposed approach, the On–line view aims at selecting in dynamic manner the suitable Ad–hoc BP sub–process. This is done by applying the reasoning step using checking business conditions, business rules and by necessity the verification of the cross–environmental variable values. Generally, after the reasoning step we record all the relative information such as violations and solutions (into the Information system entity). This is rentable for future Ad–hoc BP improvement.

In our case, we use the CEP for checking conditions and business rules. We ensure the reasoning phase with the purpose of selecting the suitable Ad–hoc BP sub–process at runtime. In this example, we suppose that we arrive to the execution phase, with the specific goal (possible End after the DLC activity). Therefore, the Ad-hoc sub-process must be executed (see Figure 6.4.8).

After checking conditions, we can determine the adaptive Ad–hoc sub–process. For instance, we execute the category B with the resource Rs2, in t4 to t5 time interval and the A as antecedent. This presents which sub-process will be selected according to the cross-environmental variable values (see Table 6.12). Therefore, we select the Category B with 20% of frequency degree (see Figure 6.4.8).

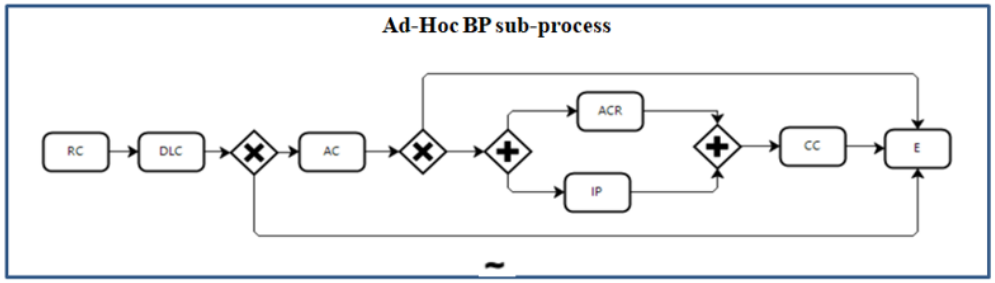


Figure 6.4.8: Adaptive Ad-hoc sub-process according to specific conditions.

6.4.3 Synthesis

In this section, we have illustrated how to treat dynamically Ad-hoc BP. This approach aims at adapting Ad-hoc BP sub-processes dynamically with changes according to the cross-environmental variable values: business conditions and business rules.

Throughout this approach, we use process discovery and conformance checking techniques. In the Off-line view, event logs are used in combination with the process discovery technique (frequency concept) for constructing all Ad–hoc BP behaviours (sub-processes). This helps in obtaining information about BP representation, business conditions and rules, i.e., which set of conditions will define a specific category? And with which business rules?

In the On–line view, the cross-environmental variable values are checked for reasoning in which way the Ad-hoc BP sub–process will be adaptive, to achieve the BP goal. This is done by comparing the obtained model (according to different frequency degrees) with event logs using the CEP checking mechanism.

In this section, we have applied our proposed approach on a concrete example. This example is

about a BP of claims handler. Therefore, our approach consists of two views: The Off-line view defines different behaviours of the Ad-hoc BP. This is done based on the combination between the process discovery technique and the frequency concept (illustrates different frequency models for constructing the full Ad–hoc BP). Then, we define business conditions (mapping categories according to specific business conditions) and rules (attached to each Ad-hoc BP part = category) relatively to the generic discovered model. All these conditions values are attributed to a cross-environmental variable. The On-line view illustrates how to execute dynamically the suitable Ad–hoc BP sub-process. This view is achieved using the conformance checking technique.

6.5 Summary

In this chapter, we presented three approaches dealing with unstructured BP challenges using process mining techniques.

The first approach takes into consideration the refined process mining framework. This later contains a set of activities that use extracted information from event logs, discovered models and normative ones. Among these activities, we find those dealing with running events in a SBP context, which are the Detect, the Predict and the Recommend activities. These three activities are nominated as operational support system that performs well on SBP while, it stills a challenging task for an UBP, because of its complex structure. In this regard, a special interest is given to the use of existing process mining techniques to analyse unstructured processes, from the extraction of a process model based on event data to the recommendation stage. To this end, we have proposed the orchestration of process mining activities into an UBP operational support approach, through the following phases: 1. Preparing Normative model, 2. Detect violations, 3. Preparing predictive model and Predictions and 4. Preparing the recommender model and Recommendations.

The second approach is developed to treat related challenges to self-defined BPs, in terms of process model's representation and variability management. Indeed, we study the applicability of process mining algorithms, to model the generic self-defined process model of user's behaviours in interaction with the e-administration domain (services provided by ministers, municipalities or local communities, and prefectures or states.). In these systems, users can have diverse ways to perform their research according to their objectives. In this context, users apply self–defined processes that may vary in terms of significance, structure, and results. At this stage, the resulted self-defined process model requires variation point elaboration, to define possible choices related to the execution process. To do so, the use of decision miner algorithm is required. This algorithm aims at detailing all sub-processes of the generic self-defined process model for defining self–defined BP ontologies, which are: user objective, user requirement and engine knowledge level. Hence, the configurable process model can be obtained. Last, the combination between the semantic reasoning through ontologies and the CPM can be released, to manage self–defined BP variability.

The third approach aims at treating dynamically Ad–hoc BPs using process mining techniques. Ad-hoc processes are not predefined, and the dynamic selection is not matched. Thus, the lack of adapting processes according to real-time variables is observed. To this end, we present requirements that must be respected in Ad-hoc BP definition. The Ad-hoc BP must be generic

and dynamic, i.e., adaptive to real-time variable conditions (changes). Besides, we illustrate how process mining techniques are used to define Ad-hoc BP content and how the CEP tool can be rentable in terms of verifying the cross–environmental variable values and executing dynamically the suitable Ad-hoc BP sub–process. In this context, our approach encompasses two views: The Off-line view aims at constructing generic model, using the process discovery technique in combination with the frequency concept. The On-line view uses the conformance checking technique, to adapt the suitable sub-process of the modelled Ad–hoc BP taking into consideration the dynamicity concept. After execution, all information will be recorded for future improvement of the Ad-hoc BP.

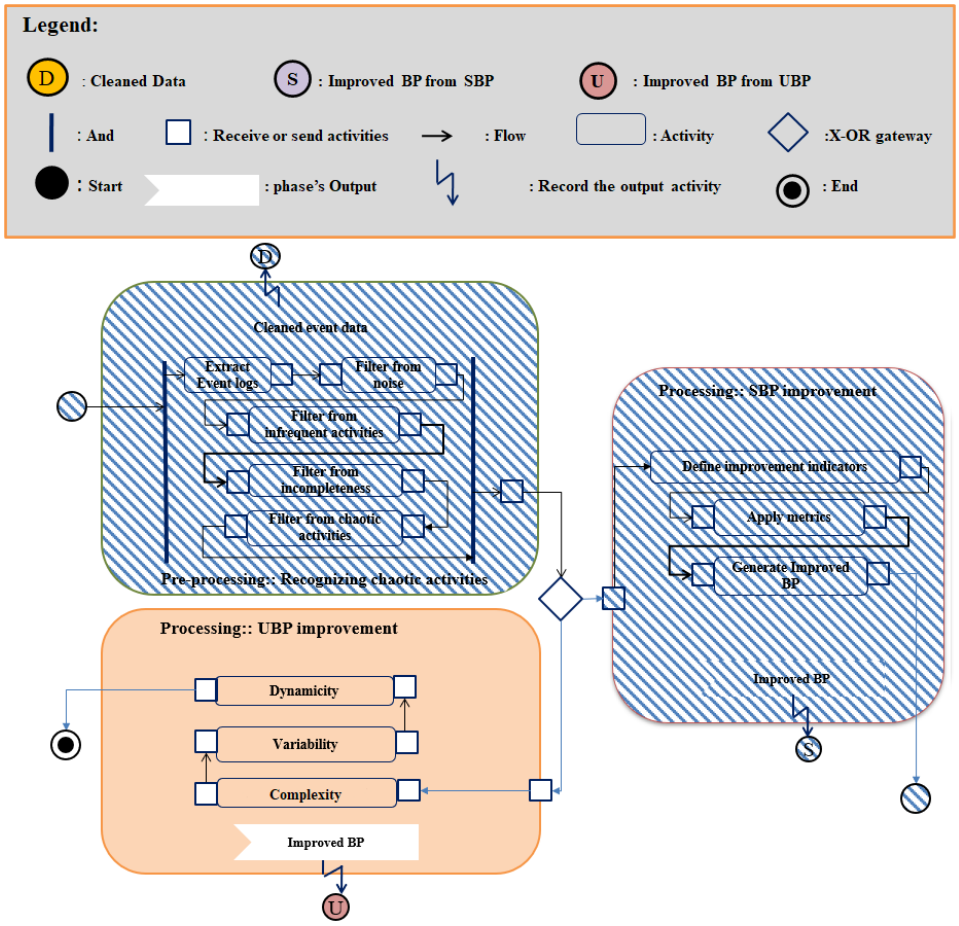


Figure 6.5.1: Processing: UBP Improvement

Chapter 7

Conclusion

In this thesis, we proposed three contributions that all deal with the improvement of business processes. In particular, we treat the following main challenges: 1) How to refine the quality of event data, 2) How to maintain BP structuration, 3) How to treat Unstructured BPs challenges?. All proposed approaches were systematically evaluated with synthetic data sets and applied in real-life event logs in the context of different illustrative examples according to the particularity of each approach.
This last chapter concludes this document and is structured as follows. In Section (7.1), we summarise our contributions and reflect on how they relate to this thesis research goals. In addition, several challenges in the field of process mining issues remain open. We acknowledge these possible improvements in Section (7.2). We suggest directions for future work in Section (7.3) and conclude this thesis by reflecting on the broader context in Section (7.4).

7.1 Summary of contributions

In Chapter 1, we introduced the general context of the thesis as well as the problem addressed, the main contributions and the thesis structure. As detailed in Section (1.5), the goals of our research were threefold: (G1) Recognizing chaotic activities from event logs in order to refine the event data quality; (G2) Maintaining the simplified BP structure related to the BPM life-cycle and process mining; (G3) Improving unstructured BPs by dealing with complexity, variability, and dynamicity. Chapter 2 is concerned with Background concepts and related definitions used throughout this thesis. Indeed, concepts and definitions covered the aspects of BPI and process mining. Also, chapter 2 provided preliminaries such as notations and formal expressions.
In Chapter 3, related works were analysed in the scope of process mining, especially in the preprocessing and the processing phases. This chapter briefly presented the necessary context to understand the contributions that were made in this thesis. Moreover, this chapter shown the process life-cycle and existing technological advances to recommend process mining as a mature BPI methodology. Therefore, we proposed three BPI approaches in combination using process mining techniques in chapters 4, 5 and 6.

7.1.1 Chaotic Activities Recognizing

In chapter 4, we proposed a recognizing approach that deals with chaotic activities. Here, our approach uses unsupervised machine learning algorithms to filter out event data from chaotic activities. The approach consists of four phases. We start by filtering out deficiencies from event data using PM4PY. In this order, we obtained cleaned events (without noise, infrequent and incomplete behaviours). Then, we proceeded to select the data frame on which we applied machine learning algorithms. We targeted cases and their corresponding sequence of activities. This consists of converting events from XES format to CSV format. Afterwards, we focused on extracting valuable features and discard irrelevant information from the original data set. This is applied in the objective of unifying data and encoding qualitative events to quantitative ones. Thereafter, we reduced the event data complexity. This allows acquiring simple data. Last, we applied the Isolation Forest algorithm on simplified data for detecting chaotic activities. Activities were filtered out from events to improve the BP quality. This work, entitled Chaotic activities recognizing during the pre-processing event data phase, was approved in [17]. This work develops two main algorithms that aim to filter out chaotic activities for simple and complex event data.

7.1.2 Improving Structured BP

In chapter 5, we proposed an approach that defines a set of improvement indicators according to the BPM life-cycle and the process mining techniques. Firstly, we define improvement indicators for each BPM life-cycle phase, then, we combine process mining indicators with the last BPM phase, i.e., we present the four process mining perspectives as improvement indicators (control flow, organisational, time and data). Also, an illustrative example is treated to demonstrate the applicability of the proposed approach and how it can lead to maintain BP improvement from the design phase to the optimise phase. This approach was approved in [19].

7.1.3 Improving Unstructured BP

Chapter 6 presented three approaches that deal with the improvement of UBPs based on process mining techniques. In this chapter, we treat different challenges appeared with the UBPs emergence. We develop a descriptive framework to reduce the complexity of UBPs. This is done to improve the visibility of events logs, by necessity its process models abstraction. We consider the complexity problematic [20]. Besides, Ad-hoc processes are treated with the dynamicity challenge [22] and the variability issue is illustrated with the self-defined BP [21].

7.1.4 Applications

To address our research goals, we treated different illustrative examples. In these examples, we used events logs that can be obtained from the repository [1], which is hosted at the 4TU (centre for research data).
Note that the focus of the conducted illustrative examples was on the evaluation of the proposed

[1] https://data.4tu.nl/search?q=collection+event+logs

approaches in a real-life environment. Thus, we only sketched the business context of each example briefly and restrict ourselves to applying the proposed approaches instead of conducting a full case study which would be focused one business question. Therefore, each of the illustrative example tested specific approach in real-life situations. To do so, the approaches correctness is verified.

7.2 Possible Improvements of our contributions

In this section, we acknowledge some improvements to our work. We summarize the most important ones related to our proposed approaches.

Improving structured BPs:

- It is necessary to define the business process type, from early stage, to target the indicators without going through all the improvement indicators and waste more time with energy on the processing phase.

Improving unstructured BPs:

- **Complexity:** An advanced logic to solve complexity related to the uncertainty issue, which until now cannot be solved by most discovery algorithms.
- **Dynamicity:** The prototype is capable with a large real-life log. However, there is a limitation with a log which has repeated executions of the same process instance. With this limitation, the prototype considers the repeated executed tasks as the same iteration as the previous execution. For instance, a rule is defined that A must not be followed by D either immediately or later. The two tasks are executed in different iterations but in the same process instance. With this limitation, the prototype concludes that a process instance executes A and later followed by D, so that means the rule is violated even though A and D are actually executed in different iterations. Therefore, the prototype should separately consider each iteration to avoid crossing to other iteration checking.
- **Security:** It is impossible to prove any non-trivial software to be error free. Thus, we cannot be sure that the data transformation during data preparation or the actual analysis is error free. It is required to take into consideration event logs confidentiality.

7.3 Future works

We summarize the most promising future works related to each of the approaches that we have identified while conducting this research.

Improving structured BPs:

- **Multi-perspectives improvement indicators according to the BP types:** for the whole BPM life cycle, it is interesting to define multi-perspectives improvement indicators in

combination with the process mining techniques (discovery, conformance checking and enhancement). Because of the particularity of each BP type (Operational, Support, Management), we may define more improvement indicators in the monitor and the optimize phases.

Improving unstructured BPs:

- **Advanced logic to solve complexity:** We must develop an advanced logic to solve complex concurrent logic on BPs under uncertainty, which until now cannot be solved by most discovery algorithms. To improve the discovery quality, we may analyse the heuristic impact based on frequency and time causality, and also the task weight causality used to adjust discovery view levels, considered with the four discovery level perspectives, which are fitness, simplification, precision and generalization. Also, to support implementation, it is important to develop several feature and tools that cover various standard BP models, such as BPMN, for conformance evaluation and BP auditing as ProM plug-ins library.
- **Sustainability:** The most used method to meet a sustainability goal, in term of reducing energy consumption, was to re-think and re-design the general usage of resources, such as electricity usage for lightning and cooling of the facility and the energy for treating data. Based on this percept, to ensure sustainable process, we have to: reduce processing errors, Monitor processes for quick corrective reaction. reduce computing resources when extracting the log information. Indeed, we have to evaluate our proposed approaches in the sustainability context by introducing statistical values. For instance, 1-reducing the complexity related to the treated data, by necessity, this is influencing the energy consumed during the process mining treatment. 2-optimize the resulted SBP to a refined normative model; this may help in monitoring business processes, at runtime, of the same context for instantiate corrective reaction. These two stages can be rentable in term of conserving energy. A research is required to treat the combination between the sustainability objectives and process mining implementation.

7.4 Reflection on the broader context: Towards Big Data

We conclude this thesis by reflecting on our contributions in a broader context. Due to the growing computing power and storage capacity of today's IT systems, organizations have the opportunity to store information about all their activities. Therefore, leveraging knowledge from such recorded data is widely acknowledged to be an important challenge. For example, consider deep learning methods that have been very successful in learning specific complex tasks based on large amounts of data. Process mining is part of this trend towards organizations that are driven by data. Process mining methods operate on event logs that contain traces recorded from the execution of a process. These methods, such as our contributions, can already be useful when applied to smaller amounts of data that is provided in the form of event logs. Often, huge amounts of data are not available in the context of a specific business process. However, process mining requires the input data to be structured in (or convertible to) the form of an event log. Towards big data ecosystem, the most

important approach deals on how to distribute our improvement approaches in combination with Map-reduce (techniques used to treat distributed data on Hadoop clusters) algorithms.
From the technical standpoint, our proposed approaches can be applied on smaller data distributed on each Data node (number of the cluster). To do so, smaller part of treatment, orchestrated by a Name node (Master node) can be integrated. The challenge, here, is how to adapt the map and reduce classes to our proposed approaches. For instance, how to use different levels of knowledge gathered from event logs during the mapping operation. As a basic answer, we can develop different plug-ins implementing Map-Reduce based on process mining techniques. No matter what programming (python, C++) is used, there are different APIs that can take care to adapt the source code with Hadoop environment.
However, as with any approach that leverages recorded data, the results rely on the quality of the recorded data. Any data-based methods can only be denoted as being "evidence based" if the recorded data actually represents factual information. For example, events are not always representing what happened in reality. Thus, any conclusion drawn upon the results of our methods on unreliable data should be treated with care. Indeed, it does not always need to be the source data that was wrongly recorded. Error may be inadvertently introduced due to mistakes during data preparation or due to errors in the employed software. Traceability on how the underlying raw data is processed and transparency on how these data are used by the analysis method is required. Moreover, ethical considerations should also be considered when dealing with data that could, possibly, affect the employees and customers of an organization.
To conclude, the responsible usage of data in the context of process mining is an important topic. The result of a data-based analysis should always be seen in the context of the real process execution, which might not be 100% reflected in the data that is available.

Bibliography

[1] W.M.P van der Aalst. *Process mining: Data science in action*. Springer-Verlag Berlin Heidelberg, 2016.

[2] Z. Lamghari, M. Radgui, R. Saidi, and M.D. Rahmani. *Passage challenges from data–intensive system to knowledge–intensive system related to process mining field*. Proceedings of the ArabWIC 6th Annual International Conference Research Track–ACM, 2019.

[3] W.M.P. Van der Aalst. Using process mining to bridge the gap between BI and BPM. *IEEE Computer*, 44(12):77–80, 2011.

[4] M. Hammer and J. Champy. *Reengineering the Corporation: A Manifesto for Business Revolution*. Paperback edition, HarperBusiness, New York, 1993.

[5] T.H. Davenport. *process innovation: reengineering work through information technology*. Harvard Business Press, 1993.

[6] V. Elapatha, J. Wijeratne, and N. Shahzadah. An analysis of the implementation of business process re-engineering in public services. *Journal of Open Innovation: Technology, Market, and Complexity*, 6(4):114, 2020.

[7] R. Saxby, M. Cano-Kourouklis, and E. Viza. An initial assessment of lean management methods for industry 4.0. *The TQM Journal*, 32(4):587–601, 2020.

[8] A. Thomas, R. Barton, and C. Chuke-Okafor. Applying lean six sigma in a small engineering company–a model for change. *Journal of Manufacturing Technology Management*, 20(1):113–129, 2009.

[9] N.H Baluch, A. Che Sobry, and S. Mohtar. TPM and lean maintenance: A critical review. *Interdisciplinary Journal of Contemporary Research in Business (IJCRB)*, 4(2):850–857, 2012.

[10] M. J. Harry. Six sigma: A breakthrough strategy for profitability. *Quality Progress*, 31(5):60–64, 1998.

[11] S. J. de Jong, W. Wouter, and B. Van Blokland. Measuring lean implementation for maintenance service companies. *International Journal of Lean Six Sigma*, 7(1):35–61, 2016.

[12] G. K. Kanji. Total quality management: the second industrial revolution. *Total Quality Management*, 1(1):3–12, 1990.

[13] P. Ayeni, P. Ball, and T. Baines. Towards the strategic adoption of lean in aviation Maintenance Repair and Overhaul (MRO) industry: An empirical study into the industry's lean status. *Journal of Manufacturing Technology Management*, 27(1):38–61, 2016.

[14] J. Méndez, M. David, and R. Rodriguez. Total productive maintenance (TPM) as a tool for improving productivity: a case study of application in the bottleneck of an autoparts machining line. *The International Journal of Advanced Manufacturing Technology*, 92(1):1013–1026, 2017.

[15] L. Webber and M. Wallace. *Quality control for dummies*. John Wiley & Sons, 2011.

[16] M. Dumas, W.M.P. Van der Aalst, and A.H Ter Hofstede. *Process–Aware Information Systems: Bridging people and software through process technology*. John Wiley & Sons, 2005.

[17] Z. Lamghari, R. Saidi, M. Radgui, and M.D. Rahmani. Chaotic activities recognizing during the pre-processing event data phase. *International Journal of Business Intelligence and Data Mining*, 18(2), 2021.

[18] Z. Lamghari, M. Radgui, R. Saidi, and M.D. Rahmani. *A set of indicators for BPM life cycle improvement*. International Conference on Intelligent Systems and Computer Vision–IEEE, 2018.

[19] Z. Lamghari, M. Radgui, R. Saidi, and M.D. Rahmani. Defining Business Process Improvement metrics based on BPM life cycle and process mining techniques. *International Journal of Business Process Integration and Management*, 9(2):107–133, 2019.

[20] Z. Lamghari, R. Saidi, M. Radgui, and M.D. Rahmani. An operational support approach for mining unstructured business processes. *Revista de Informática Teórica e Aplicada*, 28(1):23–38, 2021.

[21] Z. Lamghari, R. Saidi, M. Radgui, and M.D. Rahmani. Mining self–defined business process in electronic administration [in–press]. *International Journal of E-Services and Mobile Applications*, 14(1), 2022.

[22] Z. Lamghari, R. Saidi, M. Radgui, and M.D. Rahmani. *Leveraging dynamicity and process mining in Ad-hoc business process –Innovations in Smart Cities Applications*. The Proceedings of the 5th International Conference on Smart City Applications–Lecture Notes in Network and Systems–v 183–Springer, 2020.

[23] W.M.P. van der Aalst, A. Adriansyah, and B. F. van Dongen. *Replaying History on Process Models for Conformance Checking and Performance Analysis*. WIREs Data Min Knowl Discovery, 2012.

[24] B. Curtis, M.I. Kellner, and J. Over. *Process Modeling*. Acm, 1992.

[25] ISO 9000. *International Standards for Quality Management*. Geneve, Switzerland, 2015.

[26] C.W. Gunther and W.M.P van der Aalst. *Fuzzy mining adaptive process simplification based on multi perspective metrics in International conference on business process management*. Springer, Berlin, Heidelberg, 2007.

[27] M.P. Bonacina aand G. Stéphane and N. Shankar. Conflict–driven satisfiability for theory combination: transition system and completeness. *Journal of Automated Reasoning*, 64(3):579–609, 2020.

[28] Carl Adam Petri. *Kommumkation mit Automaten*. Phd thesis, Institut fur Instrumentelle Mathematik, Universitat Bonn, 1962.

[29] A.A. Kalenkova, W.M.P. van der Aalst, I.A. Lomazova, and V.A. Rubin. Process Mining using BPMN: Relating event logs and process models. *Software & Systems Modeling*, 16(4):1019–1048, 2017.

[30] C. Di Ciccio, A. Marrella, and A. Russo. Knowledge–intensive processes: Characteristicsrequirements and analysis of contemporary approaches. *Journal on Data Semantics*, 4(1):29–57, 1991.

[31] M. Werner and N. Gehrke. Process mining. *die Zeitschrift fur den Wirtschaftsstudenten*, 7(13):1–16, 2013.

[32] Z.A. Bukhsh, M. van Sinderen, K. Sikkel, and D.A. Quartel. *Understanding Modeling Requirements of Unstructured Business Processes*. In 14th International Conference on e–Business, King Juan Carlos University, Madrid, Spain, 2017.

[33] J. Taylor. *Decision management solutions*. Retrieved 2014 from http://www.decisionmanagementsolutions.com/, 2014.

[34] M. Pegoraro and W.M.P van der Aalst. *Mining uncertain event data in process mining in International Conference on Process Mining*. IEEE, 2019.

[35] R. Sang. The finite element method for elliptic problems. *Journal of Elasticity*, 38(2):209–218, 1991.

[36] S.K. Vanden Broucke and J. De Weerdt. Fodina: A robust and flexible heuristic process discovery technique. *Decision support systems*, 100:109–118, 2017.

[37] A. S. Bogarin Vega, Cerezo R. Menendez, and C. Romero. Discovering learning processes using inductive miner: A case study with learning management systems. *Psicothema*, 30(3):322–329, 2018.

[38] J. M. E. Van der Werf, B.F. van Dongen, C. A. Hurkens, and A. Serebrenik. *Thriving on Adaptability: Best Practices for Knowledge Workers*. Springer, 2008.

[39] W.M.P Van der Aalst, V. Rubin, H.M.W Verbeek, B.F. van Dongen, E. Kindler, and C.W. Gunther. Process mining: a two-step approach to balance between underfitting and overfitting. *Software & Systems Modeling*, 9(1):87, 2010.

[40] M. Kebede. *Comparative Evaluation of Process Mining Tools*. P.hd thesis, University of Tartu, Estonia, 2015.

[41] P. Gonella. *Process mining: A database of applications*. HSPI Management, 2017.

[42] M. Bozkaya, J. Gabriels, and J.M. Werf. *Process Diagnostics: a Method Based on Process Mining*. In the International Conference on Information, Process, and Knowledge Management–IEEE, 2009.

[43] C. Di Francescomarino, C. Ghidini, F. Maggi, and M. Fredrik M. Milani. *Predictive process monitoring methods: Which one suits me best?* The International Conference on Business Process Management–Springer, 2018.

[44] N. Palmer, L. Fischer, and S. Reddy. *Thriving on Adaptability: Best Practices for Knowledge Workers*. Future Strategies Incorporated, 2015.

[45] L.T. Ly, C. Indiono, J. Mangler, and S. inderle Ma. *Data transformation and semantic log purging for process mining*. in International Conference on Advanced Information Systems Engineering–Springer, 2012.

[46] V. Priyadharshini and A. Malathi and. An efficient process mining model for petri nets in process discovery. *IOSR Journal of Computer Engineering*, 164(4):65–68, 2014.

[47] H.J. Cheng and A. Kumar. Process mining on noisy logs–can log sanitization help to improve performance. *Decision Support Systems*, 79(1):138–149, 2015.

[48] J. Wang, S. Song, X. Lin, and J. Pei. *Cleaning structured event logs: A graph repair approach*. in the 31st International Conference on Data Engineering–IEEE, 2015.

[49] X. Lu, S.A. Tabatabaei, M. Hoogendoorn, and H.A. Reijers. *Trace Clustering on Very Large Event Data in Healthcare Using Frequent Sequence Patterns*. in International Conference on Business Process Management–Springer, 2019.

[50] S. Suriadi, R. Andrews, A.H. ter Hofstede, and M.T. Wynn. Event log imperfection patterns for process mining: Towards a systematic approach to cleaning event logs. *Information Systems*, 64(1):132–150, 2017.

[51] Z. Wynn and M. Sadiq. *Responsible process mining–A data quality perspective*. in the International Conference on Business Process Management–Springer, 2019.

[52] C.J. Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998.

[53] P.N. Tan, M. Steinbach, and V. Kumar. *Introduction to data mining, Pearson education*. Pearson Addison-Wesley, 2006.

[54] A. Liu, C. Martin, T. Hetherington, and S. Matzner. *A comparison of system call feature representations for insider threat detection*. In Proceedings from the Sixth Annual IEEE SMC Information Assurance Workshop–IEEE, 2005.

[55] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo. *A geometric framework for unsupervised anomaly detection*. Applications of data mining in computer security–Springer, 2002.

[56] D.J. Cook, L.B. Holder, T. Hetherington, and S. Matzner. *Mining graph data*. John Wiley & Sons, 2006.

[57] A. Berti, S.J. van Zelst, and W. van der Aalst. *Process mining for python (PM4PY): Bridging the gap between process–and data science*. arXiv preprint, 2019.

[58] M.F. Sani, S.J. Van Zelst, and W.M.P. Van der Aalst. *Applying sequence mining for outlier detection in process mining*. The Confederated International Conferences On the Move to Meaningful Internet Systems–Springer, 2018.

[59] J.C. Buijs, B.F. Van Dongen, and W.M.P. Van der Aalst. *On the role of fitness, precision, generalization and simplicity in process discovery*. The Confederated International Conferences on the Move to Meaningful Internet Systems–Springer, 2012.

[60] A. Augusto, R. Conforti, M. Dumas, M. La Rosa, and A. Polyvyany. Split miner: automated discovery of accurate and simple business process models from event logs. *Knowledge and Information Systems*, 59(2):251–284, 2019.

[61] S.J. Leemans, D. Fahland, and W.M.P. Van der Aalst. *Discovering block-structured process models from event logs containing infrequent behaviour*. in International conference on business process management–Springer, 2013.

[62] W.M.P. Van der Aalst. *Improving Process Discovery Results by Filtering Outliers Using Conditional Behavioural Probabilities*. in Business Process Management Workshops: BPM 2017 International Workshops–Springer, 2018.

[63] X. Lu, D. Fahland, F.J. Van den Biggelaar, and W.M.P. Van der Aalst. *Detecting deviating behaviors without models*. in in International Conference on Business Process Management–Springer, 2019.

[64] M.F. Sani, M. Boltenhagen, and W.M.P. Van der Aalst. *Prototype Selection Based on Clustering and Conformance Metrics for Model Discovery*. arXiv preprint, 2019.

[65] M.F. Sani, S.J. Van Zelst, and W.M.P. Van der Aalst. Repairing outlier behaviour in event logs using contextual behaviour. *Enterprise Modelling and Information Systems Architectures*, 14(1):1–5, 2019.

[66] W. Eberle and L. Holder. *Discovering structural anomalies in graph-based data Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007)*. IEEE, 2007.

[67] M. Dumas, J. Mendling, M. La Rosa, and A.H. Reijers. *Introduction to Business Process Management*. Fundamentals of Business Process Management–Springer, 2019.

[68] A. Arsanjani, N. Bharade, M. Borgenstrand, P. Schume, J.K. Wooed, and V. Zheltonogov. *Introduction to successful business process management*. Business Process Management Design Guide: using IBM process manager–NY:IBM Redbooks, 2015.

[69] S.R. Koster. *An evaluation method for Business Process Management products*. Master Thesis–The University of Twente– Enschede–the Netherlands, 2009.

[70] S. Simmons. *BPM Voices: Evaluating BPM applications: BPM design reviews and Rubik's Cubes*. IBM–developer works, 2013.

[71] B. Wetzstein, Z. Ma, A. Filipowska, M.Kaczmarek, S. Bhiri, S. Losada, J.M. Lopez-Cobo, and L. Cicurel. *Semantic Business Process and Product Lifecycle Management*. Proceedings of the Workshop SBPM–Springer, 2007.

[72] M. Weiss and D. Amyot. *Business Process Monitoring and Alignment: An Approach Based on the User Requirements Notation and Business Intelligence Tools*. DBLP, 2007.

[73] C. Bai and J. Sarkis. A grey-based dematel model for evaluating business process management critical success factors. *International Journal of Production Economics*, (1):281–292, 2013.

[74] A.H. Mousa, N. Shiratuddin, and M.S. Abu Bakar. Evaluation framework for business process evaluation approaches. *Journal of Computer Science & Computational Mathematics*, (13):281–292, 2016.

[75] W.M.P. Van der Aalst, N. Mariska, and A.R. Hajo. Supporting the full bpm life-cycle using process mining and intelligent redesign. *Contemporary issues in database design and information systems development*, (13):100–132, 2011.

[76] N. Balaban, K. Belic, and M. Gudelj. Business process performance management: theoretical and methodological approach and implementation. *Management Information Systems*, (4):3–9, 2011.

[77] R. Bhagwat and M.K. Sharma. Performance measurement of supply chain management: a balanced scorecard approach. *Computer Industrial and Engineering*, (1):43–62, 2007.

[78] A. Del-Rio-Ortega, M. Resinas, C. Cabanillas, and A. Ruiz-Cortes. On the definition and design–time analysis of process performance indicators. *Information Systems*, (4):470–490, 2013.

[79] A. Gunasekaran and B. Kobu. Performance measures and metrics in logistics and supply chain management: a review of recent literature for research and applications. *International Journal of Production Research*, (12):37–41, 2007.

[80] P. Kueng and A.J.W. Krahn. Process performance measurement system: some early experiences. *Journal of Scientific Industrial Research*, (3):149–159, 1999.

[81] K.Y. Kutucuoglu, J. Hamali, J.M. Sharp, and Z. Irani. Enabling BPR in maintenance through a performance measurement system framework. *International Journal of Flexible Manufacturing Systems*, (1):33–52, 2002.

[82] D.B. Mirsu. *Monitoring help desk process using KPI*. Soft Computing Applications. Advances in Intelligent Systems and Computing–Springer, 2013.

[83] M. Spremic, Z. Zmirak, and K. Kraljevic. *IT and business process performance management: case study of ITIL implementation in finance service industry*. The 30th International Conference on Information Technology Interfaces, 2008.

[84] P. Walsh. Finding key performance drivers: some new tools. *Total Quality Management*, (5):509–519, 2010.

[85] W.M.P. van der Aalst, A. Adriansyah, A.K. De Medeiros, F. Arcieri, T. Baier, T. Blickle, and B. Chandra. *Pocess Mining Manifesto*. In the international Conference on Business Process Management–Springer, 2011.

[86] A. Kalenkova, I.A. Lomazova, and W.M.P. Van der Aalst. *Process Model Discovery: A Method Based on Transition System Decomposition*. In International Conference on Applications and Theory of Petri Nets and Concurrency, 2014.

[87] W.M.P. Van der Aalst. *A General Divide and Conquer Approach for Process Mining*. In Feder-ated Conference on Computer Science and Information Systems, 2013.

[88] J. Munoz-Gama, J. Carmona, and W.M.P. Van der Aalst. Single-entry single-exit decomposed conformance checking. *Information Systems*, 46(1):102–122, 2014.

[89] W.M.P. Van der Aalst. Process Cubes: Slicing, Dicing, Rolling up and Drilling down event data for process mining. *Lecture Notes in Business Information Processing*, 3159(1):1–22, 2013.

[90] H. Irshad, B. Shafiq, J. Vaidya, M.A. Bashir, S. Shamail, and N. Adam. Preserving privacy in collabo-rative business process composition. *The 12th International Joint Conference on e–Business and Telecommunications*, 4(1):112–123, 2015.

[91] H.M.W. Verbeek and W.M.P. Van der Aalst. *Merging Alignments for Decomposed Replay*. Lecture Notes in Computer Science–Springer, 2016.

[92] J. Munoz-Gama, J. Carmona, and W.M.P. Van der Aalst. *Partitioning and Topology. International Conference on Business Process Management*. International Conference on Business Process Management–Lecture Notes in Computer Science, 2013.

[93] A. Polyvyanyy, L. García-Bañuelos, D. Fahland, and M. Weske. Maximal structuring of acyclic process models. *The Computer Journal*, 57(1):12–35, 2014.

[94] M. Polato, A. Sperduti, A. Burattin, and M. de Leoni. *Data-aware remaining time prediction of business process instances*. The International Joint Conference on Neural Networks–IEEE, 2014.

[95] G. Oulsnam. The algorithmic transformation of schemas to structured form. 1987.

[96] Artem A. Polyvyanyy, L. García-Bañuelos, and M. Dumas. *Structuring acyclic process models*. International Conference on Business Process Management–Springer, 2010.

[97] A. Augusto, R. Conforti, M. Dumas, M. La Rosa, and G. Bruno. Automated discovery of structured process models from event logs: the discover-and-structure approach. *Data & Knowledge Engineering*, 117(1):373–392, 2018.

[98] J. Nakatumba, M. Westergaard, and W.M.P. Van der Aalst. *Generating event logs with workload–dependent speeds from simulation models*. Springer, 2012.

[99] F. Folino, Francesco, G. Folino, and L. Pontieri. *An ensemble–based P2P framework for the detection of deviant business process instances*. The International Conference on High Performance Computing & Simulation–IEEE, 2018.

[100] R. Conforti, M. De Leoni, M. La Rosa, and W.M.P. Van Der Aalst. *Supporting risk–informed decisions during business process execution*. The International Conference on Advanced Information Systems Engineering–Springer, 2013.

[101] M. De Leoni and W.M.P. Van Der Aalst. *The Feature Prediction Package in ProM: Correlating Business Process Characteristics*. BPM (Demos), 2014.

[102] B.F.A. Hompes, J.C.A.M. Buijs, W.M.P. Van der Aalst, P.M. Dixit, and J. Buurman. Proceedings of the 27th Benelux Conference on Artificial Intelligence, 2015.

[103] M. De Leoni, W.M.P. Van Der Aalst, and M. Dees. A general process mining framework for correlating, predicting and clustering dynamic behavior based on event logs. *Information Systems*, 56(1):235–257, 2016.

[104] N. Mehdiyev, J. Evermann, and P. Fettke, editors. *A multistage deep learning approach for business process event prediction*. The 19th Conference on Business Informatics–IEEE, 2017.

[105] L. Lin, L. Wen, and J. Wang, editors. *Mm-pred: A deep predictive model for multi-attribute event sequence*. 2019.

[106] W.M.P. Van Der Aalst, M. La Rosa, and A.H. ter Hofstede. Questionnaire–based variability modeling for system configuration. *Software & Systems Modeling*, 8(2):51–274, 2009.

[107] L. El Faquih, H. Sbai, and M. Fredj. *Semantic variability modeling in business processes: A comparative study*. The 9th International Conference for Internet Technology and Secured Transactions–IEEE, 2014.

[108] K. Athukorala, D. Glowacka, G. Jacucci, A. Oulasvirta, and J. Vreeken. Is exploratory search different acomparison of information search behavior for exploratory and lookup tasks. *Software & Systems Modeling*, 67(11):2635–2651, 2016.

[109] S.P. Detro, E. Portela, E.L. Rocha, H. Panetto, and M. Lezoche. Configuring process variants through semantic reasoning in systems engineering. *International Council on Systems Engineering*, 20(4):36–39, 2017.

[110] M. La rosa, W.M.P. Van der Aalst, M. Dumas, and M.P. Fredrik. Business process variability modeling: A survey. *ACM Computing Surveys–CSUR*, 50(1):1–45, 2017.

[111] H. Schonenberg, R. Mans, N. Russell, N. Mulyar, and W.M.P. Van der Aalst. *Process flexibility: A survey of contemporary approaches*. In Advances in enterprise engineering -Springer, 2008.

[112] S. Dustdar, T. Hoffmann, and W.M.P. Van der Aalst. Mining of Ad–hoc business processes with teamlog. *Data & Knowledge Engineering*, 55(2):129–158, 2005.

[113] D. Duma, R. Aringhieri, and W.M.P. Van der Aalst. An Ad–hoc process mining approach to discover patient paths of an emergency department. *Flexible Services and Manufacturing Journal*, 55(2):1–29, 2018.

[114] D. Duma, R. Aringhieri, and W.M.P. Van der Aalst. Mining of ad–hoc business processes with teamlog. *Flexible Services and Manufacturing Journal*, 55(2):129–158, 2020.

[115] M. Kiedrowicz. *Dynamic business process in workflow systems*. In MATEC Web of Conferences–EDP Sciences, 2017.

[116] P. Jain, P.Z. Yeh, K. Verma, A. Kass, and A. Sheth. *Enhancing process–adaptation capabilities with web–based corporate radar technologies*. In Proceedings of the first in-ternational workshop on Ontology-supported business intelligence, 2008.

[117] S. Vasilecas, T. Kalibatiene, D., and Lavbic. Implementing rule-and context-based dynamic business process modelling and simulation. *Journal of Systems and Software*, 122(1):1–15, 2016.

[118] X. Zhu, J. Recker, G. Zhu, and F.M. Santoro. Exploring location-dependency in process modeling. *Business Process Management Journal*, 20(6):794–815, 2014.

[119] M. Adams, R. Mans, N. Russell, N. Mulyar, and W.M.P. Van der Aalst. *Dynamic workflow. In Modern Business Process Automation*. Springer, 2010.

[120] ECM. Object Management Group. *Case management model and notation (CMMN) Formal Version 1.0.2014*. Object Management Group, 2016.

[121] N.M. El-Gharib and D. Amyot. *Process Mining for Cloud-Based Applications: A Systematic Literature Review*. The 27th International Requirements Engineering Conference Workshops–IEEE, 2019.

[122] J. Lever, M. Krzywinski, and N. Altman. Nature methods. *Journal of Elasticity*, 14(2):641–642, 2017.

[123] W. Es-Soufi, E. Yahia, and L. Roucoules. *On the use of Process Mining and Machine Learning to support decision making in systems design*. IFIP International Conference on Product Lifecycle Management–Springer, 2016.

[124] A. Leontjeva, R. Conforti, C. Di Francescomarino, M. Dumas, and F.M. Maggi. *Complex symbolic sequence encodings for predictive monitoring of business processes*. Business Process Management–Springer, 2015.

[125] R. Davis. *What makes a good process?* BPTrends, 2009.

[126] A. Kandukuri. *Predictive Process Monitoring related to the remaining time dimension: a value–driven framework*. Performance Metrics, 2014.

[127] E.E. Adam and P.M. Swamidass. Assessing operations management from a strategic perspective. *Journal of Management*, 15(2):181–203, 1989.

[128] D.A. Garvin. *Managing Quality*. Free Press, 1987.

[129] Z. Parasuraman and Berry. *Delivering Service Quality*. Free Press, 1990.

[130] W.M.P. van der Aalst. Process mining: Overview and opportunities. *ACM Transactions on Management Information Systems*, pages 1–17, 2012.

[131] W.M.P. van der Aalst. Process mining: X–ray your business processes. *CACM*, 99(99):1–11, 2011.

[132] E. Saad. The social process model: strategize at the personal level. *Business and Economics Journal*, 4(1):1–2, 2015.

[133] A. Augusto, R. Conforti, M. Dumas, M. La Rosa, F.M. Maggi, and A. Marrella. Automated discovery of process models from event logs: Review and benchmark. *IEEE Transactions on Knowledge and Data Engineering*, 31(4):686–705, 2018.

[134] W.C. Carmo A.B. and Albuquerque. *O Gerenciamento de projetos apoiado por uma ferramenta de Business Process Management*. Proceeding of the 9th Iberian Conference on Information Systems and Technologies–IEEE, 2014.

[135] A. Lütje, A. Möller, and V. Wohlgemuth. *A preliminary concept for an IT-supported industrial symbiosis (IS) tool using extended material flow cost accounting–Impulses for environmental management information systems*. Advances and New Trends in Environmental Informatics–Springer, 2018.

[136] F. Mannhardt, M. De Leoni, H.A. Reijers, and W.M.P. Van Der Aalst. Balanced multi-perspective checking of process conformance. *Computing*, 98(4):407–437, 2016.

[137] W.M.P. Van Der Aalst, A. Bolt, and J.S. van Zelst. Rapidprom: mine your processes and not just your data. *arXiv preprint*, 1703:1–25, 2017.

[138] Z. Lamghari, M. Radgui, R. Saidi, and M.D. Rahmani. *Predictive Process Monitoring related to the remaining time dimension: a value–driven framework*. 1st International Conference on Smart Systems and Data Science–IEEE, 2019.

[139] F. Mannhardt. *Multi–perspective Process Mining*. Phd thesis, Technische Universiteit Eindhoven, 2018.

[140] F. Gottschalk, W.M.P. Van der Aalst, and M.H. Jansen-Vullers. *Configurable process models-a foundational approach*. Reference Modeling–Physica–Verlag HD–Springer, 2007.

[141] C. Ayora, V. Torres, M. Reichert, B. Weber, and V. Pelechano. *Towards runtime flexibility for process families: open issues and research challenges*. In International Conference on Business Process Management–Springer, 2012.

[142] Y. Liao, M. Lezoche, H. Panetto, N. Boudjlida, and E.R. Loures. Semantic annotation for knowledge explicitation in a product lifecycle management context a survey. *Computers in Industry*, 71(1):24–34, 2015.

[143] M. Hepp and D. Roman. *An ontology framework for semantic business process management*. In International Conference on Business Information Systems-Springer, 2007.

[144] I. Szabo and K. Varga. *Knowledge-based compliance checking of business processes. In On the Move to Meaningful Internet Systems Confederated International Conferences*. Springer, 2014.

[145] H. Fei and N. Meskens. *Discovering patient care process models from event logs*. In The 8th International Conference on Modeling Simulation, 2010.

[146] M. Rafiei, L. Von Waldthausen, and W.M.P. Van der Aalst. *Ensuring Confidentiality in Process Mining*. In the 7th international symposium on data-driven process discovery and analysis–Springer, 2018.

[147] R.S. Mans, M.H. Schonenberg, M. Song, W.M.P Van der Aalst, and P.J. Bakker. *Application of process mining in healthcare a case study in a dutch hospital*. in International joint conference on biomedical engineering systems and technologies–Springer, 2008.

[148] M.J. Cole, C. Hendahewa, N.J. Belkin, and C. Shah. User activity patterns during information search. *ACM Transactions on Information Systems*, 33(1):1–39, 2015.

[149] C. Hai and I. Jeong. *Fundamental of development administration*. Scholar, 2007.

[150] Z. Lamghari, M. Radgui, R. Saidi, and M.D Rahmani. *A Framework Supporting Supply Chain Complexity and Confidentiality Using Process Mining and Auto Identification Technology in International Conference Europe Middle East & North Africa Information Systems and Technologies to Support Learning*. Springer, 2019.

[151] Y. Huang, Z. Feng, K. He, and Y. Huang. *Ontology-based configuration for service-based business process model*. In the IEEE International Conference on Services Computing– IEEE, 2013.

[152] G. Decker, H. Overdick, and M. Weske. *Oryx an open modeling platform for the BPM community*. Springer, 2008.

[153] D.S. Horgan, J.R. Holliday, and E. O'toole. *Buildig access control system with complex event processing*. Johnson Controls Technology Co, 2020.

[154] D. Geneiatakis, I. Kounelis, R. Neisse, I. NaiFovino, G. Steri, and G. Baldini. *Security and privacy issues for an IoT based smart home*. In the 40th International Convention on Information and Communication Technology Electronics and Microelec tronics– IEEE, 2017.

Résumé

De nos jours, les entreprises utilisent diverses méthodologies pour analyser leurs processus métiers, ce qui leur permet d'apporter des améliorations en terme d'efficacité et d'efficience, ainsi que l'optimisation de ces derniers dans le contexte de la gestion des processus métiers (BPM).
Parmi les méthodologies d'amélioration des processus métiers qui ont été élaborées et déployées, on trouve la fouille des processus métiers (Process Mining). Cette méthodologie se situe entre l'intelligence computationnelle et la fouille de données d'une part, et la modélisation des processus métiers d'autre part. Les techniques de cette méthodologie visent à extraire les connaissances des journaux d'évènements, enregistrées lors de l'exécution des processus métiers, afin de les découvrir, les contrôler et les améliorer.
Dans cette thèse, nous abordons un ensemble de défis pour lesquels une vision d'amélioration des processus métiers est nécessaire, prenant en considération la qualité des données d'exécution et les différentes structures des processus métiers résultants.

Mots-clefs : *Process métier ; amélioration des processus métiers ; fouille des processus ; indicateurs d'amélioration ; complexité ; technique de la découverte.*

Abstract

Nowadays, many business process improvement methodologies have been developed and implemented in organisations. These methodologies allow overcoming business improvement challenges in a continuous Business Process Management (BPM) context.
Among these business process improvements methodologies, we focus on the Process Mining methodology. Process mining is a scientific discipline that combines machine learning algorithms and business process design methods, based on recorded information systems data in the form of event logs. This is done to discover (process discovery), analyse (conformance checking), and improve (enhancement) processes.
In this work, we aim to improve business processes using process mining techniques. Indeed, we use process execution data, to recognize and understand ground realities of processes that are being followed inside organizations, thus reaching their improvement or redesign. To do so, we address research challenges in which improvement operations of business processes are required. We also look beyond the impact of event logs quality on business process structures and its improvement process.

Key Words : *Business process ; business process improvement ; process mining ; improvement indicators ; complexity; discovery technique.*