DETECTING NOVEL BEHAVIOR AND PROCESS IMPROVEMENT WITH MULTI-MODAL PROCESS MINING

ÇOK MODLU SÜREÇ MADENCİLİĞİ İLE YENİ DAVRANIŞIN TESPİTİ VE SÜREÇ İYİLEŞTİRME

ABDURRAHMAN TELLİ

DOÇ. DR. AYÇA KOLUKISA TARHAN

Supervisor

Submitted to

Graduate School of Science and Engineering of Hacettepe University

as a Partial Fullment to the Requirements for the Award of the Degree of

Master of Science in **Computer Engineering**

2024

ABSTRACT

DETECTING NOVEL BEHAVIOR AND PROCESS IMPROVEMENT WITH MULTI-MODAL PROCESS MINING

Abdurrahman TELLİ

Master's Degree, Department of Computer Engineering
Supervisor: Doç. Dr. Ayça KOLUKISA TARHAN
Co-supervisor: Dr. Tuğba GÜRGEN ERDOĞAN
July 2024, 99 pages

The importance of data has increased, especially with the spread of Internet of Things (IoT)-like technologies as a result of the 4th Industrial Revolution. In order to make sense of the data, valuable sciences have also gained importance in this direction. Data science is one of these sciences. Although it differs from data science in order to make sense of the data, process mining (PM) plays a dominant role in modeling process-based systems by intersecting with data science. When PM intersects with data science and includes other perspective data in the analysis, problems may occur in performing error-free operations due to data science-related errors. One of these processes is; to detect new behavior with the combination of other perspective data (e.g., data perspective). For existing studies, multi-perspective analysis can be performed with the support of data science by including other perspective data in the analysis.

Within the scope of this thesis, by adding other perspective data to the control-flow perspective, multi-modal analysis can be carried out with a PM-intensive approach without loss of context. Multi-modal analysis is the analysis of an activity in combination with the attributes that affect that activity. In addition, instead of using ready-made event logs, event logs can be produced through virtual factories that shine with the 4th Industrial Revolution, and improvement can be made on modeled processes without data from real processes.

Essentially, with this strategy, the gap for approaches that use other perspective data with loss of context is filled, but unlike existing process improvements, process improvement can be worked on without waiting for real system event logs. With these advantages, a PM framework that can produce event logs between independent units (i.e., from producer to processor) is proposed. The transmission of these logs between units in a distributed structure is carried out with a broker-based architecture. Root-cause analysis (RCA) can be performed with the help of metrics through other perspectives added to the control-flow perspective for the taken records, and the detection of new behavior in desired level can be done with multi-modal analysis, without loss of context, unlike especially multi-perspective analysis with data exploration mode.

In order to verify the framework within the scope of the thesis, an event log is produced via the Business Process Modeling Notation (BPMN) model found with the repairExample.xes event log. Verification is carried out for the generated event logs with multi-perspective BPMN and multi-modal process discovery. Event logs that are verified are transferred to the processor part. The records taken in the processor part are first written to disk. Afterwards, one of the new event logs is selected and root cause analysis is performed. Through other perspective data added to the control perspective, root cause analysis is done multi-modal. Token and alignment based replay is used for predictor and response. If necessary, process improvement is provided through model repair. Fitness is the most important metric for improvement. Precision and generalization can provide insight into the system. In time-related mode, analysis can become a hybrid with variant and query support.

Keywords: Process mining, software architecture, machine learning, multi-modal analysis, quality in process mining, root cause analysis, process enhancement.

ÖZET

ÇOK MODLU SÜREÇ MADENCİLİĞİ İLE YENİ DAVRANIŞIN TESPİTİ VE SÜREÇ İYİLEŞTİRME

Abdurrahman TELLİ

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Danışmanı: Doç. Dr. Ayça KOLUKISA TARHAN
Eş Danuışman: Dr. Tuğba GÜRGEN ERDOĞAN
Temmuz 2024, 99 sayfa

Özellikle Dördüncü Sanayi Devrimi sonucunda Nesnelerin İnterneti (İng. Internet-of-Things (IoT)) gibi teknolojilerin yaygınlaşmasıyla birlikte verinin önemi artmıştır. Verilerin anlamlandırılması için değerli bilimler de bu doğrultuda önem kazanmıştır. Veri bilimi de bu bilimlerden biridir. Süreç madenciliği, veriyi anlamlandırmak açısından veri biliminden farklılaşsa da veri bilimi ile kesişerek süreç tabanlı sistemlerin modellenmesinde baskın bir rol oynamaktadır. Süreç madenciliği veri bilimi ile kesiştiğinde ve diğer perspektif verilerini de analize dahil ettiğinde veri biliminden kaynaklanan hatalardan dolayı hatasız operasyonların gerçekleştirilmesinde sorunlar yaşanabilmektedir. Bu süreçlerden biri; diğer perspektif verilerinin (örneğin veri perspektifi) birleşimiyle yeni davranışın tespitidir. Mevcut çalışmalar için diğer perspektif verilerinin de analize dâhil edilmesiyle veri biliminin desteğiyle çok perspektifli analiz gerçekleştirilebilir.

Bu tez kapsamında, kontrol akış perspektifine diğer perspektif verileri de eklenerek süreç madenciliği yoğun bir yaklaşımla, bağlam kaybı olmadan çok modlu analiz gerçekleştirilebilir. Çok modlu analiz, bir aktiviteyi etkileyen diğer niteliklerin kombinasyonuyla yapılan analizdir. Ayrıca hazır olay günlükleri kullanmak yerine 4. Sanayi Devrimi ile parlayan sanal fabrikalar üzerinden olay günlükleri üretilebilmektedir. Gerçek süreçlere ait veri olmadan modellenen süreçler üzerinde iyileştirme yapılabilir. Bu stratejiyle

temel olarak, bağlam kaybıyla diğer perspektif verilerini kullanan yaklaşımlar arasındaki boşluk doldurulur, ancak mevcut süreç iyileştirmelerinin aksine, gerçek sistem olay günlüklerini beklemeden süreç iyileştirme üzerinde çalışılabilir. Bu avantajlarla bağımsız birimler arasında (yani üreticiden işlemciye) olay günlükleri üretebilen bir süreç madenciliği çerçevesi önerilmektedir. Bu günlüklerin dağıtık bir yapıda birimler arası aktarımı broker tabanlı bir mimari ile gerçekleştirilmektedir. Alınan kayıtlar için kontrol-akış perspektifine eklenen diğer perspektifler aracılığıyla, metrikler yardımıyla kök neden analizi yapılabilmekte ve özellikle çok perspektifli analizin veri keşif modundan farklı olarak bağlam kaybı olmadan çok modlu analiz ile yeni davranışın tespiti istenen seviyede yapılabilmektedir.

Tez kapsamında çerçevenin geçerlenmesi amacıyla, repairExample.xes olay günlüğü ile bulunan BPMN modeli üzerinden olay günlüğü üretilmektedir. Üretilen olay günlükleri için çok perspektifli BPMN ve çok modlu süreç keşfi ile geçerleme gerçekleştirilmektedir. Doğruluğu görülen olay günlükleri, işlemci kısmına aktarılmaktadır. İşlemci kısmında alınan kayıtlar öncelikle diske yazılmaktadır. Sonrasında yeni olay günlüklerinden biri seçilerek, kök neden analizi kontrol perspektifine eklemlenen diğer perspektif verileri aracılığıyla çok modlu yapılmaktadır. Tahmin ve yanıt için simge ve yerleşim tabanlı geri oynatım kullanılmaktadır. Gerekli durumda, model tamiri yoluyla süreç iyileştirmesi sağlanmaktadır. İyileştirme için uygunluk en önemli metriktir. Kesinlik ve genelleştirme metrikleri, sistem hakkında fikir sağlayabilir. Zamanla ilgili modda analiz, varyant ve sorgu desteğiyle hibrid hal alabilir.

Anahtar Kelimeler: Süreç madenciliği, yazılım mimarisi, makine öğrenimi, çok modlu analiz, süreç madenciliğinde kalite, kök neden analizi, süreç geliştirme.

ACKNOWLEDGEMENTS

I am grateful to my supervisor, Assoc. Prof. Dr. Ayça KOLUKISA TARHAN, for her support and patience during my thesis journey. I would also like to thank to the members of the thesis committee, Assoc. Prof. Dr. Ebru GÖKALP AYDIN and Assist. Prof. Dr. Özden ÖZCAN TOP, and my co-supervisor Tuğba GÜRGEN ERDOĞAN for their reviews and suggestions to enhance this study. I am also thankful to my father and my mother for their love and support that they have always provided to me.

CONTENTS

TABLES

FIGURES

ABBREVIATIONS

BPMN	Business Process Modeling Notification
CMMI	Capability Maturity Model Integration
CSV	Comma Separated Values
DSR	Design Science Research
FN	False Negative
FP	False Positive
FPR	False Positive Rate
GQFI	Goal Question Feature Indicator
IoT	Internet of Things
ML	Machine Learning
MXML	Mining Extensible Markup Language
OCEL	Object Centric Event Log
PDM	Process Diagnostic Methodology
PM	Process Mining
PM^2	Process Mining Project Methodology
RCA	Root Cause Analysis
ROC	Receive Operating Characteristic
SLA	Service Level Agreement
TN	True Negative
TP	True Positive
TPR	True Positive Rate
WF-net	Workflow Network
XES	Extensible Event Stream

1. INTRODUCTION

Industrial revolutions cause development and change in the universal sense. Within the scope of this thesis, primarily as the underlying motivation, first, the interaction of Process Mining (PM) with the effect of industrial revolutions will be discussed.

As it is known, with the 1st Industrial Revolution, the invention of the steam engine took place, and then the production method that people made by labor was transferred to the machines. It can be seen as the first step from the rural world to the industrial world.

With the 2nd Industrial Revolution, machines that work in a simple sense were supported by more powerful technical equipment. As a result of the invention of electricity and its transfer to machines, mass production lines were established. With the use of new energy sources such as oil in production alongside electricity, it can be seen that mass production will no longer be abandoned.

The 3rd industrial revolution has been electronic and information technologies intensive. Important developments for this Industrial Revolution are the use of the first micro-level computer Altair (8800) and the establishment of Apple in 1976 by S.Jobs and S.Wozniak. There have been developments that have allowed the development of renewable energy sources. Computer-aided monitoring and management of these resources has led to the development and operation of the idea and its use in production.

With the 4th Industrial Revolution, which can also be called Industry 4.0, the rise of robots has taken place. The idea of using computers and electrical equipment in the production line has found a ceiling. It is possible to talk about a cyber-physical world created by sensors and actuators with physical systems. Thus, digital monitoring of any physical event is provided rather than an object that leaves a fingerprint.

As a result of all these experiences, the importance of data has increased and the concept of big data has come to the fore. Fields such as data mining and Machine Learning (ML) are focused on this bulk of data. Although it is possible to work and analyze with data, PM [1] has come to the fore as a result of the need to focus on the nature of the process.

1.1. Digital Twin

For any tangible or intangible asset that can be stored in a digital environment and is not digital, its copy in the digital environment can be called a digital twin. Virtual factories can also be considered within the scope of the digital twin concept [2].

With the 1st Industrial Revolution, the understanding of production based on agriculture and manual labor has left its place to simple machines. With the 2nd Industrial Revolution, the concept of factory can be mentioned. The 3rd Industrial Revolution resulted in the digitizing factory phenomenon with the addition of computer-based data to factories. The 4th Industrial Revolution, on the other hand, made virtual factories and smart factories active, interrelatedly.

The definitions and types for the virtual factory are grouped under four categories [3], as described in the following paragraphs.

According to the first category, they are the types in which all aspects of a factory are integrated into a simulation model. One definition is a test environment for designing and operating production systems with high accuracy. The other is systems where software, tools and methodologies provide unity and integrity in order to provide solutions to problems in the field of production. It can be said that it contains the dynamics of the technical infrastructure.

The second category is the partnership formed by many virtual organizations in product production. In other words, it is the collaborative sharing of simulation-based designs specific to a product via network.

In the third category, there are virtual reality applications that investigate the impact of product groups, employee experiences and reviewers on productivity via three-dimensional and visual way.

The fourth category is factories that offer the opportunity to emulate production activities. It provides the opportunity to test operational, tactical and strategic decisions with object behavior modeling and interactive decision support system.

In this thesis, unlike virtual factory types that develop starting from production activities, any process representation is categorized as a virtual factory. It is related to them in that it contributes to the previous definitions and types.

1.2. From Related Disciplines to Multi-modal Process Mining

Although data science is seen as a continuation of statistics, data scientists do not contribute to the field of statistics. Statistics focuses on theoretical results rather than real-world problems. It is about data quality and size, and generative modeling, rather than prediction. Statistics is the origin of data science.

There are opinions that argue that data science originates from database and ML originates from artificial intelligence [1,4]. Data mining is used to extract unsuspecting relationships and summarize data on large data sets [4]. Statistics is built on databases and algorithms. Compared to statistics, the focus is on scalability and practical applications.

Nevertheless, the difference between data mining and ML is ambiguous. The ability that ML algorithms provide to the computer is that it draws models that learn from experience rather than explicitly coding. The evolving model is used for prediction and decision.

In PM, where evidence-based transactions are made, the approach of basing information on evidence differs as associating information with process-centered evidence.

Both data science and statistics try to understand and obtain information from data. It is possible to make predictions for the future with both of them. It is possible to say that data science comes to the fore with large-scale data, allows pattern recognition, provides meaningful and understandable information, and these are useful.

At the point of purpose, data science and statistics both perform the process of analyzing from data and learning from it. It can be said that more methodological approaches are used in data science. Operations such as making sense of the data and determining the factors affecting a result can be done with both.

Looking at the analyses, it will be seen that methods such as clustering, regression and correlation are used in both statistics and data science fields.

Although data science originates from the science of statistics, they are different fields from each other. The fact that the methods on which it is based are based on statistics does not necessitate seeing data science as a sub-field.

In statistics, there are hypotheses established after preliminary examinations. These hypotheses are included as alternative and null hypotheses. These established hypotheses are clearly stated as a requirement of the confirmatory [5] method. In data science, even if there are hypotheses, they are a mathematical function of the output and may not be explicitly stated. The data to be analyzed in statistics is collected for a specific purpose. In data science, on the other hand, analysis is usually made on the data that is available. For this purpose, in statistics, a hypothesis is established based on prior knowledge. As a result of the analysis, the validity of the hypothesis is examined. In data science, the findings as a result of the analysis are examined and scrutinized.

The approaches used in the analysis also differ. In data science, the entire data set of interest is generally handled and as a result, more specific-local data is reached. It can be said that deduction is made with the method of knowledge discovery from general to specific. In statistical analysis, analysis is made with the population selected from the data set and the results are generalized.

Despite the existence of differences, attempts are made to make sense of data with data science and other related disciplines. With the focus on production activities, different approaches are needed such as regression and clustering that are used to make sense of data in data science. A solution has been found with the introduction of PM that makes a significant contribution to the examination and improvement of processes in many sectors [1,6]. PM is to provide analysis by obtaining process models from event records. Manual or automatic process discovery from event logs, comparison of event logs with the existing model with conformance checking, and improvement of the process by developing or improving the basic model are provided. Event records can be obtained from digital twins or virtual factories instead of real systems. As a result of advances in technology and industrial revolutions, especially with the Internet Of Things (IoT) systems brought by the 4th Industrial Revolution, other characteristics of a process, rather than just the behavior that characterizes the process, is gaining importance.

Multi-perspective PM provides a viewpoint-oriented examination of the processes [7,8] from five perspectives. Discovery of the process with the control-flow perspective, detection of deadlocks over time with the time perspective, and identification of roles with the resource perspective are examined. Data perspective is related to the other data attributes and function perspective maps high level events and low level activities. Conformance checking relates to the process model's fitness ability to reflect an event log. Multi-modal PM, on the other hand, is the analysis of an activity in combination with the attributes that affect that activity. Rullo et al. [9] make an analysis that includes other components of the multi-perspective PM related to multi-modal analysis. While the main focus is the control-flow perspective, it is enriched with time (time of occurrence and frequency of events), resources (system, human-like) and other operational data (situation) [9]. Rebmann et al. [10] focus on image data and it is understood that even if the perspective does not change, other attributes of the related image give an idea about the behavior [10]. These existing studies [9,10] perform ML supported process analysis and provide a post embedding difference detection with other process data. By performing the feature combination with ML in the first phase of the analysis, differences are extracted about events.

In this study, unlike other studies, the modes that will be formed by considering one of the other perspective components of the process (e.g., data perspective), together with the control-flow perspective, are seen as a category of modularity. Thus, modes related to each behavior can be created. Multi-modal process models are kept and it is possible to see which behavior differs with which attribute or resource. With the definition that a modal is one of a series of categories in a modality, it can be used in different places.

More concretely, in the existing studies, multi-modal expression can be used due to data modularity from different sources. The second use of multi-modal, which is formed by the combination of perspective data and modes under modularity, can be called a modularity name (e.g., multi-perspective data can be stored in one or more dimensional data structures and have different modes with case modularity, even when the activity sequence is the same). In this study, the control-flow perspective behavior combined with other perspective data has modes in the control-flow perspective under the multi-perspective behavior modularity. To do this, distributed PM with broker-based architecture [11] is supported by multi-modal analysis.

1.3. Research Purpose

This research is carried out for academic purposes following a design science approach [12]. It is thought to answer the identified research questions and propose solutions.

The main purpose of the research is to support multi-modal analysis in PM, which can be located at the cutting edge of technology. Additionally, we offer a framework that performs other PM operations. For this purpose, the research questions are given below.

Research Question 1: What are the existing studies in multi-modal PM?

Before any study is carried out, it is expected to examine selected and distinguished studies in the literature related to that field. Although there are rare articles in this direction, the literature review was carried out to include related studies.

Research Question 2: How to discuss the structure designed in the context of digital transformation and smart-virtual factory concepts?

In order to find an answer to the question, in the introduction part, digital transformation and Industrial Revolutions are examined within the frame of historical sequence. In the digital twin part, concepts such as digital twin and virtual factory are mentioned. Accordingly, the framework designed and described in Section 4 will be discussed in the conclusion section.

Research Question 3: Can the detection of new behavior be studied with multi-perspective analysis?

The term multi-perspective mining is frequently used in articles. Even if it is called multi-modal mining, it can contain reference to multi-perspective structure. For this reason, both concepts are tried to be examined.

Research Question 4: How is improvement achieved through quality criteria and root cause analysis in the multi-modal approach?

Root cause analysis will be tried to be examined through quality metrics for enhancement, which is one of the three main pillars of PM.

1.4. Research Structure

Firstly, a detailed and understandable background information will be given in Section 2. Studies on multi-perspective PM, which are close to the multi-modal structure and at the same time different from it, studies on root cause analysis in PM and studies on multi-modal structure will be covered in the related work (Section 3). The framework will be conveyed in Section 4. In the case study part, the repair process for the telephone company will be processed in accordance to the purpose in Section 5. Assumptions and constraints related to this study will be discussed in Section 6. In the latest Section 7, the answers to the research questions will be reviewed and concuding remarks will be provided.

2. BACKGROUND

This section provides all background information for the reader. The event log format used within the scope of the thesis, process model representations, the definition of PM, the PM techniques and approaches used and importance of decision making are explained. In addition, details of the developed distributed architecture are also included.

Within the scope of the study, PM techniques are used with multi-modal analysis. Multi-modal analysis is the analysis performed through the combination of multiple components of the multi-perspective structure with zero angle. It uses perspective data similarly to the multi-perspective structure. In multimodal analysis, the angle difference of other perspective data added to the control perspective is the main difference compared to multi-perspective analysis.

Root cause analysis is the analysis performed to determine why a problem occurs. In order to realize the improvement purpose of PM, root cause analysis is carried out through a multi-modal structure. It is based on metrics.

2.1. Event Log, Trace and Event

Process oriented systems derive the rules for these processes by monitoring the events extracted from their operating information. The form in which this information about the operation is kept is the event log. Abstracting from system specificity, an event log exists as a set of traces.

Separate event records can be found for each process. However, if the same event log is used for many processes, there is an additional identity (ID) specification for event logging.

A trace consists of trace attributes that describe its unique properties at a global level, and an ordered sequence of events that describe the operating activities of the process. A case identity specification can also be used instead of a trace identity specification. Any sequence of events that share the relevant trace ID specification is an instance of the same trace [1].

An event is an atomic process unit selected from a set of activities and characterized by attributes such as name, timestamp, and resource.

A process includes traces with a holistic view. A trace contains events, and the event sequence is unique to a trace. Traces within events are sequential. Events contain attributes that characterize them.

2.2. XES

Event records can be obtained from a complex x-ray device, web browsing, or from a company. Mining Extensible Markup Language (MXML) was used as a standard structure for the existence of files to be analyzed in such a wide range. As the successor to MXML, Extensible Event Stream (XES) is seen as less restrictive and more extensible.

An XES document contains an event log containing many traces. Each trace contains an ordered list of events specific to the relevant situation. The event log, the trace of the event log, and the events of the event log can have attributes. Core types for event attributes are string (String), date (Date), integer (Integer), decimal number (Float) and logical (Boolean).

The activity name, which can be defined at the global level for event log, and when defined, attributes such as timestamp and resource are mandatory for all events. While these fields are located at the global level, they define semantic types. For example, with the name org:resource, the string type is also specified, indicating that there will be a string type resource attribute in the events.

The attributes of the event can be used as classifiers [1,13]. It can be viewed as a function. The function assigns the value it receives as input to the class it matches. It is included in the event record size with name and prefix information. At the event level, tracking in XES formatted event records can be achieved through key and value matching.

Event logs can exist in various formats such as Comma Separated Values (CSV), or Object Centric Event Log (OCEL) [14] different from XES.

2.3. BPMN

Business process management is concerned with increasing the efficiency and effectiveness of businesses. Activities it can monitor include design, modelling, operation and optimization. Problems in the design of the process may lead to unexpected surprises and hinder gains. Therefore, the improvement process takes place in a cycle and includes continuity.

Modeling has an important place in business process management. While the management focuses on the general, modeling and visualization and graphical presentation are provided. Business Process Modeling Notation (BPMN) [15], which can be considered as a modeling demonstration, aims to involve business stakeholders in the process. BPMN components have in a common language that concerns all stakeholders.

Task can be accomplished with one or more resources at a predetermined time; it is an atomic unit of work [16]. In cases that require many stages, these will be called activities [16]. According to the business process management system, which focuses only on management issues, the task is an external work. The resource can be a human or an inanimate entity.

General BPMN components are gate, connectors and events [16,17]. Gate represents the way in which sequential flows converge or diverge in the context of communication when considering sequential task flow. Connectors show the flow between components in the graph. There are types such as message flow, activity flow, and relationship flow. Events are things that start, finish or trigger the process and happen.

BPMN is the defacto standard for modeling processes. Representation of existing processes at the conceptual level and in a structure that will affect all stakeholders can be provided. From this state, it will go to the state that should be in the future, and it provides a starting point for it. In addition, they enable to start work with assumed processes and to make improvements as a result of simulations.

2.4. Petri Nets

Petri nets are a graphical and mathematical modeling framework that can be applied to discrete event systems [18]. Petri nets provide representation of systems with simultaneous, distributed and asynchronous characteristics. As a graphical tool, Petri nets can be used as

visual communication aids similar to flow charts. Symbols are used to simulate the dynamic and simultaneous activities of systems. The term system refers to software, hardware, and physical systems as well as social systems.

Thanks to its inherent generality and permissibility features, its applicability is also recommended for a wide variety of applications. The trade-off between model generality and analytical ability must be carefully considered. The enlargement of the generality measure of the model may affect the analysis capability.

The basic components of the Petri net graph are places and transitions. Relationships between these transitions and places should be shown. The arc is located as the structure that connects the transition and places. An arc does not connect components of the same type. Transition nodes can be connected to ground nodes or ground nodes can be connected to transition nodes.

2.5. Process Mining

Before defining the PM, it will be useful to refer to the event logs. An event log contains the events related to any event that occurred, the activities related to these events and various attributes of the activities.

These logs are available in various formats. With the use of information, processes can be discovered and modeled, monitored and optimized. With the provided event records, process discovery is created algorithmically, and with this model, the incompatibility control and development of the process can be achieved.

PM can be defined as a management system that enables the use of event logs recorded in systems [1,6].

Another definition can be the field of technology that detects and repairs inefficiencies, deviations and improvement points using logs.

The most frequently used tools for PM can be given as ProM, Disco and pm4py [19]. ProM stands out in terms of its ability to produce results through input and output. Disco, which provides ease of use similar to the ProM software, has a disadvantage in that it is a commercial application. Pm4py is preferred because it provides customizable use.

The three main purposes of PM are process discovery, conformance checking and process improvement. There are also other usage areas/purposes such as multi perspective PM, which are applied and referred by multi-modal structure to in this thesis. These will be explained in the following sections.

Algorithms are derived from event records by one of the three main purposes of PM which is process discovery. Process discovery is done to create a model representation of the process. It creates a graph representation of the process depending on the algorithm it uses from the event logs it receives as input. Frequently used algorithms are Alpha Mining, Heuristic Mining, Fuzzy Miner and Inductive Miner algorithms.

2.5.1. Process Discovery

Process discovery generate model representation of the process from event logs take as an input.

2.5.1.1. Alpha Algorithm

The alpha algorithm creates a petri net over four basic relationships. Based on these relationships, a digital footprint matrix is created and a solution is reached. The Alpha algorithm has the problem of dealing with noise, incomplete and infrequent behavior [20].

The logic of its operation can be understood through the four basic relationships on which the algorithm is based. These are explained below as linear relationship, causality, parallelism and selection.

Linear Relationship: It is defined as one activity following another activity.

Causality: If there is a linear relationship between activities a and b, but there is no linear relationship between b and a, there is a causal relationship between these two activities.

Parallelism: If there is a linear relationship between activities a and b, and there is a linear relationship between activities b and a, there is a parallel relationship between a and b. Considering that a third activity c comes before these activities, it can be said that there is a parallelism between activities a and b, as there are causal relationships between activities c and a and activities c and b. Similarly, if the third activity c comes after these two activities,

it can be said that there is a causality between activities a and c, while there is a causality between activities b and c, and there is a parallelism between these two activities.

Selection: If there is no linear relationship between both activity pairs a,b and b,a, it can be said that there is a selection relationship between these two activities.

2.5.1.2. Heuristic Miner

It uses a notation similar to causal networks and considers event frequencies. It is a practically applicable mining algorithm that can deal with noise and can be used to express the main behavior (i.e. all details and not exceptions) recorded in an event log [21].

The number of times each event is followed by another event is weighted by the number of times the event occurs as a result of any given event. For the other two activities following the first of the three activities, it can be said that the connection with the first activity is stronger for the activity that follows the first activity at a higher rate. In other words, if an activity is followed by another activity, there is likely to be a dependency. In order to determine the degree of this dependence, the event log is analyzed in terms of causal dependencies.

2.5.1.3. Fuzzy Miner

It was introduced to provide a more abstract and high-level presentation of complex processes. It proposes the metaphor of the map, drawing on the field of cartography, which has overcome similar difficulties before. The locations in the topology can be matched with the activities and the paths connecting these locations with their degree of importance.

It uses two important metrics to show the activities and the links between activities. One is the degree of importance and the other is the correlation. Based on these two metrics, the important behavior is retained in the model. Less important but highly correlated behaviors are shown within the cluster. Less significant and low-correlated behavior is abstracted from the model [22].

The algorithm is universally accepted as configurable. However, finding the right parameter settings can be time consuming. It is an algorithm that shows that the exact behavior of the

process and the algorithm aim to present a more understandable and simple process to the user are contradictory to each other. It is also seen that the approach to handle complex models cannot be fully achieved with this algorithm.

2.5.1.4. Inductive Miner

It creates a process tree from the process model it takes as input. Process tree is a rooted tree in which leaves are explained as activities. Information about the execution order of activities is obtained from the internal operators of the process tree. These are sequential composition, exclusive choice, parallel composition, and iteration loop.

All children of the sequential operator are executed from left to right. Only one of the children of the exclusive choice operator (XOR) is executed. An operator whose children can all be executed but whose execution is in any order is parallel composition. There are children divided into two groups by the iteration loop operator. The process always starts and ends with do part. Because when the operator is activated, do part is executed at least once, and if there is a redo part executed afterwards, do part follows it [1,13].

The algorithm puts the event log into a linear form with the sub-event logs it divides. In other words, it implements sub-solutions for the cut points it detects by means of operators and operates the resolved sub-sections in sequential form. Thus, it uses minimal information to represent the process model. It offers algorithm complexity that provides a viable computational cost. It also facilitates the solution of the variability of complex processes by removing the deviations from the model.

Other PM algorithms allow intervention in the scope of analysis through interactive interaction. However, they only partially produce models with executable semantics. The algorithm, unlike other algorithms, provides improvement by following the states by using tokens in the representation.

2.5.2. Conformance Checking

A process is a general concept of behavior, that is, the coordinated execution of a set of activities to achieve a specific goal and result.

The process concept alone may not provide a certain level of automation. We are talking about a manual operation when this level of automation is not provided. Namely; morning activities that we perform as a daily routine can create a process. It starts with getting out of bed, it can end with straightening the bed, stopping by the bathroom, preparing breakfast and having breakfast. Here, too, the activities take place in coordination, but the subject of the activities is human, that is, they are done manually. In the case of automation, it is possible to talk about the existence of a software as a part of the whole or as a system component.

The process model provides information about the context of the execution of a process with a conceptual model. It is possible to say that there must always be an abstraction here. It provides a general understanding of the process as well as a representation of the set of features that can affect the context of the process.

Process can be represented as a process model or an event log. Conformance checking refers to the analysis of the relationship between a behavior described by a process model and the event logs that can be obtained from the execution of the process [23,24,25,26].

Previously, the 4th Industrial Revolution was mentioned. The developments related to the 4th Industrial Revolution should not be seen only as intended. Many companies have already successfully started the individual digitization process towards these aspects. As in agile software development, it is possible to talk about a phenomenon such as agile process management. With agile process management, it is possible to talk about a dynamic way of conforming operations to legal requirements after the processes are explained with models. It is known that companies try to implement a series of rules in order to have Capability Maturity Model Integration (CMMI) certificate. They are subject to an audit to show that these defined rules are followed and eventually they have a certificate. In such an important situation, the regulations related to CMMI may be modeled by the company and they may go into a state of self-regulation without auditing yet.

There is a situation where the automation that came with the 4th Industrial Revolution accelerated and slightly differentiated. We consider the case where the model was designed manually despite the technology. In such a case, the difficulties brought by the theory come to mind. There may be cases where even algorithms cannot identify the desired relationships. When this is the case, there are problems such as not being able to capture the required level of detail and not being able to fully characterize the process change.

It should be seen that process models provide decision support through the automation and insight they provide. In cases where this support is based on estimation, results are often made with ML and artificial intelligence support. As the name suggests, the rules are produced tolerantly over the estimable measurable values that remain a bit gray. These disciplines can be used in conjunction with PM for detailed analysis. However, there may be situations where we need to provide clearer insights. In such cases, it is possible and likely to use PM techniques together with clearer solutions and to provide human-assisted decision support without neglecting the human being.

We resort to the concept of deviation when analyzing the relationship between the process model and the event log through conformance checking. The detection of deviant behavior is often considered to be a bad sign, as it is defined as a way of acting different from the general way. In some cases, deviations may be positive or not need to be eliminated. One of these situations is having a wrong idea about the reference process. These additional behaviors that are not included in the reference process may be necessary to complete the task. Another situation may be related to stakeholder satisfaction that concerns the process. It is a deviation, but positively, to give priority to those who are customers of the bank in a process that involves a person who wants to make transactions at the bank by taking a queue from the kiosk device.

At this point, a definition can be made about conformity control; it is a technique that provides a detection, especially by including deviations, to create methods and tools to provide fact-based insights by combining process model-based approaches and data mining.

Again, close to the concept of big data, PM has some features that include the data it will obtain and process. It is normal for the event record to be large when the number of realization samples per unit time of the data obtained is high. This property can be called volume. With the importance of the IoT, it is obvious that the data does not come from a single source. In this way, the data obtained from heterogeneous sources brings the diversity feature of PM to the fore. The third feature is speed. This feature should not be considered only in relation to the transfer of data. It indicates the speed of action to be taken regarding inappropriate behavior arising from data [26].

Considering these features, it is seen how important the automation element is. We will now talk about the relevance of people to this conformity detection, although manual PM

techniques do not perform. While people analyze complex data, they can add new information to the results they get from software and create value for their organizations. At this point, especially when we think about deviation, the notification of these values to people by software applications can also be provided through visual interfaces. Often deviant behaviors are presented using different coloring. Thus, it is possible to make a quick analysis by non-technical stakeholders.

It should be the task of PM applications to present the analysis results, reveal the relationships to be obtained with the data, and provide undirected presentation. Through this presentation, conformance auditing “why?”, “how?” and what?" questions can be focused [27].

We state that the questions asked here can also be asked in multi-perspective analysis, but they do not fully overlap. The "how" question here is to evaluate how the feedback was provided to interpret the analysis. Even a user who does not have much technical knowledge can detect the deviations of the flow with the help of coloring, usually on the visual output. In case the output provided by the analysis result can be numbers, text and graphics, the answer to the question can be one of three types. The “what” question often provides a data-centric presentation. The “why” question is about getting more detailed information. It summarizes the task performed. Different from the task in the process model, a task within the scope of the analysis is mentioned. It can be exemplified as measuring compliance, detecting deviations, explaining and diagnosing deviations.

2.5.3. Multi-perspective Process Mining

Multi-perspective mining goes beyond the activity column in the event logs to perform conformance checking. It examines the attributes related to time, resource and data from the perspective of the title it is related to. It would be appropriate to give another definition of conformance control here. The conformance checking reveals deviations between the actual process and its representation in any context, or within the representation itself.

2.5.3.1. Control-flow Perspective

The control-flow perspective is concerned with sequencing events. It fulfills the purpose of characterizing all possible paths with Petri net or some other notation. The model created by

the control-flow perspective should provide the order of activity of the process in a realistic manner. However, since the configuration of the model differs from algorithm to algorithm, the constructed model may not conform to reality. Therefore, the inconsistencies between the model and the event log file are checked and their relationship is observed with the conformance checking.

It is stated that many PM approaches focus on this perspective and focus on process discovery and conformance control [1].

The exploratory task realization of the control-flow perspective is referred to as process discovery. The process discovery algorithm can be viewed as a function that maps the event log file to a model that provides a representation of the events in that file [16].

It is seen that the control-flow perspective is often based on Alpha Mining, Inductive Miner, Fuzzy Mining and Heuristic Mining algorithms. Especially after the discovery made with the Inductive Miner algorithm, the deviating behaviors show the incompatibility.

It usually focuses on the "how" question and provides insight into what the process looks like.

2.5.3.2. Time Perspective

The time perspective is related to the time of occurrence of events. Event logs contain timestamps. The timestamp allows seeing deadlock points, service level analysis, resource usage status, and estimating the remaining uptime of operational states.

The most important use of timestamps is replay. Working times can be calculated for all samples by considering the start and end times. In the places between activities, there is information about how long a token will stay when it comes to replay.

The residence time of tokens for places can be seen as a multiset and statistical operations can be performed. In addition, the waiting times of the activities can be calculated by taking into account the end of the previous activity.

From here, the playback logic offered by the time perspective provides performance clues from the visualization of service and wait times, analysis of deadlock points, flow time and

Service Level Agreement (SLA) analysis, and frequency and resource analysis. Clarifying activities with a lot of waiting time is an action of visualizing service and waiting times. For situations that wait too long in one place, deadlock point detection can be made with the information received from the cluster containing multiple waiting time elements. From the routing probabilities, the usage percentage of the passageways is subtracted and in this way, frequency and resource analysis can be made [1].

2.5.3.3. Resource Perspective

Resource perspective focuses on the resource attribute included in the event records. In multi-perspective mining, the equivalent of resource mining is resource perspective. Before starting the analysis, it is necessary to know which activity is performed by which possible resource. The resource that performs the activity can be a system, role, person, or department. Social network analysis definition would also be appropriate since it can be thought of as a relationship between entities in a graph. Social network analysis, similar to network terminology, associates nodes with actors in the network and uses arcs to show relationships between actors. Arcs and nodes can have weights to indicate their importance. With the dotted chart display, a similar display to the social network graph can be provided. A dotted chart is a visual analytics technique that helps find patterns and trends in large data sets. Displays can be provided by showing events on one axis and participants on the other [28].

Depending on the configuration of the network, various terms can be mentioned. Handover of work describes the transfer of work between the participants. Participant B's causally high frequency monitoring of an activity is performed by participant A indicates a strong relationship between participants A and B. The similar task checks whether the participants are engaged in similar work. It is stated that participants who work together on a situation frequently have a higher relationship than those who rarely work with each other. The subcontract specifies a flow from participant A to participant B between the activities of participants A and B performing the two activities. Collaboration refers to the number of times participants work together in the same situation. Reassigning the same activity to a resource is defined as reassignment [29].

The social network graph can be seen as a pattern of connections. With the social network graph, nodes and arcs can have weights. A high weight between two nodes will show that the workload is heavy [28].

With the presentation of the analysis, the business management will be able to see the communication flow more closely. Those who are active during the execution of the work and those who remain more passive can be selected. It will be possible to see who works in which tasks and how many people work in that situation by considering the structuring of the tasks in a special situation. In addition, resource-like organizational elements that can be included in the display can be used to make sense of the critical needs of businesses.

2.5.3.4. Case Perspective

It is known that a decision structure will be formed as a result of processing an event log for modeling purposes. There are decision points in the models that will emerge for event logs that contain possible and probable situations for more than one activity to follow an activity.

The case perspective looks at the case data of the situations and creates such a decision mechanism in a way that will form the characteristics of the situation.

In order to analyze the choices made in past process executions, it is examined which alternative ways are considered for a particular process execution. In the model, starting from the definition point of the selection, the branch where the decision activity is carried out can be determined. While the footprints in the event logs provide a flow in the mapped model, a decision is made as to which alternative path to choose.

It may come to mind that control-flow perspective outputs will be sufficient to perform this process. At the point of making a classification, there is a situation where we only have answers. ML is used with the idea of turning decision points into a learning problem [30].

There may be properties specific to that situation for the relevant situations that cause branching. For cases with an attribute ‘a’, one branch of the decision point can always be operated, and for special cases with attribute ‘b’, the other branch of the decision point can be operated.

The inference of such rules can be done with the decision tree. It should be remembered that the purpose of the decision tree is to arrive at the response variable with predictive terminology when the activity-specific attributes are included as the predictor for the relevant row and the activity as the classification category in a table.

This stage of analysis may present difficulties with data quality. Here, a solution can be reached through the semantic evaluation of the qualities by including human intervention. Considering that dirty and noisy data will lead to wrong analysis, it will be seen that the operation can be evaluated within the scope of improvement.

Finally, analysis including the attributes of decision trees are used for conformance checking. Here, a decision analysis in which the response variable is this distinction can be handled by separating the records that do not comply with the compatible situations [31]. It can be thought that the predictive properties leading to the non-fit response variable are for the cases that tend to deviate.

2.5.3.5. Multi-Perspective Explorer

A multi-perspective explorer is ProM plugin developed by Mannhardt that recommended to make multi-perspective PM practical due to the difficulty of switching between perspectives [8].

The explorer's first impression of the control-flow perspective can be provided through the input model and event log. It provides the paths in the event log along with the frequency of occurrence. Paths with high frequencies are displayed as thick, and those with low frequencies are displayed as thin.

With the explorer, it is possible to switch to data exploration mode under modality, which can be called visualization. With this mode, operations are carried out with the selected traces based on the compatibility coming from the process mode related to the control-flow perspective. Decision analysis can be entered for the guard point related to the selected decision tree classifier. Along with adjustable minimum compliance values, the minimum number of samples can also be determined. A small number of samples may lead to overfitting, while a large sample size may lead to underfitting.

After determining that the process behavior provides the model with acceptable compliance, the examination of the time perspective can be carried out by selecting the performance mode. There may be limitations regarding attributes. It is possible to calculate waiting times in time perspective.

Evaluation regarding the quality criteria of fitness and precision can also be made with the corresponding modes. These quality criteria will be detailed in the following section. Unlike compliance in the control-flow perspective, the impact of decision analysis is also included in the analysis in these modes.

2.6. Importance of Decision Making

Evaluation of algorithms or methods used in scientific studies provides an idea about performance and quality.

In the statistical process, hypotheses are produced over populations. A proof is tried to be made by the method of proof by contradiction. The phenomenon that is tried to be proven is put forward as a hypothesis. This hypothesis is called the alternative hypothesis. With this hypothesis, a hypothesis that is diametrically opposite is put forward, which is tried to be proven to disappear. This is called the null hypothesis.

Two types of errors are evident here. A type 1 error incorrectly rejects the null hypothesis. A hypothesis that is actually true is called false. A type 2 error is accepting a false null hypothesis. The wrong ones of the processes made by the sample are dealt with. False Positive (FP) refers to the negative situation as positive. It is within the scope of type 1 to reject the hypothesis that is established negatively by saying that the sample is correct. False Negative (FN) refers to a situation that should be positive as a negative. At this point, the hypothesis that we accept with the idea that the sample is correct and the result is positive in reality is to lead to type 2 error.

Which error type is relatively acceptable for various situations may vary. Considering an e-mail filtering system, the purpose of filtering is to look for clues to reject the hypothesis. These will be false positives when the mails that the filterer deletes are actually non-spam. The filterer will incorrectly reject the null hypothesis by inferring that there is spam mail (alternative hypothesis), believing that the action is correct. The other case is that the filter

evaluates the hypothesis over the mails that fall into the inbox even though they are spam. For these data, FN results were produced with the approach that it is not spam although it is spam. Based on the accuracy of the produced result, the hypothesis will be accepted this time as well. This means that the hypothesis is incorrectly accepted. According to the types of errors made by the filterer, instead of deleting a very important mail, it can be considered more acceptable to drop a spam mail into the inbox. In this case, type 2 error may be preferred.

Another situation can be considered through hospital data. There may be a patient group that can be concluded as a result of various tests whether they are sick or not. The null hypothesis is that there is no evidence of cancer. A FN result will be produced when the diagnosis to be made after the tests is that the patient is not sick for the relevant group, but is not. Considering that the result is correct, the false acceptance of the hypothesis will be realized. The error made here is of type 2. Similarly, it would be a type 1 error to reject the hypothesis incorrectly by saying that it is sick for the patient group that should be diagnosed as not sick as a result of the test results. In such studies, type 1 error is accepted as it is preferable to learn that there is nothing afterwards, than to live carelessly with the thought of nothing.

When it comes to ML, these errors are reflected in the metrics. These metrics are confusion matrix at classification point, F1 score, Receive Operating Characteristic (ROC) curve, recall, precision, True Positive Rate (TPR) and False Positive Rate (FPR) [32].

The confusion matrix as shown in Table 2.1 evaluates the performance of the model over the test data in the presence of values known to be true. In binary classification, errors can be drawn as 0 and correct ones as 1, as follows. An arrow drawn on the diagonal from northwest to southeast will underline where high values are expected.

Table 2.1. General confusion matrix representation

	Predicted Value 1	Predicted Value 0
Real Value 1	True Positive (TP)	False Negative (FN)
Real Value 0	False Positive (FP)	True Negative (TN)

Many values can be calculated with this matrix. The value of the ratio of true positives to the sum of true positives and false negatives is called the recall or true positive ratio. It shows how accurately the values that are actually correct are predicted.

The FPR is the ratio of false positives to the sum of false positives and True Negative (TN) values. It is the ratio of true values to all predicted values.

Precision can be calculated by the ratio of True Positive (TP) values to the sum of true positives and false positives. It gives how many of the correctly predicted values are actually correct.

As can be seen from the graph in the Figure 2.1, the points (0,0), (0,1) and (1,1) are combined in an ideal ROC curve [32] without false values. It is stated that the behavior of the curve for undesirable performance is to combine the points (0,0) and (1,1). For the ROC curve varying between these two states, the left and upward approach indicates a near-ideal attitude. As can be understood, ROC curves similar to the y=x function are undesirable.

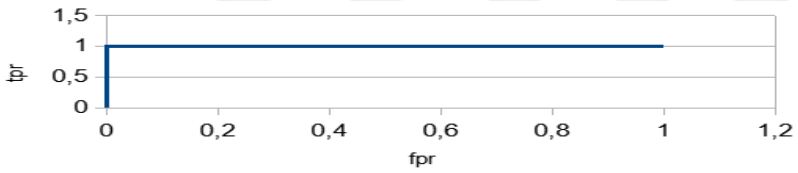

Figure 2.1. The ideal ROC curve

As it is known, classification is made with the aim of getting values below the threshold value determined by the logistic regression function as 0 and above the values as 1. It gives the probability of realization of the prediction with the s-like graph that is the output. Considering that the estimated values are handled in a linear form, it will be seen that there are no false positives and negatives for the ideal ROC as you go from right to left, and there are 0 outputs on the left of the structure and 1 outputs on the right [33]. When moving from right to left, the TPR value will increase up to the first 0 field value. Because when the threshold is lowered, the values classified as positive fall to the right to be close to reality. When the threshold starts to be lowered a little more after the 0 value, the values that are actually 0 will remain to the right of the separator, and there will be no 1 values to the left. Therefore, the logistic model starts to increase the FP after the true positive is constant.

In a linearized distribution with similar logic, the real values from left to right are always negatively distributed up to the ideal threshold. When we take the data in a linear form, the negative values of the classification are in the majority in the left part and the positive values in the right part for the case that the threshold value divides the data right in the middle. On the right, we consider the case where there are several negative values due to the nature of the data. When there is such a model, it can be said that this model is close to ideal. Accordingly, when we start to go from right to left, the increase in the true positive ratio will slow down a bit due to the few negative values that are among the positive values due to the falling threshold values. When the threshold drops to collect positive values to the right, the prediction made by the model is positive, while some actual negative values may become positive. There is a TPR that will be constant after the ideal threshold is reached. If we continue the same process this time, while the actual values remain negative, if we pull the threshold down, the model will produce positive results for the values to the right of the threshold, and this will increase the FPR. Continuing to lower the threshold will increase the FPR of the model.

Incorrect results can also be produced for decision trees, which are frequently used for guard points in PM. Each of the branches of the tree can produce true or false results. Calculations related to these values can be seen with the ROC curve. Accordingly, for a tree with n leaves, (n+1) curves are formed with the optimal ones out of 2^n possible combinations and calculations are made [34]. The area obtained from the curve may be less than 1 and it is obvious that the tree makes incorrect predictions. Advice should be given at these guard points, taking into account the situation. Therefore, the importance of decision making emerges and is brought to the attention of decision makers.

2.7. Replay Techniques

In the petri net, which is a model representation of the process, it is possible to replay the event log, which is another representation, with two different logics [26]. As a result of these replays that are conformance checking techniques, deviation can be detected and quality metrics can be calculated.

2.7.1. Token Based Replay

By manipulating tokens through transitions and places in a petri net, a state matrix can be created based on the presence or absence of symbols in places. The state of the environment can be defined through this matrix. These state changes correspond to the transitions in the model being achieved. It is possible to create a representation of the executions together with the initial state of the process model.

The assumption made is that the traces in the event log have a valid execution order by the process model. As a result of the validation through the model, that is, when the tasks related to monitoring the network are run one by one, it can be seen that there are situations that are incompatible. According to this; fulfilling a task requires consuming a token from the place belonging to that task. However, there is no token in that place. The second situation occurs when the token produced at the exit location of the task is not consumed. As a result of what has been said, it is possible to talk about the existence of four tokens. The produced token goes from transition to place, and the consumed token goes from place to transition. Not consuming the token on the place will make the token a remaining token. If the token that is intended to be consumed by the transition is not found on the place, it means that it is a missed token [24,35].

To illustrate the replay, at the beginning of the repeat, the token produced by the initial marking is incremented. The transition is fired as a result of the transition corresponding to the activity. Tokens are added to the consumed tokens in line with the number of passes from places to transitions. The number of tokens generated from transitions to the next places is increased by the number of tokens produced. If there is no transition corresponding to the activity, the tokens required to enable the transition are added in the appropriate places. After this process, the number of missed tokens is updated. If the last mark is reached after the replay is played, the symbol is consumed by the environment. The number of consumed tokens is updated. If the last marking is not reached, the number of remaining tokens is updated by keeping these tokens in place. The sum of the number of tokens produced or missed during replay is greater than or equal to the number of tokens consumed, and the number of tokens consumed is greater than or equal to the number of tokens missed. At the end of replay, the sum of the number of produced and remaining tokens is equal to the sum of the number of tokens consumed and missed.

As a result of a transition that is expected to be included in the event record at the end of the playback, tokens may remain in a certain place in the model. If the remaining token is excessive somewhere in the model, it can be concluded that an activity that occurs frequently in the event record is not included in the model.

While trying to diagnose such situations, different problems arise that affect correct diagnosis. This problem is a token flooding problem [24]. As a result of the situation being very different from the model, there may be too many symbols added or left in one place. Even if algorithmic intervention in this environment is possible, it may cause undesirable behavior. It will be thought that the correct diagnosis has been made, but there is behavior that leads to misleading diagnoses. A solution to the problem was suggested by freezing the tokens. It is also used within the scope of this thesis. Accordingly, by freezing unnecessary tokens, marking is not done with more tokens than necessary according to the model, it prevents the activation of unwanted parts of the process model and provides a solution to the slowdown problem.

2.7.2. Alignment Based Replay

Alignments show how a trace can be replayed in the process model. The problem here is the concatenation of an event log and a process model. Taking a trace and thinking of aligning it according to all possible traces in the model will make the solution seriously difficult. It increases the cost significantly and approaches the so-called 'brute force approach'. Afterwards, the optimal one among these will need to be considered. Optimality here is the alignment allowed by the model that is closest to the trace. As a result of all these difficulties, the A* algorithm [35], also known as synchronized product network, was proposed to synchronize the trace and petri net. This synchronized product network is a combination of trace and process model, and optimal alignment can be achieved by solving the shortest path problem. The result is reached by calculating the costs on the status graph created by considering the situations reached through the transitions from the process model and the progresses following the trace.

After applying the shortest path algorithm, the layout process is connecting a trace in the event record with the operation in the process model. It is usually represented by a two-row matrix. One row in the matrix shows the progress made for the trace in the event record, and

the other row shows the progress made with the operation of the process model. It is possible to talk about three possible situations [36].

In synchronous progress, the progress includes both the event log trace and the activity name for the process model operation. In other words, a similar operation is provided in the process model for a progress in the trace. The “>>” symbol is generally used only for movement in the trace or only in the process model. For the matrix cell with the activity name, there is no ">>" symbol in the event log, corresponding to the trace, and there is no ">>" symbol in the event log, corresponding to the operation in the process model.

Log move occurs when an activity that should not be executed by the model is executed by an event in the trace in the event log. The activity name that is executed will be used for the progress made in the event log, and the ">>" symbol will be used for the progress made in the process model. This is a deviation. It can be customized in the form of unnecessary operation of the activity.

Model migration occurs when the executed task is not in the trace in the event log while the task is executed on the structure of the model found after discovery.

This progress made by the model indicates a deviation between the process model and the event log, as the corresponding progress has no counterpart in the event log. Therefore, the “>>” symbol will be used for the progress made in the event log, while the progress in the model will be the corresponding activity name.

With alignments, event log only or process model only progresses are assigned a cost that is greater than synchronized advances. This cost can also be thought of as the penalty imposed by the function on these transactions. Reference is made to minimizing this cost by optimal alignment.

If any optimal alignment does not show any deviation, only synchronous progressions can be said to exist. If no deviations are detected, it can be read as a guarantee that the trace actually provides a valid operation of the process model.

2.7.3. Fitness

Process models obtained after process discovery provide permissible operations for event logs. Fitness can be defined as a measure of the process model's ability to replay traces in the event log [36]. Accordingly, a model with a good fitness value will be able to replay many behaviors in the trace. If it is a perfect fit model, it can replay all traces for the event log under review.

The motivation underlying the fitness calculation is the recall metric in ML. Although the underlying motivation is the same, an inaccurate measurement will occur if a crude approach is used to calculate fitness. Thinking at the trace level, it can be said that if the traces in the event log are the same as the traces produced by the model with control-flow logic, the model produces the correct result; and if the trace does not coincide with the production of the model, it produces the wrong result. The case is considered where all traces in the event log differ from the traces in the model by only one activity. Accordingly, the intersection of the event log trace with the trace produced by the operation of the model will be zero. Dividing the value by the length of the trace in the event log will leave the result zero unchanged. With the truly inspired recall metric, true positives will be zero. The sum of the value plus false negatives—in this case, the number of event log traces—will yield the total of true positives and false negatives. It is possible that the result will be zero, which will be similar. It is underlined that within the scope of this thesis, new calculation criteria are used, stating that it is only inspired by the recall metric.

It will be possible to obtain more accurate results when compliance is calculated at the event level. Thus, a calculation is made based on possible events that can be played by the model. The calculation is provided by token based replay and alignment based replay, as explained previously. In both cases, a penalty action is applied. Penalties are applied for remaining and missing tokens in token based replay and for skipped steps in alignment based replay. Ultimately, a compliance check is performed and the important thing is to get an idea about the deviations. While all the traces in the log are replayed on the network, the result is reached by subtracting the penalty points based on the suitability through the remaining and added tokens. In the alignment based approach, the result is achieved by normalizing the optimal fit of the trace to the model with the worst case.

With alignment based compliance control, the trace occurring in the event log, in accordance with the nature of the compliance, is a perfect fit with minimal change from the modeled behavior [35]. Essentially, the modeled behavior is taken as the ground truth, and we can group deviations into recorded behavior but ‘not expected’ and ‘expected to be included in the model’ but not recorded. The first one is caused only by the moves made in the event record, and the second one is caused only by the moves made in the model.

As will be explained later, other conformance metrics deal with the behavior of the model that has not yet emerged, while fitness deals with the behavior that has emerged. We can also explain compliance through alignments by considering that the playback of the movement is currently taking place. Movement progress occurs in the model and results in it are recorded incorrectly or not recorded by the event log. This view clearly reveals the difference with other metrics. Although this is an assumption, it is reasonable and can be used to repair the event log to ensure that events that occurred are recorded correctly.

The section will be concluded by considering the interpretation of any incompatibilities [26] that may occur while performing the conformity check process.

First, the situation where the event record's difference from the model coincides with reality will be discussed. The fact that the actual system behavior is included in the event record and not in the process model will reduce the fitness value of the process model. In this case, since the aim is to accurately represent the real system behavior, it should be considered that the model should be intervened and repaired.

Secondly, the incompatibility that occurs may not be supported by the real system. With this, it is tried to be depicted that the difference of the event record from the model is also different from the system. The interpretation of such a difference is that the event log contains incorrect data. Manual interventions are one of the causes of deterioration. The second reason is that there is noise in the environment even if there is no manual intervention.

2.7.4. Generalization

When learning algorithms perform the prediction process, low bias occurs if the predicted values of the model are very close to the real values. It can be thought that the model, which has the ability to flexibly emulate the training set, has done a very successful job. When the

model is tested with unseen data, it may give poor results. In such cases, bagging algorithms (i.e. random forest) [10] can be used to solve the high variance problem. With bagging algorithms, random features are selected among the features that make up the data set. Afterwards, the results of many algorithms are selected by voting and the most ideal model that works in harmony with the data it has not seen can be selected. In addition, methods such as ridge and lasso regressions, which impose a penalty on the complexity of the model, can also be used. It is also possible to use the features that most affect the output of the model. As a result of all these processes performed and applied, a model with high generalization ability is always tried to be obtained.

The generalization ability of the PM model can be seen with the cross-validation method, which is also used in ML. K-fold cross-validation can be performed by using a certain part of the traces in the event log in training. The part of the model that is not used in training is used for testing purposes to show how well the model supports this part. The process is repeated k times and one of the data divided into k subsets is used for testing purposes and k-1 is used for training purposes. It is clear that the k-1 part changes in each process in parallel with the change of a selected test subset in the k repetitions. The final result can be produced by averaging the cross-validation results. In the meantime, the fitness metric is used. When the model shows high fit with the traces in the test logs, it can be concluded that it has high generalization. In the opposite case, it can be said that it makes low generalization [1].

In order to provide measurements specific to PM, token based replay and alignment based replay are again used. It is thought that with token based replay, there should be more development for less used parts of the model. That's why the penalty it reflects on generalization is high. It is expected that the parts with more operations will reduce generalization less, as they are thought to be more generic. The metric is calculated based on the square root of the operations and normalization is performed with the negative power.

A separate metric is also presented with the alignment based approach. With a similar logic, after an activity transition is achieved, the frequency of repetition of the next activity is checked. It is thought that an activity with a high frequency of repetition will most likely be seen in the future. Otherwise, generalization is expected to decrease further for the next action that is unlikely to be seen in the future [37].

Models provide an abstraction of system behavior. Therefore, there may be behaviors that are seen in the system but are neither recorded nor modeled. Generalization is concerned with minimizing these behaviors. It is expected that the behaviors seen in both the model and the event log and that actually exist are high. However, the fact that the created model reflects the recorded event log exactly but does not have any forward-looking inferences about the system is an overfitting problem. This is not preferred [26].

There are behaviors that are logged and partially modeled by the process model, but do not actually occur. This type of behavior can be described as noisy. It is important to emphasize partially. Because the noise is expected to be low [38]. Thus, instead of transferring the permutation of all possible sequences for a trace by the model parallel to the trace, the metric is derived to meet this logic. In other words, noisy records can be produced when producing automatic event records with PM. These records are also included in the model when it is produced. It is possible that records are included in the model even though they are not in the real system. Modeling this is expected to be low. However, modeling may be required and the problem can be overcome by giving less penalty for less operation with the generalization metric.

2.7.5. Precision

With the fitness metric, a measurement was made regarding the extent to which the recorded behavior was provided by the model. Turning the thinking around a bit, precision is used when you want to measure the extent to which the modeled behavior is met by the event log. It will be high if the model does not contain many more records than the event log, and low if it allows much more behavior than the event log.

With the token based replay approach, parts of the model that deviate from the event log can be detected through the state spaces obtained for the process model and event log. These parts are referred to as escaped edges. During each activity operation, the tasks up to the last marking after the activity are included in the calculation and active tasks are determined for that trace. The ratio of the escaped edge to the active task is calculated from activity to trace, from trace to the entire log [39].

With the alignment based approach, the most appropriate placements of the traces are found by taking into account the event log and model. Afterwards, these alignments are transferred

to the state space and the metric is calculated over the states in the state machine. For each state, the states that the traces allow after it are included as operable activities. With the help of all the situations after the relevant situation, the probability for an activity that can be run after the activity can be calculated. Taking into account the state weights, the ratio of operable activities to suitable activities is calculated by taking all states into account [23].

It will start with the situation where imprecise behavior also takes place in the system. When this situation occurs, the important thing to consider is that the event log is generated incorrectly. This error in the production of the event log can occur due to incorrect documentation of system requirements or incorrect performance of control and monitor components.

There may be imprecise behaviors that are allowed in the model but are not included in either the system or the event log. In such a case, these behaviors do not need to be eliminated from the model to increase accuracy. This may also be included in the model regarding the generalization metric of the model. When activities are carried out in parallel, it may be possible that one of the activities in different order occurs too much and the other occurs relatively little. What appears to be true is that the flexibility of the process is not being fully utilized. However, this uncertainty is indicative of a justified generalization of the model [26].

2.7.6. Simplicity

It may be noted that ML is biased towards an overly simple flat model trying to fit a curvilinearly distributed dataset. If the model becomes too complex and shows too much bending and twisting in new data, it is known as overfitting.

Trying to reduce biases can complicate the model and lead to overfitting. Simplifying the model and reducing biases with a slightly looser fit can reduce overfitting. Regularization is used to avoid these complexities. The effectiveness of regularization in ML is that it prevents further thinking. Once the ideal path to the solution is found, thinking more can lead to worse solutions. Therefore, with regularization, better decisions can be made by thinking less and doing less work.

Simplicity is a measure of leaving behind the complex structures of models of processes made with existing event records as in ML. The elimination of infrequently used roads increases the simplicity as well as the use of generalization. If it is considered on a metric basis, the number of arcs entering and leaving the nodes are added to the arcs' degree of the nodes. With the balance of the number of nodes and arcs, the basic arc degree will be approached, which will reflect the simplicity.

Within the scope of this thesis, metric calculations are made based on the difference between the basic arc degree and the arc degree of the model obtained from the event log. The same metric is used for the token and alignment based approach. Since the token based approach is already on the discovered process model, we proceed through the difference between the two models. When it comes to the placement-based approach, the resulting network event record has some kind of placement.

2.8. Process Enhancement

Process models that have not yet reached their final form are descriptive models. There may be ambiguous requirements in the agile software development process. As a result of negotiations with the customer, processes can be carried out based on what has been agreed. Afterwards, a continuous update can be made depending on the progress of the project. Considering that the process discussed here is modeling the system with requests from the customer with agile software development, it can be seen that the process model reflecting the system capabilities is dynamically updated.

In many cases, process models that satisfy process stakeholders are used. These models include norms and regulations. Process models determined by following these rules or by excluding process parts that involve cost and dissatisfaction in the modeled behavior are called prescriptive models.

Process improvement means making a good or bad process or its representation better [40,41].

A process representation that is modeled solely focusing on the control-flow perspective will not fully reflect reality. It would be a kind of simplification of reality. When talking about data, resources and time constraints, addressing these or one of them will in most cases

ensure that the truth is fully reflected. The absence of this information, although it is needed, will put the representation in a very general form. In such a form, there will be no distinguishing point for more than one representation with the same control-flows. For these reasons, expanding the representation to include other perspective data is called process extension.

Process enhancement includes both process improvement and process extension [40].

Beyond an expansion with data containing only perspective data, expansion can also be achieved through a measurement or use obtained from perspective data [41]. It can be thought that such values may belong to the data perspective. However, it would be extremely useful to point out that although it is not directly the data of this perspective, it can be obtained as a result of additional operations and added to the representation.

By expanding the representation or analysis of the process, the process can be repaired to reflect reality as a result of examinations from other perspectives. If we consider the data perspective, the conditions under which operations can be performed can be determined by combining the decision trees created for the decision points with the guard points. Improvement can be achieved by ensuring that traces containing values that do not meet these conditions are not included in the process. After determining which roles perform which activities through an activity matrix, improvement can be achieved by eliminating resources in the same role that are less than a certain threshold. The expansion made in decision point analysis is on the process model. Analysis expansion can be provided to ensure improvement through resource analysis. Therefore, bringing the process to a better, desired position after an analysis that includes other perspective components is process enhancement.

Depending on the result of the suitability analysis, improvement can be made through the event log or process model that represents the process. While the event log and process model provide representation of the process, inconsistencies may exist. Whichever representation is more reliable can be improved by repairing the other.

Model repair is the transformation of the model by repairing it with the idea that the deviations between event logs and models are caused by the model. A similar situation may also occur due to event logs. This time the representation that needs to be repaired will be

the event log. By paying attention to the requirements indicated by the scenarios, repair of one of the two representations can be achieved.

If the models are the prescriptive ones, the undesirable parts are determined as a result of the analysis. Accordingly, the problems that will arise from giving suggestions to process owners and seeing these parts in the real process are explained. In such a case, instead of providing an improvement in the representation with the idea that the modeled structure reflects the rules and norms, the main aim is to ensure that the actual process complies with certain rules as a result of the analysis. However, it is possible that these models may change as costs and rules may change.

When process representations are descriptive, they should not be considered final. Within the scope of this thesis, there is a process model that is evolved into the final model by filling this gap.

Descriptive processes are first in the form of “to be” and then in the form of “as is”. Prescriptive processes contain predefined rules and norms. Enhancement can be achieved by not including the parts that do not comply with the rules in normative processes. On the other hand, in models that evolve from “to be” to “as is” it is the modification of the model to reflect reality, while other quality criteria remain at a reasonable level [40,41].

2.9. General Terminology About Artifact

Event-driven architecture is used to develop highly scalable and high-performance systems [42]. It is a software architecture that deals with the generation and analysis of events. Thanks to its high adaptability, it can be used for all small or large and complex systems. With its ability to provide communication between services, it can be used embedded in another technology when used with microservices. Event-driven architecture can also be used alone and allows events to be processed asynchronously with elements that are less dependent on each other.

When event-driven architecture is mentioned, there are two basic architecture types that can come to mind. While the structure, in which there is a center that takes control at certain points regarding the flow of events, is the mediator architecture; the structure in which

response and dynamism gain importance at the point of high level dynamism and event processing is the broker architecture.

Broker architecture is an architecture that does not have a central structure as in mediator architecture and is considered relatively simple. Broker architecture does not provide central coordination and orchestration.

The term ‘technical standard’ is used to measure whether a method, process or application complies with certain norms or requirements. The term ‘gold standard’ is used in medicine. It indicates the test that is used as a reference among many tests for disease diagnosis. If we consider the term gold standard together with the engineering discipline, we can say that it is the reference for measuring compliance with norms or requirements. In the field of PM, the gold standard for evaluating the quality of algorithms is to use quality metrics. In order to achieve the gold standard, an event log is first needed. Event logs can be produced.

Log generation’s working logic is similar to play-out. With play-out, also called an engine as it deals with the general operating logic, with the help of a control algorithm, an event log is created from the executable moves in the process model. However, generally, in addition to the control-flow perspective, one of the components of the multi-perspective structure is the time perspective.

With event log generation, a play-out process is carried out by considering gate-like elements as workflow objects and values of the data perspective as data objects, along with start and end events and other tasks. The engine used is an engine belonging to the PLG [43] library. Process models are viewed as a dependency graph. An event log is created with the help of a context-free grammar according to the objects that may be encountered in operation.

Kafka technology is used for broker-based architecture. The producer [44] sends messages to the Kafka cluster. While a message can be of basic types such as string or integer, it can also be a special object or a file. A cluster that consists of at least one broker and provides high availability, storage and scaling. However, a cluster consisting of one broker may have problems with high availability. A message broker [44,45] is a server that gets message from a wide variety of systems and sends it to the relevant parts, and can manipulate data in flight. Messages are conveyed on the topic they are related to. The topic allows messages to be grouped according to the category they belong to. A topic can be divided into more than one

partition. Therefore, partitions can be defined as more specific categories. By subscribing to the message broker on a topic, the consumer [44] can receive messages related to the relevant topic.

3. RELATED WORK

In this section, as a result of the related readings about the multi-modal PM, methodologies that are encountered and frequently used in PM are mentioned. Afterwards, the derived research methodology is explained. Finally, the articles divided into groups as a result of the research methodology are explained and a comparison table is presented.

3.1. Frequently Used Methodologies in PM

As in data mining, there are guiding methodologies for planning and executing projects by saving time and cost. Since their names will be frequently encountered in the literature, it was deemed useful to mention them here.

Unlike data mining projects, Process Diagnostic Methodology (PDM) [46], L* Life Cycle Model [47], Process Mining Project Methodology (PM^2) [46] are used in PM that focuses on process discovery, conformity control and process enhancement. On the other hand, PDM, L* and PM^2 do not jointly provide guidance for identifying quality problems.

Except for one or two articles [48], data quality is not taken into consideration. It is aimed to provide a broad perspective with PDM - in the context of generalization. Therefore, it cannot be applied to every project. It contains few PM techniques and the use of domain knowledge in the analysis process is low. Its applicability to very large and complex projects is less common [46].

The L* life cycle model is mostly applied to processes with a complex structure [47]. The first stage is based on a deep understanding of the problem. It is necessary for the other stages and cannot be underestimated. The second stage focuses on data transformation (like filtering). It is important to generate insight on process behavior and determine objectives and questions based on domain experts and the current system. The third stage is the one where PM techniques (process discovery, compliance checking and improvement) are applied. In the fourth stage, insight generation and explanation of the results in the context of the field are carried out. The results may indicate problems for existing processes, and

approaches such as root cause analysis are applied to diagnose these. The final stage results in the improvement of existing processes.

Since PDM and L* methodologies did not support iteration, PM^2 project management methodology was proposed [46]. Moreover, both PDM and L* provide an additional guide to the general challenges for the inexperienced. $PM^2$is designed to support the development and compliance control of processes based on rules and regulations. It uses PM and additional techniques for regular and complex processes. It includes planning, data extraction, mining and analysis, evaluation and process improvement and operational support stages. Analysis begins with predefined questions and is supported by qualified questions. Methodology outputs are performance and compliance findings, problem diagnosis and evaluations, and answers to questions. It can contain iterations in subsections.

3.2. Review Methodology

To express the literature review methodologically, within the scope of the study, searches were made regarding multi-modal PM to scan the approaches in the literature. As a result of the research, not many studies were found with this nomenclature, so related readings were made. With the academic support received, articles using the multi-modal definition with a multi-perspective structure were found. The scope of research was expanded to include multi-modal and multi-perspective studies. Along with related readings on PM, RCA was also included, with the determination that root cause analysis was used for improvement. Subsequently, research was carried out on these three topics, especially in the Google Scholar database. Support was received from existing literature reviews in improvement and root cause analysis. The study was shaped by identifying gaps from the articles found. Unlike other methodologies, the scope was subjected to the divide and conquer approach and then synthesis was made.

3.3. Studies on Multi-perspective PM with/without RCA and Multi-modal PM

3.3.1. Multi-perspective PM Studies

There are two studies that apply multi-perspective PM on blockchain. In the first study, Ekici et al. [17] develop a tool. This tool includes a service, a blockchain network, and a database server. With the service contained in the container, BPMN files enriched with the resource and data perspectives of the multi-perspective structure can be presented to the blockchain.

With the help of the component containing the database, the BPMN model created by each perspective can be converted into an intermediate model. Validation is carried out with the data coming through the intermediate model and enriched BPMN files. The BPMN visualization provided by the tool through the intermediate model is validated for direct use with other tools. Tiftik et al. [49] design a framework that includes similar components. All outputs of the control-flow, resource and data perspectives are included in the intermediate model. Enrichment is provided by combining them through. The output is returned by the service in BPMN format.

Mans et al. [28] make the first use of the multi-perspective structure. Hospital data is used and process discovery is carried out with the control-flow perspective of the multi-perspective structure. From the process discovery, a finding is reached regarding patients referred to the hospital from different hospitals. From an organizational perspective, it is possible to identify the department most relevant to the department being worked on. An examination of the time perspective is carried out, expressed as a performance perspective. The focus is on average waiting time.

Mannhardt et al. [8] highlight the difficulties experienced with multi-perspective PM. As a result of these difficulties, an approach is proposed to enable multi-perspective analysis to be carried out through an explorer. The proposed approach is the one that really makes multi-perspective mining shine. Accordingly, perspective changes are represented by modes under modality, which can be called view modality. The control-flow perspective can be associated with the process mode, the performance mode with the time perspective, and the resource and data perspective with the data mode. Additionally, metrics related to fitness and precision mode can be calculated.

Again, Mannhardt et al. [31] demonstrate and confirm that a multi-perspective analysis approach can be applied to healthcare processes using iterative analysis methodology. They combine the control-flow perspective with the data perspective through time perspective output data to enable the detection of disease-related problems. They provide insight into the process through the defined rule.

3.3.2. Multi-perspective PM Studies with Root Cause Analysis

Aside from the studies mentioned above, PM studies involving root cause analysis can be carried out from a multi-perspective structure. Because root cause analysis complements the improvement purpose of PM. Although PM and data mining or ML are separate disciplines, it does not mean that these techniques cannot be combined to improve results. In the following paragraphs we provide an overview of the studies that combine both.

Erdoğan and Tarhan [7] in their study provide process discovery through model discovery from records, compliance control through control of deviations with models and records, and performance improvement with ML-supported analytics. Multi-perspective PM techniques are applied to the target-oriented project and supported by data-driven. RCA is performed with the support of Goal Question Feature Indicator (GQFI) [50] and learning machines. To elaborate: Three questions are asked about the process. The first question asked is about the operation of the emergency process. Deviations are detected with the help of control-flow and time perspective. These deviations are generally related to the long duration. The suggestions made regarding the long duration are using a call system and performing simultaneous transactions. The suggestions given are based on the actual process. The second question concerns frequently and infrequently used paths. Frequent and sparse paths are extracted according to the relative frequency of a trace within the traces. The event log is enriched with this frequency value. Then, by using multi-perspective data with a decision tree, it is determined that infrequent events and elderly patients cause delays. The suggestions for improvement based on the findings include unit-based classification in the emergency department and increasing the triage scale to five colors. The third question is which traces deviate from the modeled behavior of the process. Deviating traces are detected and a control is added to the system to guarantee entry into the triage process.

Suriadi et al. [47] examine the reason why possible simple applications take more time than expected using the L* life cycle model. The insurance claim process categories of the insurance compensation provider are extracted with the attributes of the data perspective, as simple and complex. Both have subcategories as fast and slow. In the analysis phase, performance distributions and process views for simple and non-simple are obtained through the control-flow perspective. Event-based operating frequency and event log distribution metric are discussed. The root cause is estimated based on the increase in the difference between the metrics. This result of the interpretation phase is verified based on event

duration at the same stage. During the improvement phase, with these findings, the activities that cause the problem are marked as damaging points and improvement is provided.

Goel et al. [51] examine the educational processes of a higher education institution in Australia. The study is built on six determined research questions. The first question is about data quality. Data cleaning and pre-processing operations are carried out in line with the first question. Filtering is done on students who complete on time and those who have a decline. The second question is in line with the exploration of the process through different behaviors, and the situations that cause departure from the program and delay in the training period are determined by taking into account the milestones. With the third question, the examination ends with the deviations between the real and actual processes that occur due to the wrong progress of the process. The investigations up to this point concern control-flow and time perspectives. Regarding the data perspective, from the fourth question, it is determined that factors such as faculty and gender did not affect the completion time. With the fifth question, the root causes are analyzed by comparing the students who experienced a decline with those who completed the program, based on indicators between opposite groups. With the sixth and last question, improvement is suggested with the findings.

By applying the PM² methodology, Eck et al. [46] start with an abstract process view question and focus on IBM in-house purchasing processes by taking data from the SAP system. It is aimed to overlap the time, cost and resource perspectives of PM with the time, cost, resource and quality perspectives defined from the BPMN model. Analysis is supported with qualified questions, and through iterations, deviations and factors that disrupt performance are identified and process improvement is attempted by overcoming the root causes of the points that hinder improvement.

Aguirre et al. [52] use values of time, control-flow and data perspectives. Performance indicators are determined with elements such as customer complaints, cost and time, based on the known definition of the process. After these processes, approaches such as process discovery, conformity control, and detection of deviations are applied. Problems that do not meet the performance indicators are identified and estimated with the help of fishbone and simulation model. The predictions make are advanced through data mining. From here, the required process model is determined and improvements are done.

Lagraa and State [53] use limited other data types for behavior data and organizational, and time and data perspectives, only with a focus on process discovery. With the work done through network analysis, malicious entries are first filtered on a user basis. For activities between successful logins, attributes and attribute names that vary between activities have been extracted. These extracted values are used for process discovery, and malicious nodes are detected. RCA is performed with the obtained process map [53].

Meincheim et al. [54], use the control-flow and time perspectives of the multi perspective structure. With the study carried out, the complex process is subjected to the incremental trace clustering approach. In this way, process variants are clarified. As it becomes easier to monitor certain process performance indicators, the detection of the production line that causes them is carried out by playback. The root cause of the disrupted node in the production line is determined by the difference in queue size compared to the status of other nodes. Verification is carried out with stakeholders. They offer improvement suggestions for better resource allocation.

3.3.3. Multi-modal PM Studies

Two articles focus on new/different behaviors in multi-modal fashion. The first deals with text data while the second focuses on image data. Rullo et al. [9] develop a framework using multi-modal PM. With the study, the traces in the record are embedded in attribute and behavior-based vectors, and trained by ML algorithms with known behaviors. The new-different behavior is detected on the newly arrived traces and a score is produced. Afterwards, the percentages of the occurrences of the features with respect to the feature-based analysis are examined for top scored behaviors, while in the regional analysis, the results of the examination are given in a certain range for a specific column. The comparative results of the applied ML algorithms are processed [9]. In another study by Rebman et al. [10], manual processes with data from IoT devices is verified, and disambiguous behaviors are detected. Multi-modal motion analysis is performed by taking the image data as input, the process is discovered from the detected motions, and the conformance checking is provided with a predefined business model [10].

Multi-modal structure is also thought with deep learning architectures. In study [55], with the help of deep learning algorithms, it is aimed to make process estimation by giving the traces of the event log as input to the network. Traces in a multi-modal structure and other

components of the event are also handled by deep networks. The results of the study are not very successful due to the skewness of the trace length and it is stated that more progress should be made. Another study [56] investigates deep learning methods for predicting the future state of ongoing processes It is pointed out in the study that there are perspective data that are hidden in context outside of time, source, and control-flow and these are also studied.

In the learning domain, with video data and electrodermal signals, physiological measurements are taken after learning and regulating activities with the structure called multi-modal [57]. The effect of regulating activities on student learning is studied. The triggering of physiological impulses by individual and group arrangements after cognitive, emotional and task operation is examined with PM.

In another domain, for the construction industry processes, multi-modal process data are identified [58]. In line with very little information, the operation processes in the construction industry cannot be followed correctly and project estimates are made incorrectly. In order to create realistic simulations, work on multi-modal data affecting truck activity in the construction fleet process is carried out. Data is taken by the sensors and the correct modeling is performed by their fusion.

In addition to the studies summarized above, and as shown in Table 3.1 below, this study includes the improvement and extension parts of process enhancement together, in terms of providing the enhancement by expanding the context. Model related enhancement is provided by using four quality metrics after the conformance checking stage, not in the discovery stage. A decision mechanism can be created with a decision tree whose tree branches are activities. In this case, results can be obtained with attribute values that do not fully reflect the context of the activity. We prevent the loss of context of activities with our multi-modal analysis. The study is primarily designed for descriptive PM, which is not overworked. With the rules to be defined, the declarative process can be easily used for mining. Also, it will be possible to use it with a different analysis nature for prescriptive processes. Multi-modal distributed support is another key difference for this multi-modal PM analysis.

Table 3.1. Summary and comparison with related studies

Study	**Multi-perspective or Multi-modal**	**Conformance Checking**	**RCA**	**Process Enhancement**
[28]	Multi-perspective	No	No	No
[8]	Both	Yes	No	Process model extension
[31]	Both	Yes	No	Process model extension and recommendation on actual process
[7,47,51]	Multi-perspective	Yes	Yes	Suggestion on actual process
[52]	Multi-perspective	Yes	Yes	Redesigned BPMN model
[46]	Multi-perspective	Yes	Yes	Modification on actual process
[54]	Multi-perspective	Yes	Yes	Suggestion on actual process
[53]	Multi-perspective	No	Yes	Process representation without malicious node
[10]	Multi-modal	Yes	No	No
[55,56,57,58]	Multi-modal	No	No	No
This study	Multi-modal	Yes	Yes	Enriched process model enhancement

4. FRAMEWORK

4.1. The Need for Designing a Loosely Coupled Process Mining System

In order to start an analysis in PM, the existence of an event log representing the process is essential. Although direct interaction with stakeholders can be achieved to obtain these event logs, inter-system transmission over a network environment is also possible. Although file sharing may come to mind because event logs are files, the importance of designing a scalable system in real world where change is inevitable should also be taken into account.

In line with this need, an integration solution is needed to provide data transmission between two different systems. Integration solutions should include loose coupling between the systems they integrate, as integrated systems also require adaptation to the nature of change [59].

By loose coupling, we refer to an approach to connect the elements that make up the system. This part can be a system, software system or a network. It may also be preferable for the connected elements to have minimal knowledge about each other and their communication requirements. It is aimed to reduce the impact of a change made in one element on other elements. By minimizing interconnections, it can also facilitate error solving, testing, and maintenance by directly monitoring the input and output of the elements.

If tightly coupled systems are considered in terms of output, the system works together with other elements to create an output. This type of close interdependence will increase the likelihood that changes in one element will affect other elements. Since the goal is to maintain overall integrity, if a change in one element requires more changes in other elements, it will require more effort and resources. As a result, it is possible to talk about increasing costs.

The degree of dependence can be measured in order to express a system as loosely or tightly coupled. The number of changes that may occur in other elements after a change in one element of the system can be used as an output of this measurement. With this use, the measure of changes in other elements by adding, removing or changing an element in the system will allow determining tight or loose coupling. Elements in a loosely coupled system are less dependent on other elements and are less affected by operations related to other

modules. As a result, it can be said that it provides greater interoperability and is more scalable.

It was stated that other integration solutions, such as file sharing, were not preferred in terms of scaling. Another solution that might come to mind would be using remote procedure call.

Local procedure calls can be given as an example for tight coupling. The calling procedure may need to pass too many expected parameters. The called method will start processing as soon as it makes the call, and the calling method will not be able to continue on its way until the called method has finished its work. This will mean that the call is synchronized, and it can be said that synchronization increases dependency by causing a wait.

In order to ensure scalability, the aim was to design a loosely coupled system. Today, many organizations prefer loosely coupled systems unless there is a compelling reason. Many applications use request-based architecture. In this approach, requests are sent to the element called the orchestrator, and the request orchestrator carries out the operations deterministically and synchronously. As a concrete example, operations can be performed with many graphical elements within the container through an application that provides a graphical interface. Here, different elements can be used for different purposes and they respond to a request made by the user through interactive interaction. For each graphical element, the services provided for the element in the relevant class can be processed with the anonymous inner class technique, and the set of anonymous operations provided for the relevant element will provide the implementation of a logical interface. The class that provides these logical operations can be considered an orchestrator. Thus, each logical interface will provide a service to the class related to the implementation it provides. Here, in case the results of more than one service affect each other, the retained service response can be used and the user can be enriched for the next response and returned. It will be seen that event handling is done through an element and there is a request response approach.

In order for a chain of events to occur, the first event must occur, there must be guidance regarding which processing unit the events will be transferred to, there must be processing units that receive and process the directed events, and at least one event must occur. If an event has occurred, we can talk about the existence of a processed event. The unit that receives and sends domain-specific data federatedly is the broker. The processed events are pressed to the relevant federated channel set and served to the relevant event listeners.

Mediator architecture draws the architecture into a form similar to the request response architecture with the centralization of flow control points. Broker architecture, on the other hand, can produce distributed-centered solutions for flow control-oriented architectures. Let us think about an order transaction scenario that we frequently do in daily life. For this scenario, broker architecture will be considered first. The scenario will include operations such as creating the order, receiving the payment, informing if the payment is unsuccessful, packaging and sending it if successful, and deducting the relevant order from the stock. When this scenario is wanted to be implemented with a broker-based architecture, three event handlers will come to mind initially: Order taking element, payment and inventory elements. The order receiving element will send an event to its listeners, the payment and stock elements, via the order receipt channel. If the order receipt information is transmitted to the orderer, information will be provided to the user with a notification-related element, and at this point, the notification element will also listen to the order receipt element. The payment element will process the event to the order completion element via the payment receipt channel. The next stop of the order fulfillment element will be the delivery element. The channel between the delivery and order completion elements is the order completion channel. The notification-related element also subscribes to the order complete and delivery complete channels. What is to be said about the payment part of the flow is that the failure of the payment is transferred to the notification service through the payment failure channel. The stock element that receives the event from the service receiving the order at the same time will deduct the order from the stock, allowing it to communicate with another element. The problem that needs to be kept in mind here is that the deduction from the stock and the failure of the payment transaction occur simultaneously.

To solve this problem, mediator architecture will be discussed first. Subsequently, the solution will be expressed as the reader will remember.

When the solution is made with the mediator architecture, an order-related demand will come to the mediator element, which controls certain points regarding the flow. The mediator, which transmits the request to the order creation element via the order creation channel, will simultaneously send the transaction to the payment and stock drop services with the confirmation response of this element. The solution appears here. If there is no problem with payment and stock, the mediator who receives positive approval will continue on its way. If there is a problem, first of all with the payment, the transaction will be

transferred to the notification unit through the notification channel, together with the negative confirmation. There has not been a deduction from the stock yet and payment is made for the product in stock. If the positive situation continues, the order preparation instruction will be sent to the relevant element through the relevant channels. The mediator element, which receives confirmation of the order preparation and destocking process, will be able to send its instructions to the product delivery unit through the channel it speaks to. There may also be a notification here, such as the order is ready. The mediator element, which receives control again positively, will be able to perform the necessary action regarding the notification.

The solution to the transaction made using the broker is to send a positive stock update event to the stock update unit simultaneously with the notification failure element when the payment fails.

In the mediator architecture, there are commands for the desired steps to occur as well as the orchestration of the flow. In the broker architecture, the event handlers react after the events are broadcast to the relevant units. It was stated that a design idea was started to ensure the transfer of event records in PM. Therefore, a broker-based architecture, which is a loose-coupled solution that also allows for asynchronous structure, will be used to transfer an event record or listen to event perspective data from multiple channels.

4.2. Research Methodology

The research method used in this study was inspired from a guideline by Hevner [12], the principles of which is presented be low.

G1. Design Science Research (DSR) must produce one or more viable artifact in the form of a construct, a model or a method.

G2. Providing a suitable technology based solution to the problem according to the purpose of DSR.

G3. Verification of a design product rigorously proven through well-conducted evaluation methods.

G4. Research contributions: Whatever is designed (artifacts, methodologies, etc.), effective DSR provides verifiable and clean contributions.

G5. Research rigor: The artifacts should be defined through formal definitions, pseudo code, source code or robust technological parts in order to ensure reproducibility.

G6. An effective artifact comes to the final stage by making use of what is available.

G7. DSR should be presented to a technology and management oriented audience.

The artifact was created as described in the following sentences, in accordance with the specified steps. The artifact was designed as a result of the studies before the proposal (G1). The technology selection was not completed when the artifact was designed. The technology selection was completed during the literature review (G2). During the literature review, it was confirmed that the artifact was relevant and suitable (G2). Each element was verified with relatively little data and a case study (G3). Thus, it was clarified that it was an artifact that would contribute (G4). A rigor research was carried out with the presence of mathematically based metrics and approaches (G5). A final situation that would produce solutions to the problems was reached (G6). A presentation was made at the conference and defense exam (G7).

Following this guideline, the framework was designed as artifacts of artifact. There are more than one parts in the solution. The framework includes four parts, which are log generation, producer, processor, and broker, as shown in Figure 4.1. In the following paragraphs, these parts and the relationships between them are explained in detail.

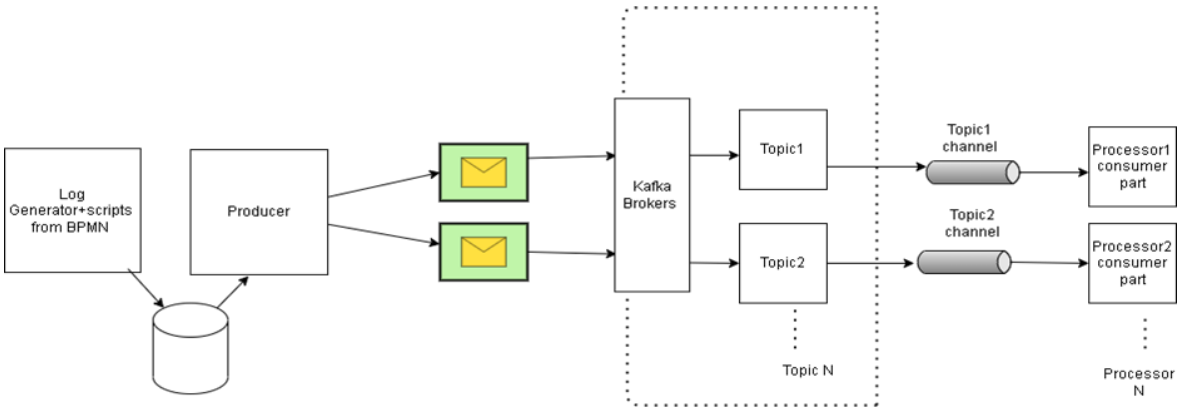

Figure 4.1. Proposed framework - an overview

For the data generation, a BPMN model is handled. Event records are created according to this model.

In order to provide a valid technology based solution, academic studies on multi-perspective structure are primarily examined. Academic studies on RCA are also examined, since RCA will be done on the proposed framework over metrics. And finally, a review of articles is carried out working on multi-modal structure definition.

In order to evaluate the robustness of the designed structure, procedures are carried out at every stage. Data generation is verified by reverse engineering the BPMN representation of the multi-perspective structure and/or checking property and behavior combination with discovery. The multi-modal structure is validated by metrics including RCA.

Broker based architecture is used to provide a structure between the producer and the processor. With the broker based architecture as a contribution, software architecture that can perform process analysis in accordance with future data modes (in the context of image, sound or process) is provided. In addition to this, subunits can be integrated in case of expansion of the system. Due to the decoupling principle, it can be thought of as a processor and producer that fits the subject rather than being dependent on each other. The main contribution is the ability to analyze the structure, which evolves into a very fashionable way, by adding data and resource components to the control-flow.

Each artifact is formally designed to justify all design decisions and include pseudocode, source code, or technological parts. The problems encountered are solved and the results are shared with the academic community through a conference paper [60].

4.3. Log Generator

It has already been mentioned that the starting point of analysis is the existence of an event log. Since these event logs were previously difficult to obtain, even the evaluation of the algorithms was difficult. The main reason for the difficulty is that companies do not want to share their event logs due to privacy policy. For these reasons, firstly, event log production focusing on the control-flow perspective was carried out, and then, event log production including other perspective data was carried out. Within the scope of this thesis,

contributions were made by combining other perspective data in the control plane and providing multi-modal situations.

The algorithm is based on the logic of adding the activity it encounters, from the starting task to the terminating task, to the activity list. In case of encountering an XOR gate, it can proceed with a random selection if no data limiting the selection is used. In case of an AND gate, through the gate it can decide according to the number of outgoing edges. Besides the control-flow perspective, other perspectives can also be included in the production. It allows adding the data of the data perspective as necessary and produced. Data can be produced in two different ways: static and dynamic. In static data generation, a fixed value is assigned to the task to which the data is connected in each operation. In dynamic data generation, different values can be produced for each operation for the task to which the data is connected. Resource data is handled with a data perspective-like approach. It allows setting the duration of the activity and the time required for the other activity to start in relation to time perspective.

There is another important contribution made within the scope of this thesis. While flow diverges from an XOR gate, it may not directly connect to other activities. There may also be another gateway in between. In such a situation, which of the activities following an activity will be selected can be determined with the help of a separate control algorithm.

Since the purpose is to create an event log, production-based definitions of the created event log elements can also be made. Accordingly, an incident occurs as a result of the execution of the business process with the help of a control algorithm. Different cases can be created in each operation. The sequence of activities in the case creates the trace.

4.4. Producer

A producer object is created that contains the value of the message to the topic to which the record is intended to be sent. This producer object contains the topic to which the message relates and optionally partition and the key values. The first thing that needs to be done before the record is sent is serialization. Serialization is when data takes the form of bytes so that it can travel across the network or be written to disk. Partition is one of the sections in which a topic can be located. The key value can help determine which section the message should be placed in, with the help of a hash-like algorithm to be applied. If the partition is

not specifically specified, the message specified in the partitioner section can be transferred to a buffer belonging to the partitioner.

The term fire and forget [42,44] can also be used when the message broker guarantees that the message will be delivered to the recipient after the producer transmits the message to the message broker. The fire and forget expression used in this section relates to the approach that does not concern whether the message was sent successfully or not after the messages are sent from the producer to the broker. In most cases, messages can be sent successfully. However, in this case, some messages may be lost and the lost messages may not be addressed.

With the synchronous approach, a confirmation containing metadata will be received for the message successfully sent to the broker. Metadata includes the topic, partition and offset information of the record. If an error occurrs while writing the message to the broker, the error message is returned and a retry can be performed.

By handling only erroneous cases using an asynchronous approach, message transmission loss can be avoided.

Configurable notification type may be used depending on the approach used. Notification should be used when reliable delivery is important. When high efficiency is desired, reliable delivery may be disregarded and not used in the notification.

The space capacity that buffers the message before it is sent to the network can also be configured. Additionally, network usage and storage space can be reduced by enabling compression. Another important parameter is the configuration of the number of messages that can be written without receiving a response when writing collectively buffered messages to the message broker. A high value can increase efficiency. However, in cases where the message order is important, the transmission is made after a confirmation is received from the target batch in order to guarantee that the message order is preserved. If this configurable value is greater than one, it will be successful in writing the second group of data after a problem in the first group of data transmitted. When the first batch is tried again and the success is achieved, it should be seen that there will be a problem with the order.

Resending a failed message carries the risk of typing the message twice. Because the message is written to the messaging system, but an error may have occurred in the network

during the notification return. The producer will want to rewrite the message into the messaging system. What the messaging system will do here is to not handle the same message twice. Due to the principle of idempotance in distributed systems, even if the same message is sent more than once, it should not change the result.

4.5. Broker

It is preferred due to its uninterrupted operation and data storage feature. Thanks to the producer units and the consumers that they serve, it can be seen as a bridge between the producer and the consumer.

Stores events as a sequential log of messages. The abstraction behind it is to simply append to a partitioned log, allowing streams to be treated sequentially, like hardware-like byte processing. These logged records can be stored for a certain period of time. The importance of storing messages is that a consumer who is slower than other consumers or who cannot adapt to traffic for another reason can receive messages again. Another plus can be noted as the fact that these messages, which are stored and increase in capacity as a result of a disruption in a service, do not slow down the entire infrastructure. Keeping data for a certain period of time can be seen as a negative as it causes storage space. Messages are stored with certain key values. Deletion is as simple as writing a null value to a specific key value. Older messages with the same key value are first marked and then deleted.

By using more than one broker, a backup of a section can be kept in another broker. At this point, a definition of the log can be given as a copy of the partition kept on a different machine. Partitions can be important for a broker to keep alive even if there is no backup. If we consider a scenario where a topic is divided into three parts on two different brokers, even if one broker goes down, a consumer will still be able to receive messages from a different part of that topic. This may be important to show and use the flow continuity regarding this issue. More importantly, it enables solutions that provide high availability by using a redundant structure. By keeping a copy of a partition in other brokers, in case of any problem in the broker, the continuity of the data flow can be ensured with the help of the broker containing the copy. It is important to see that the level evolves from topic-based continuity to partition-based continuity. If crashes occurring simultaneously in more than one agent are to be tolerated, the cluster [45] can be configured with a higher replication factor. Another important point to notice here is that when the cluster is online, many topics

can be added thanks to flexible scalability. Processing any amount of data becomes easier. An expansion is performed and is achieved without affecting the entire system.

How great the scalability of the system will be can be seen by the existence of more than one cluster. Clusters can provide isolated use for security purposes. In disaster recovery scenarios, it can also be used for transfer from one location to another if there is more than one data center.

Another important point in preventing data loss is that the broker performs continuous trials, as mentioned in the producer section. Thus, in case of a failure other than a timeout, sending messages to the broker owned by the cluster will be retried. Messages are sent in batches, and when they are sent to another batch without receiving confirmation that they were written for one batch, there may be problems such as not being written in the correct order. Therefore, in cases where order is important, attention should be paid to the number of messages sent without the approval of the broker.

4.6. Processor

Processor is a combination of consumer and analysis parts as shown in Figure 4.2**.**

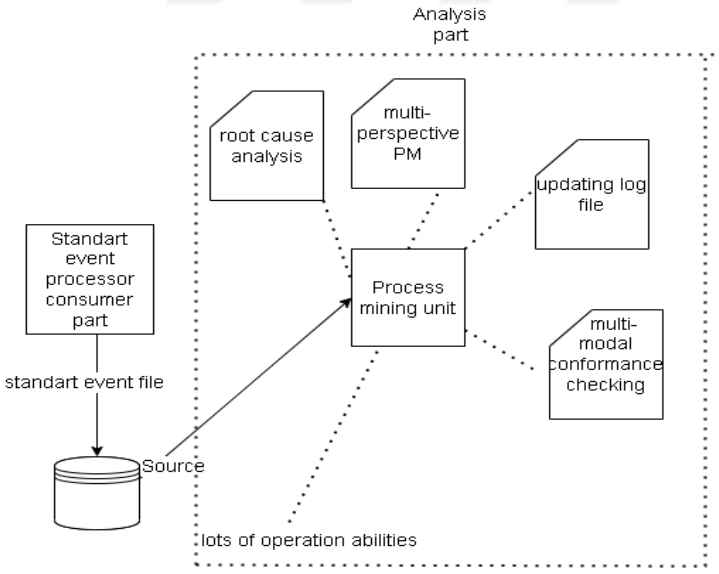

Figure 4.2. Processor subsytem

4.6.1. Processor Consumer Part

It is possible for a consumer to subscribe to multiple topics with a multi-threaded programming approach. When a topic is subdivided, a consumer group consists of a set of

consumers with different interests for each subsection. If there is a single consumer in the consumer group, that consumer will receive the message from all partitions. If there are fewer consumers in the consumer group than the relevant partitions, these partitions can be assigned to consumers by keeping the balance in mind with the help of a partition assignment algorithm.

Adding load-sharing consumers can be seen as important in terms of scaling. Dividing the topics into sections can also be used to ensure load balancing in the future. Having more consumers than partitions will result in idle consumers.

The consumer receives messages from the message broker at certain intervals over the offset it has committed to. While receiving these messages and sending offset information about the messages, a signal is sent to the message broker. These signals are also called heartbeat and mean that the consumer is well and alive.

The consumer's ability to receive messages from the message broker must occur at a certain frequency. Otherwise, problems such as rebalancing and timeout may occur. For this reason, the configurable message size parameter may be important for consumers who do not have sufficiently powerful hardware. Even for computers with already powerful hardware, the maximum message size cannot exceed a certain value. The solution that should come to mind here is to provide a compression-related configuration specification on the consumer side for a record that is subjected to compression on the producer side. The architecture was designed this way. Observing these constraints is essential to prevent rebalancing, as mentioned before.

A very important control parameter is whether offset information is automatically sent to the message broker. If automatic submission is enabled, it will relieve the developer from the burden of checking. In this case, if an error does not occur with the help of a different thread than the main thread, offset information will be sent after a certain time interval. If errors occur, records will be re-pulled from the message broker.

If the offset sending process belongs to the programmer, offset information should be sent after processing all of the records taken from the message broker. Otherwise, there is a risk of encountering a missed message.

4.6.2. Processor Analysis Part

The computer itself also consists of reusable components. This approach to building systems using reusable components can be seen as the obvious goal of engineering.

While the framework provides a frequently used solution to a specific problem in a field, it produces a reusable solution.

The use of a single framework, based on software or through software development, is often an instantiation of the framework. By using more than one framework, introducing a new framework to suit the nature of the requirements or intent can be achieved.

Within the scope of this thesis, messages can be received for the subscribed topic for the consumer part of the processor, with a structure that complies with the Hollywood principle [61]. In accordance with the philosophy of this principle; the three frameworks, pm4py [62], ProM [62], and Cobefra [1] are used for PM in the analysis unit, are not requested from the consumer part, and are not kept dependent on this component by ensuring that the event record transferred to the disk is used as input.

What the analysis unit needs to do is to enrich the event record it receives as input with data from the resource or data perspective. Although ProM provides a plugin for this purpose, it offers a more customizable algorithmic approach in comparison with the pm4py framework. Other operations to be performed are process discovery, conformance control and improvement. Both frameworks offer solutions with similar logic for all intended operations.

For this purpose, the analysis part is considered as a framework unit for the processor part, where processes for PM are carried out. It is inspired by the single responsibility principle in software development. According to this principle, the level of responsibility of classes is reduced. A class designed for more than one purpose should undergo minimal change when the purposes affected by the requirements change. Otherwise, a class that changes too much depending on requirements will slow down the speed of adaptation to change. However, the difference here is the use of much more comprehensive software. When using such comprehensive frameworks, the sole responsibility is to provide a framework that can perform operations appropriate to the purpose of PM. The responsibility taken by frameworks is specific to the field of PM.

By using this structure, multi-perspective event logs are taken as input. These event logs are enriched with data from the resource or data perspective. Event logs that evolve from a multi-perspective structure to a multi-modal structure are again candidates for entry as an event log. With these multi-modal logs, process discovery is performed for control-flow and resource mode and control-flow and data modes. The new event log is ready for compliance checking after enrichment. The compatibility check to be performed here is on multi-modal event data and multi-modal process model. After the compliance check, four metric values are calculated with two different approaches. RCA is performed based on the calculated metric values. With the RCA, it can be determined which behavior changes with which attribute.

5. CASE STUDY

In this section, in summary, the proposed framework is verified through a process belonging to the telecommunications sector [22]. Accordingly, the fault repair process for the phones produced by the telephone company is included. There are completed and not completed cases of the process. After the initial examination, information is obtained about the types and error types of the phones subject to the process as well as the operation and roles of the process. In order to detect the new behavior, which is our aim, the selection process for the algorithm to be used is carried out. Afterwards, a new event log parallel to the process is generated with the help of the event generator. Following the verification that the data generated through the selected event log is produced as desired, it is transferred from the producer to the processor part. In the processor, the old state is stored for the event log, the verification of which is provided in the producer part. A few new event logs containing minor changes are transferred to the processing part, and a new record is selected among them and the detection of the new behavior is determined by RCA based on metrics. After detection, results are provided by demonstrating the improvement. There is a process for a case study as shown in Figure 5.1 for repeating purpose.

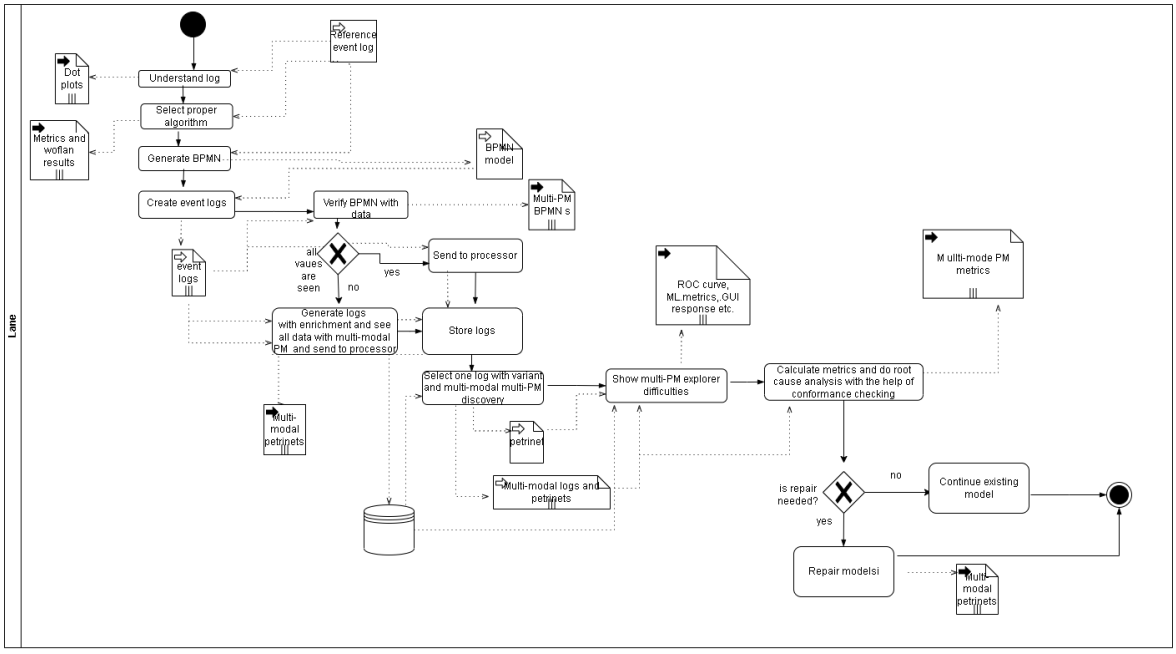

Figure 5.1 Case study steps that have been followed in this study

5.1. Algorithm Selection

The repairExample.xes event log will be used during the implementation phase. Under normal circumstances, with the proposed framework, a process model can be developed by drawing a BPMN model without any event logs. Based on the developed model, the production of new event record files can be achieved with comprehensive knowledge of the process. Within the scope of this thesis, the event log representation of the process will be used to obtain information about the process.

Information about petri nets was provided in the background section. In this section, these concepts will be briefly mentioned along with the analysis structure, since the output of the algorithms is a petri net and an analysis is made through workflow networks (WF-net), which are a special case of petri nets.

In a workflow network, there is a starting node and an ending node. All nodes between these start and end nodes must be on a path from start to finish [63].

Algorithm accuracy over a WF-net can be achieved through three basic features. The important point here is to evaluate the algorithm in terms of whether the output model produced by the algorithm is correct. With a correct model defined on WF-net, the end point should be reached without any errors or restrictions. The three basic features used when evaluating accuracy are option to complete, proper completion and absence of dead parts [63].

Option to complete refers to reaching the final marking through all possible paths in the process representation. Proper completion means that there will be no token left on the network when the process representation reaches the end point. Absence of dead task is that each transition point in the process model can be activated.

A preliminary review was carried out for the log file to be used. Accordingly, there are 1104 cases in the event log file. It appears that there are cases that are incomplete. Incomplete cases are not dealt with. Because these cases have not been concluded yet, they may be misleading. As a result of the filtering process, an event record containing 1000 cases is obtained.

Based on our theoretical knowledge through this process representation, it is shown that the correct algorithm is the Inductive Miner algorithm. For this, process discovery is carried out with Alpha Miner, Heuristic Miner and Inductive Miner algorithms. After the process discovery, the process models are analyzed with Woflan [63] and the correct algorithm is selected. In addition to analysis with Woflan, event log-based quality metrics where the process is discovered are also provided only for token based replay. Since the Fuzzy Miner algorithm does not produce a petri net output, it is not included in the comparison.

First, we use the Alpha Miner algorithm. As a result of the Woflan analysis performed with the process discovered with the Alpha Miner algorithm, a feedback is received that the network is not a WF-net. This means that the steps to advance the verification cannot be progressed because a WF-net cannot be created. There is a situation where the places and transitions in the petri net created by the algorithm do not exist on a path from start to finish.

As a result of the analysis performed with Woflan for Heuristic Miner, information is obtained that the network is a WF-net. Although the network is a workflow network, there may be places that are not covered in the subnets that make up the network. Although an area not covered may be possible with a network without free choice, a workflow network with free choice is preferred. As the analysis progresses, the symbol remains in some places when the final markup is reached, which does not provide proper completion. Therefore, it is not a sound network.

A summary of the results obtained is seen in the Table 5.1.

Table 5.1. Accuracy and metric values for algorithms

	isSound	Fitness	Precision	Generalization	Simplicity
Alpha	No	1.0	0.19	0.97	0.79
Heuristic	No	0.97	0.62	0.95	0.77
Inductive	Yes	1.0	0.70	0.97	0.74

As a result of the analysis performed with the Inductive Miner algorithm, it is seen that the network is a WF-net and that there is no violation of the option to complete option. In other words, a flow does not occur without the appropriate combination of places leaving the same transition.

In this way, each selection can be made first and then the final marking can be reached. When the final markup for each operation is reached by the network, no token is left behind. All transitions are live, meaning the network is live. It is not possible for tokens for all locations to increase continuously throughout operation. In other words, it also satisfies the network boundedness property. For any transition, there is no situation that will constantly prevent the revival of that transition. So there are no dead tasks.

For the process representation created by the Alpha Mining algorithm, the criteria that can be used to check the process model are not satisfactory. It can be commented that the model is not an accurate model, that it does not fully comply with the event log as a representation of the process in which it was discovered, that the accuracy is low, and that it is not very bad in terms of generalization and simplicity. In summary, it can be said that the process model created using the Alpha algorithm is not suitable for our purpose of representing data.

Under normal circumstances, the Heuristic Mining algorithm can be viewed as a more advanced version of the Alpha algorithm. It takes into account the frequency with which events occur and can also identify cycles. However, it does not guarantee an accurate process model, and an inaccurate model is captured in order to reject the algorithm. If we need to evaluate the process model, it is seen that it is not a correct process model, and although it produces balanced results in terms of metrics, it does not meet our purpose.

With the Inductive Miner algorithm, the accuracy of the process model is guaranteed. When the process model created with the Inductive Mining algorithm is examined, it is seen that the model complies with the criteria that can be used to control the process model; in summary, it can be said that it is an accurate, appropriate, precise, generalizable and simple model. Therefore, in practice, models created with the Inductive Miner algorithm is used and new behavior is detected.

First of all, information about the process is collected. We have an event log that provides a representation of the process. When the event log is examined with the help of the event

summary provided by the ProM tool, the first information that can be obtained about the process is that it contains 8 event classes. These event classes are Test Repair, Analyze Defect, Repair Simple, Repair (Complex), Archive Repair, Register, Inform User and Restart Repair, which can be seen in Figure 5.2.

Class	Occurrences (absolute)	Occurrences (relative)
Test Repair	2738	25,247%
Analyze Defect	2000	18,442%
Repair (Simple)	1394	12,854%
Repair (Complex)	1344	12,393%
Archive Repair	1000	9,221%
Register	1000	9,221%
Inform User	1000	9,221%
Restart Repair	369	3,402%

Figure 5.2. Event classes and their occurences

The reason why these event names in the event log are expressed as an event class is that they contain data related to the resource and data perspective other than the event name. At this point, with the help of a dot plot as shown in Figure 5.3, it is possible to determine which event is realized by which resource by showing the event names of the control-flow perspective on the x-axis and the roles of the resource perspective on the y-axis of an event log that it receives as input.

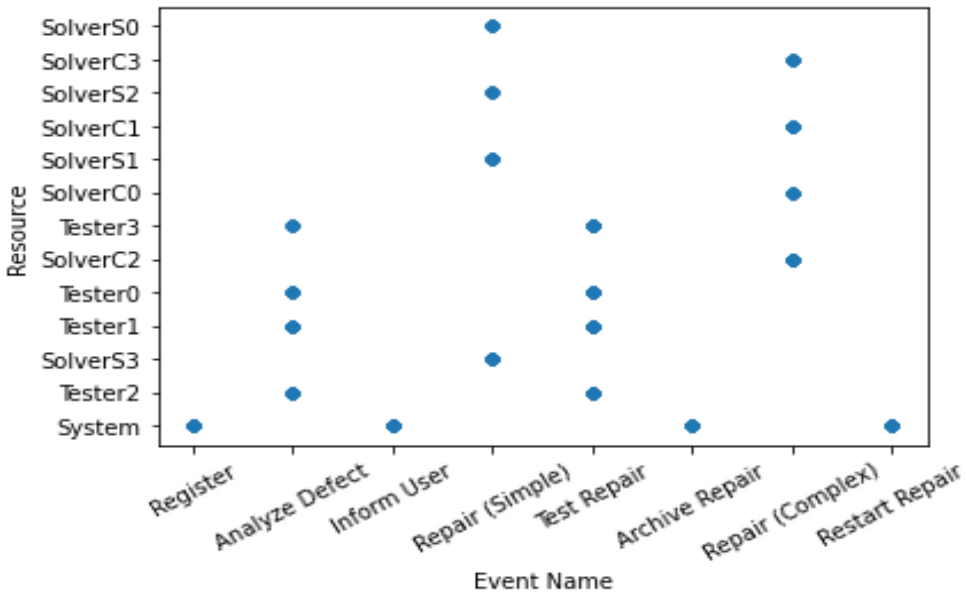

Figure 5.3. Dot plot resource vs event name

Accordingly, Register, Inform User, Archieve Repair and Restart Repair activities are performed by the System role. Analyze Defect and Test Repair activities are carried out by the tester role. There are four testers. Repair Simple and Repair Complex activities are associated with the solver role. There are eight solvers of two types: simple and complex.

When the dot plot is used for event classes and attributes, the attributes related to the Analyze Defect activity class are obtained as phoneType and defectType. DefectFixed and numberRepairs attributes are associated with testRepair and archiveRepair activity classes, and detailed information is included in the graphics in Figure 5.4.

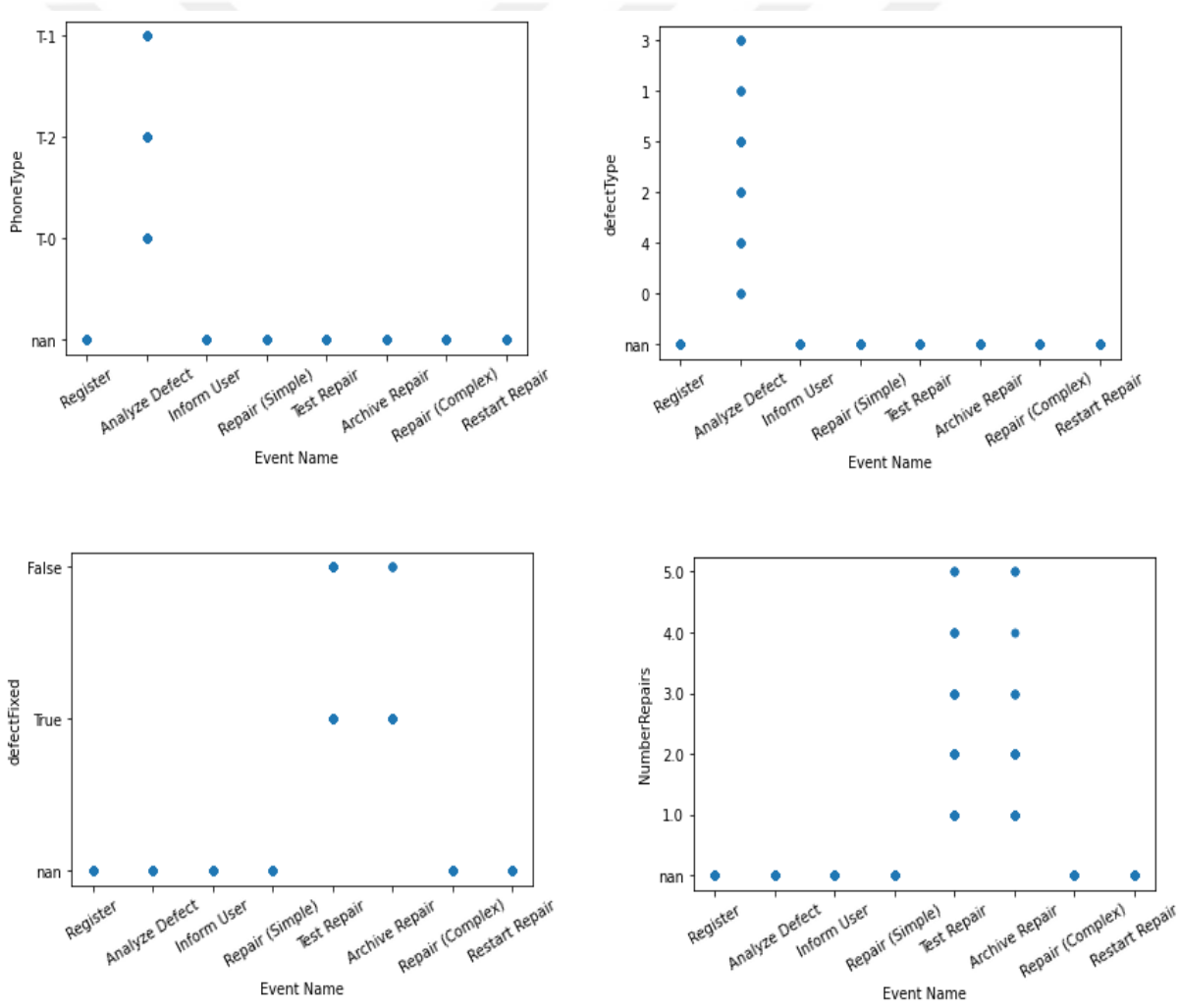

Figure 5.4. Dot plots data vs event name

After the control-flow is understood through process discovery, process information is clarified by in-memory review of the event log representation of the process and dot plots of the perspectives. Failures of phone types are handled after customer registration. After categorizing the faults, the customer is informed and presented in either a simple or complex way. Faults are corrected by fault solvers. The operation performed is verified in accordance with the nature of quality. When an error is resolved, it is added to the archive. If it is not resolved, it is sent to the solution side again, and added to the archive even if it takes a few tries.

5.2. Log Generation

Event record production is carried out based on the current representation of the process we have. Production is carried out with the BPMN representation modeled on the process agreed upon after the preliminary review. It is difficult to simultaneously reflect the operating logic related to the numberRepairs and defectFixed attributes during production. Therefore, after multi-perspective production, which includes the control-flow and time perspective, data from the resource and data perspectives are added to the event log representation of the process. Here, after production with other attributes except numberRepairs and defectFixed attributes, intervention that includes only this logic can also be performed. The BPMN representation of the process is given in Figure 5.5 and a cross-section of its state containing all the operating logic is given in Table 5.2.

Once the process flow logic is captured correctly based on the control-flow perspective, other data is added. The important point to mention is the handling of the numberRepairs and defectFixed attributes. Because it was mentioned that there is no problem in the simultaneous production of other attributes.

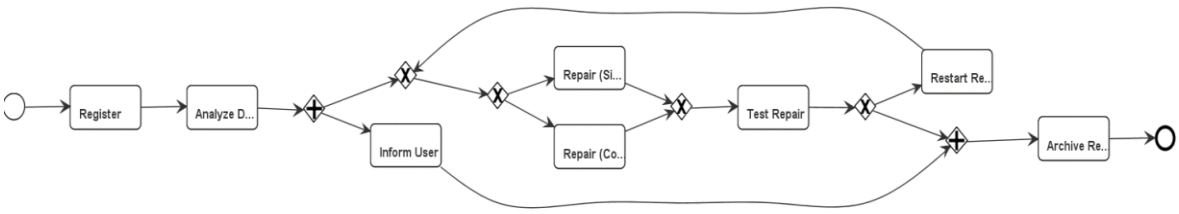

Figure 5.5. BPMN representation of the repairExample process

Table 5.2. Sample event records

Concept: name	**Time: Timestamp**	**Org: resource**	**Case: concept:name**	**phoneType**	**Defect Type**	**Defect Fixed**
Register	1970-01-01 00:00:00+00:00	System	case_55			
Analyze Defect	1970-01-01 01:00:00+00:00	Tester2	case_55	T-0	0	
Inform User	1970-01-01 02:00:00+00:00	System	case_55			
Repair (Simple)	1970-01-01 02:00:00+00:00	SolverS3	case_55			
Test Repair	1970-01-01 03:00:00+00:00	Tester1	case_55			TRUE
Archive Repair	1970-01-01 04:00:00+00:00	System	case_55			TRUE
Register	1970-01-01 00:00:00+00:00	System	case_54			
Analyze Defect	1970-01-01 01:00:00+00:00	Tester0	case_54	T-0	4	
Inform User	1970-01-01 02:00:00+00:00	System	case_54			
Repair (Complex)	1970-01-01 02:00:00+00:00	SolverC2	case_54			
Test Repair	1970-01-01 03:00:00+00:00	Tester1	case_54			TRUE
Archive Repair	1970-01-01 04:00:00+00:00	System	case_54			TRUE

For the numberRepairs and testRepair attributes, the numberRepairs value, which is initially 0, is considered through the in-memory representation of the process. The numberRepairs value is updated with each testRepair operation. The fact that the testRepair activity is followed by the restartRepair activity by selection means that the defectFixed attribute must be assigned a false value. If the archiveRepair activity is seen after the testRepair activity, the defectFixed attribute must be assigned a True value for the repair operation in less than

3 attempts. If not, a random assignment is performed, taking into account that the rapair process may fail after 3 attempts. Finally, the current value of these attributes is written for the activities to which they belong, and the multi-perspective representation of the process is accurately reflected.

5.3. Verification of Log Generation

In order to ensure the verification of the generated data, the first thought was to produce a BPMN output with a multi-perspective structure. In order to carry out this process, a high-level BPMN output was wanted to be provided by enriching the petri net obtained with the control-flow perspective with the help of the event log's data perspective and then adding the organizational perspective data.

Compared to control-flow, there is not much effort involved in decision support. It is stated that the previous study [16] on the decision perspective was Decision Miner. Due to the difficulty of dealing with deviant behavior and complex control-flows with this plug-in, the Decision Tree Miner plug-in [16] was developed. It is thought that with this plugin, accurate discovery can be achieved with incompatible traces according to the data perspective.

For this purpose, we first provide a control-flow discovery with the event log we take as input. When we use the playback result we made with the same event log after the control-flow discovery, the petri net, which is the output of the control-flow, and the event log as input, we see that the data petri net produced as output does not coincide with the logic we deduced as a result of the preliminary examination for the main event log. Even though we know that there is no format error here, we observe that we cannot get correct results when we try to check the format error. As a result of the operations we perform with the event log we produce, which contains less data, a logic that is difficult to read but more accurate can be captured. Figure 5.6 shows a BPMN output with a multi-perspective structure. We underline that problems may occur when the event record size increases. Here, the activity

conditions are separated because they are not read. The conditions are, from left to right, restart repair, archieve repair, repair complex and repair simple.

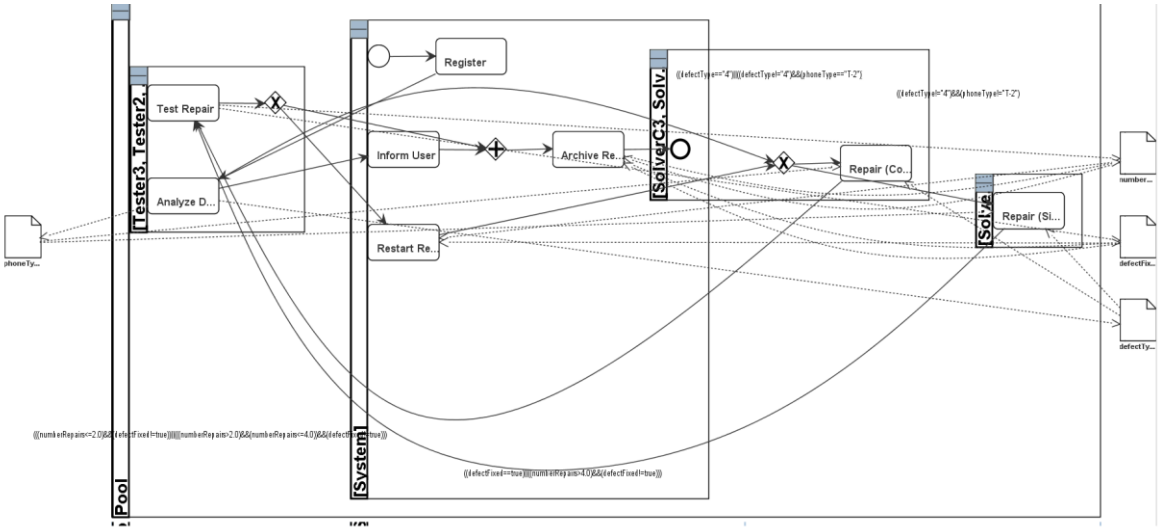

Figure 5.6. Multi-perspective BPMN representation of repairExample process

We also see that the data regarding the organizational perspective is generated correctly. When the relevant pools are enlarged, it can be seen that there are four types of testers for analyzeDefect and testRepair activities, starting from the zero index. The number of activities performed by the system is four and these are register, inform user, archive repair and restart repair activities. As expected, complexRepair and simpleRepair operations are performed by solvers. Both come in four varieties.

With the help of ProM and the Inductive Visual Miner algorithm, we can see the generated data more clearly through multi-modal process discovery using control-flow and data perspective data as shown in Figure 5.7 as filtered to the purpose of figure quality. According to the non filtered one; it is seen that three phone types and five fault types regarding the analyze defect activity provide modularity. We can see that the defectFixed and numberRepair attributes belong to the testRepair and archieveRepair activities. It can also

be read that the process logic is appropriate. Similar logic is seen when organizational perspective data is added to the control-flow perspective.

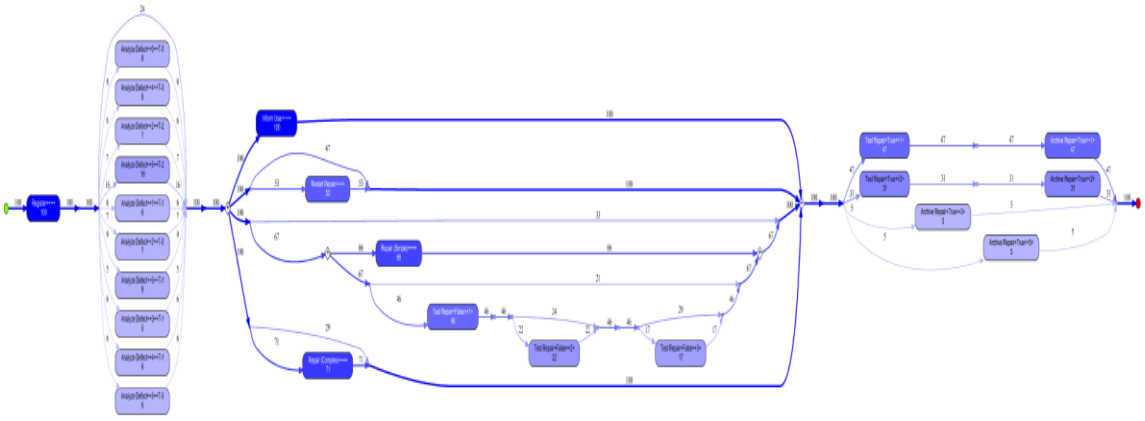

Figure 5.7. Multi-modal proces discovery with control-flow and data perspectives

5.4. Producer

Producer, as an API, is responsible for passing messages to the Kafka broker. Message generation is done in isolation from the Kafka cluster. The address of the message broker is configured by the Kafka client. This address is the connection point to the broker.

Since the flow of messages in the network environment is in serialized format and there are serializers for basic data types, recording is made into a raw byte array in order to send the data in file format.

Since the messages converted to byte array will later be buffered and written to the Kafka cluster, another important parameter is the maximum message size. We are sending files and therefore the maximum message size is set to the largest possible value. Small-sized messages are generally sent with Kafka, and the large size of the message is seen as a problem. We have to configure this value, which the Kafka broker predicts to be 1 MB by default. If we do not perform the configuration and try to send large messages, we will receive an error message from the broker. Another configuration associated with this

configuration is the message size that can be sent at a time. It is important to see the relationship. Because being able to send large amounts does not mean that the broker can carry the message.

Since messages can be sent to the Kafka broker in bulk, we can control how long to wait for other messages to be added, considering that more than one message may be included in one transmission. This control is provided with a value greater than 0.

Even if we use the default message size, we can use compression for this size. Thus, we will be able to send much larger data than the default value. In case the data size to be sent is large, we can use compression for sizes that may be even larger than the data transfer size we have increased to a high size. In this case, we use gzip compression type. Better compression is achieved using this type of compression. It is preferred assuming that the bandwidth is low.

5.5. Processor Consumer Part

As an API, the processor consumer part is responsible for reading messages from the Kafka topic and writing the results to disk.

A consumer object is created and messages are received after subscribing to the appropriate topic. In this case, automatic offset commit is used. Doing this manually has the problem of losing data. There are three types of manual processing, which are ‘exactly once’, ‘at most once’ and ‘at least once’. There is a data loss problem, although not in all of them.

In order to ensure that each message is processed ‘exactly once’, records are captured after the classical consumer object operations. The polled records are processed and the offset information of the processed record is sent to the message broker. If an error occurs during the processing of the record, the records are polled again. With the polling of records, the records following the committed offset information are sent.

With the ‘at most once’ approach, the consumer object is created similar to the ‘exactly once’ approach. After polling, offset information is first transferred for the record to be processed. Afterwards, the record is processed. This process is repeated as many times as the number of records polled. When all the records are completed, the process is performed again. In

case of an error, the record for which offset information is sent will not be re-taken, so this record will be lost.

With the ‘at least once’ approach, the operations are started by creating the consumer object. After polling, the records begin to be processed and are processed in a cycle. After all of them are processed successfully, the offset information of the last processed record is transferred. In case of any problem, since no offset transfer has occurred for the processed records, the records will be re-taken with the not updated offset. This process causes some records to be processed more than once.

The automatic offset transmission approach we use is similar to the ‘at least once’ approach. First of all, the consumer object is created. After the record polling process is performed, a separate thread is started and offset information about the polled records is sent within a certain period of time. If the error does not occur, the polled records are processed and the offset of the last record is sent. In case of an error, the last recorded ones are retaken since no offset information is transferred.

The last point to be noted is that the received records in the form of byte arrays are converted to file format and transferred to disk.

5.6. Broker

Management of brokers is provided by a controller a distributed configuration and synchronization service. The controller server, which provides coordination services between Kafka brokers and consumers, is first set up. Then, a Kafka broker is used to route messages. A topic is being created on the broker.

Since file transfer is performed, configuration is also required on the broker side for large data, as mentioned in the producer section. This regulation regarding the message size is important to take into account the producer's production with this feature. After editing, the maximum data size to be obtained after compression is determined.

5.7. Processor Analysis Part

5.7.1. The Problems of Analysis with the Multi-Perspective Explorer

In the analysis unit, we first try to illustrate why we cannot achieve our goal with multi-perspective analysis with the process representation obtained from the BPMN representation of the process.

Using Multi-perspective Process Explorer, the process of detecting new behavior can be done by ensuring that the rules created for the guard points are 100% correct, and if these rules are not met, this is attributed to the existence of the new behavior.

For this purpose, the process is first discovered with the help of the Mine Petrinet with Inductive Miner plugin. By using the discovered process model and the event log representation of the process as input, a rule can be created for the guard point before restart repair and archive repair activities with the Multi-perspective Explorer plug-in in data discovery mode. Although the event log in which we discovered the process and created the rule is the same, it is observed that the petri net with data representation of the process in data discovery mode is not 100% compatible with the event log. The reason for this is that the rule discovery at guard points is created with the help of decision tree. It will be seen more clearly when the performance indicators of the decision tree are examined.

According to the confusion matrix values in Table 5.3, 89 out of 100 data points are classified correctly for the archiveRepair activity class and 11 predictions are made incorrectly. The true positivity rate for archiveRepair is 0.89, based on the ratio of 89 correct predictions to all archieveRepair predictions. The precision value is found to be 1 by the ratio of the correctly predicted archieveRepair activated class to all archieveRepair predicted values. Although the graphical interface does not return the ROC curve, the ROC area value is close to the ideal value, with 0.945.

Table 5.3. Confusion matrix of decision tree for archieve repair and restart repair activities

	Estimated Value (Archieve Repair)	Estimated Value (Restart Repair)
Real Value (Archieve Repair)	89	11
Real Value (Restart Repair)	0	96

For the Repair Simple and Repair Complex activity classes, the rule created for the guard point before these activities is not converted by the graphical interface. Figure 5.8 shows the petri net with data representation for the two selected guard points. Since the values for the guard point before the repairComplex and repairSimple activity classes are low, an examination is also carried out by coding.

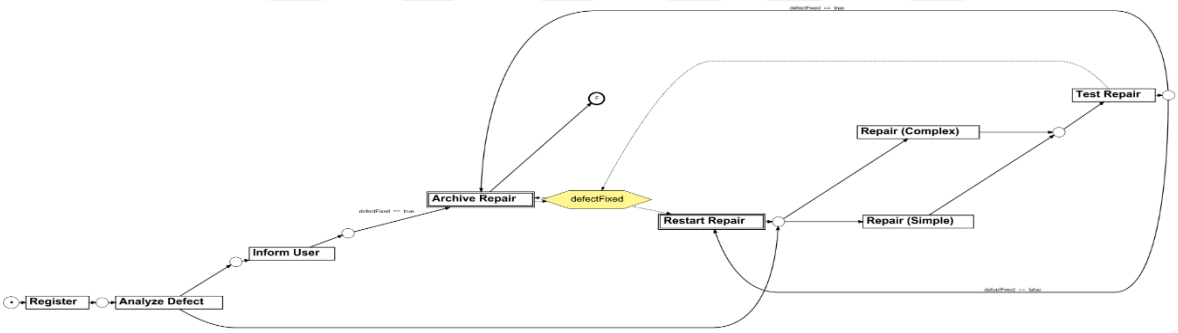

Figure 5.8. Petrinet with data representation for guard points

For this purpose, it is possible to determine which guard point is related to Repair Complex and Repair Simple activity classes through the petri net with data process representation. Afterwards, the decision tree obtained for the relevant guard point is tested with the training and test data obtained for the relevant guard point. Specifically, 90% of the data obtained is used for training and 10% for testing purposes. For each activity class, a long and difficult to read rule is obtained. The resulting confusion matrix is included in Table 5.4.

Table 5.4. Confusion matrix of decision tree for repair simple and repair complex activities

	Estimated Value (Repair Simple)	Estimated Value (Repair Complex)
Real Value (Repair Simple)	1	11
Real Value (Repair Complex)	0	8

According to the complexity matrix values in the table, 1 out of 12 data is classified correctly for the repairSimple activity class and 11 predictions are made incorrectly. The true positivite rate for repairSimple is 0.08, which is the ratio of 1 correct prediction to all repairSimple predictions. The precision value is found to be 1 by the ratio of the correctly predicted repairSimple activation class to all repairSimple predicted values.

For the repairComplex activity class, 8 out of 8 data are classified correctly. The true positivite rate for repairComplex is 1 as the ratio of 8 correct predictions to the predictions made as a whole repairComplex. The precision value is found to be 0.42 by the ratio of the correctly predicted repairComplex activated class to the entire repair Complex predicted values.

The area of the ROC curve obtained is close to the undesirable ROC value. Figure 5.9 shows the ROC curve.

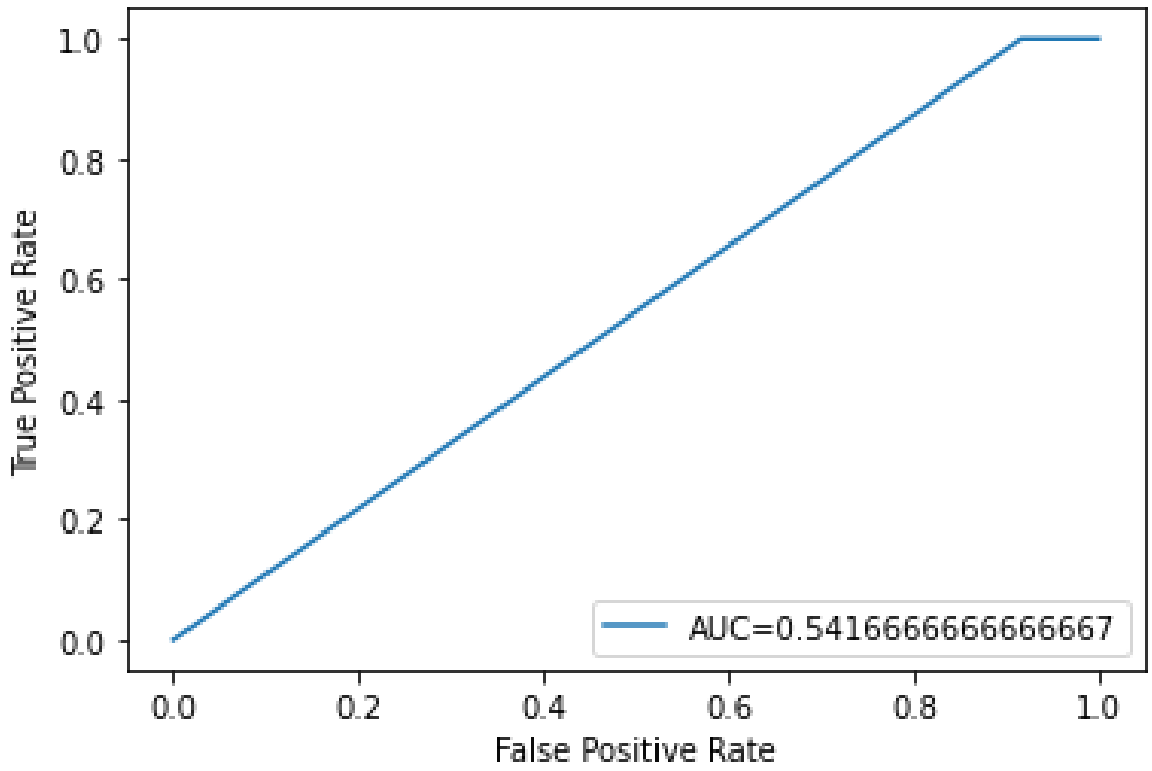

Figure 5.9. ROC curve of decision tree for repair simple and repair complex activities

Secondly, with multi-perspective analysis, we set out with the idea of performing the analysis with the event log created as a result of combining the data of the data perspective with activity classes in a multi-modal structure. Since the placements do not have detailed feedback, we cannot see which activity class turns into a new activity class with which feature.

For this purpose, firstly, the conversion to multi-modal form is provided for the old multi-perspective event log. A Petri net is obtained through process discovery with the help of the multi-modal event log and the Mine Petrinet with Inductive Miner plug-in. When the old event log along with the old petri net is given as input to the multi-perspective explorer, it can be seen that 100% fitness is achieved in one of the modes that provide fitness-related information.

For the same purpose, by giving the new event log along with the old petri net as input to the multi-perspective explorer, it is seen that the fitness value decreases in one of the

performance or data discovery modes that provide fitness-related feedback. However, since no details about the alignment are provided, it cannot be seen which feature caused this situation.

5.7.2. Root Cause Analysis

In order to detect the new behavior, the old event log produced in a multi-perspective structure is first transferred from the producer to the processor consumer part. The event log, which has a multi-perspective structure, is obtained by combining data from the data and resource perspective with data from the control-flow perspective. A section from the event log, which includes the obtained control-flow and data perspectives and which we can call multi-modal, is given in Table 5.5.

A few of the new records produced are forwarded to the consumer part. We carry out our examination with a randomly selected file.

Table 5.5. Sample event records with data extension

concept: name	time:timestamp	org: resource	case: concept: name	phone type	defect type	defect fixed	#of repairs
Register++++	1970-01-01 02:00	System	case_55				
Analyze Defect++0++T-0	1970-01-01 03:00	Tester2	case_55	T-0	0		
Inform User++++	1970-01-01 04:00	System	case_55				
Repair (Simple)++++	1970-01-01 04:00	SolverS3	case_55				
Test Repair+True++1+	1970-01-01 05:00	Tester1	case_55			1	1.0
Archive Repair+True++1+	1970-01-01 06:00	System	case_55			1	1.0
Register++++	1970-01-01 02:00	System	case_54				
Analyze Defect++4++T-0	1970-01-01 03:00	Tester0	case_54	T-0	4		
Inform User++++	1970-01-01 04:00	System	case_54				
Repair (Complex)++++	1970-01-01 04:00	SolverC2	case_54				
Test Repair+True++1+	1970-01-01 05:00	Tester1	case_54			1	1.0
Archive Repair+True++1+	1970-01-01 06:00	System	case_54			1	1.0

As a result of the token based replay process performed with the old event log, which combines control-flow and resource perspectives, a total of 1972 tokens are produced. All generated tokens are consumed and there are no remaining tokens added to the system.

When a detailed examination is performed on the petri net shown in Figure 5.9, a total of 196 tokens are produced and consumed during playback of the event log for the places seen as n7, n8, n9, n10, n11, n12, n13.

It is seen that the number of tokens produced and consumed for the places marked n1, n2, n3, n4, n5 and n6 is 100.

With the information obtained from places with a large number of symbols, it can be concluded that the tester and solver roles communicate with each other as a result of the renewal of the repair.

When the information about case fitness is examined, the tokens produced and consumed for the cases restarted by repair are high for these cases.

The model obtains information that no unwanted activity is found in the event log.

With the new recording in the same petri net, a total of 1972 symbols produced can be replayed without the remaining or added symbols.

For the places shown in the Figure 5.10, a total of 100 symbols are produced and consumed for nodes n1, n2, n3, n4, n5 and n6. For the nodes in the sections containing replays related to repair, 196 symbols are produced and consumed during replay of the event record.

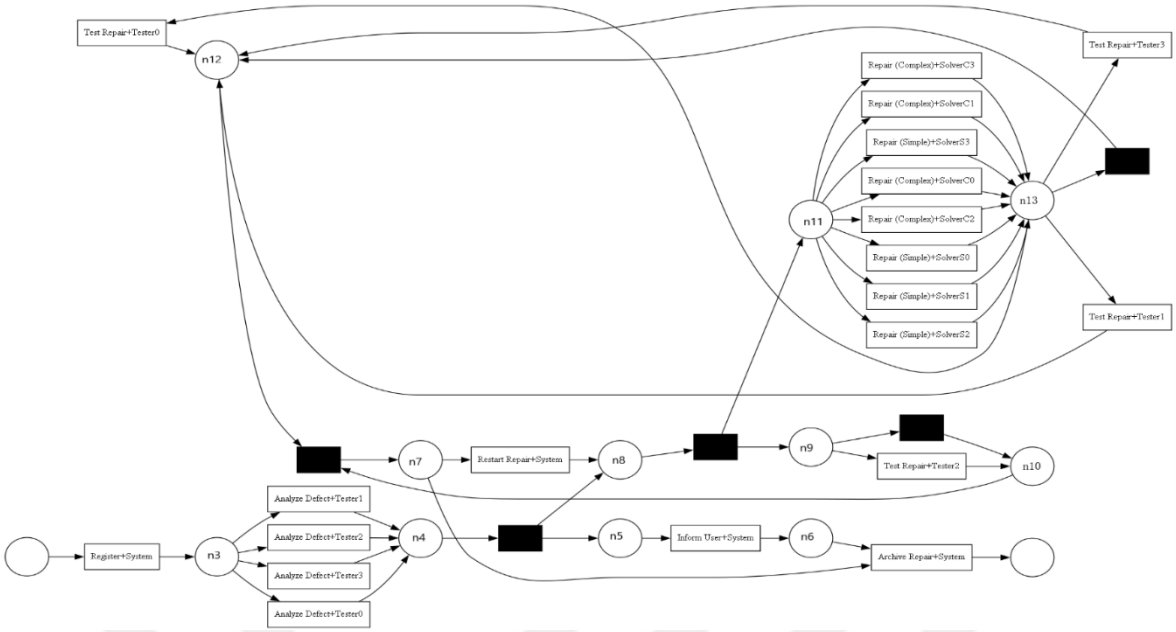

Figure 5.10. Multi-modal petri net with control-flow and resource perspectives

Based on the examination of case fitness values, symbol production and consumption increase due to the renewal of repair-related operations in the newly produced record.

For the new event record, information is obtained that there is no unwanted activity.

With alignment based replay, it is seen that appropriate placements are made for all cases and all placements are realized appropriately. Table 5.6 contains the alignment information made with the new event log in the petri net obtained from the old event log for some cases. In the model, it is seen that silent transition progressions do not reduce fitness on a case-by-case basis. It can be seen that similar results are obtained with the event log in which the model is produced.

Table 5.6. Control-flow and resource mode alignments between old model new log

Case id	**Log Move**	**Model Move**
0	Register+System	Register+System
	Analyze Defect+Tester2	Analyze Defect+Tester2
	>>	None
	>>	None
	>>	None
	Inform User+System	Inform User+System
	Repair (Simple)+SolverS0	Repair (Simple)+SolverS0
	Test Repair+Tester3	Test Repair+Tester3
	>>	None
	Archive Repair+System	Archive Repair+System
1	Register+System	Register+System
	Analyze Defect+Tester0	Analyze Defect+Tester0
	>>	None
	>>	None
	>>	None
	Repair (Simple)+SolverS1	Repair (Simple)+SolverS1
	Inform User+System	Inform User+System
	Test Repair+Tester1	Test Repair+Tester1
	>>	None
	Restart Repair+System	Restart Repair+System
	>>	None
	>>	None
	Repair (Simple)+SolverS2	Repair (Simple)+SolverS2
	Test Repair+Tester0	Test Repair+Tester0
	>>	None
	Archive Repair+System	Archive Repair+System

When looking at the precision value with the token based approach, when measured with the new log compared to the old log, the precision value decreases slightly since the percentage increase in the number of escaped edges is greater than the increase percentage of activated transitions. Similar results can be obtained with the alignment based approach.

The simplicity metric is calculated through a petri net. The current network is already in petri net format. A petri net representation of the event logs is used as the observed network. The optimal alignment for the alignment based approach will overlap with this network structure. In the token based approach, there will be no remaining or missing tokens on this network. For this reason, the two metrics are calculated with similar logic. The observed value for the petri net obtained from the old event log will be the value from its own structure. For the new event log, the constant value observed from the new network found is used. This value is calculated on the average of the entering and exiting arcs of nodes and transitions. Since the basic arc degrees approach similar values, similar results are obtained.

The generalization metric may differ between token based replay and alignment based replay. The reason for this is that in token based replay, the focus is on the number of operations, while in alignment based replay, other activities that may occur for the relevant state are also taken into account.

A summary table in Table 5.7 containing the results regarding the metric values can be seen.

Table 5.7. Metrics for control-flow and resource mode

Model and log	**Approach**	**Fitness mode**	**Precision mode**	**Simplicity mode**	**Generalization mode**
Old model old log	token based	1.0	0.52	0.86	0.84
	alignment based	0.99	0.52	0.86	0.99
Old model new log	token based	1.0	0.51	0.86	0.84
	alignment based	0.99	0.51	0.86	0.99

We record the model produced by the old record in a similar way with the multi-modal event logs we obtain by combining control-flow and data perspectives.

First of all, as a result of the token based playback we performed with the old log on the old network, it is seen that the 1698 tokens produced in total can be consumed without the remaining and added tokens.

As a result of token based replay in petri net as shown in Figure 5.11, the highest token production and consumption occurs in places n13 and n14. There are also transitions where the repair at n13 results in true. Since the n14 place will be passed before the subsequent repairs, token production and consumption are high. The number of tokens produced and consumed for transitions between n1 and n9 is 100. In addition, n10 and n11 has 96, n12 has 121, n13 and n14 has 196, n15 has 47, n16 has 31, n17 and n18 has 5, n19 has 4, n20 has 5, the number of produced and consumed tokens.

There is no undesirable behavior by the model.

When we perform token based playback with the old network, this time with the new event log, there are 1581 produced tokens and 1605 consumed tokens. The number of added tokens is 66 and the remaining number of tokens is 42.

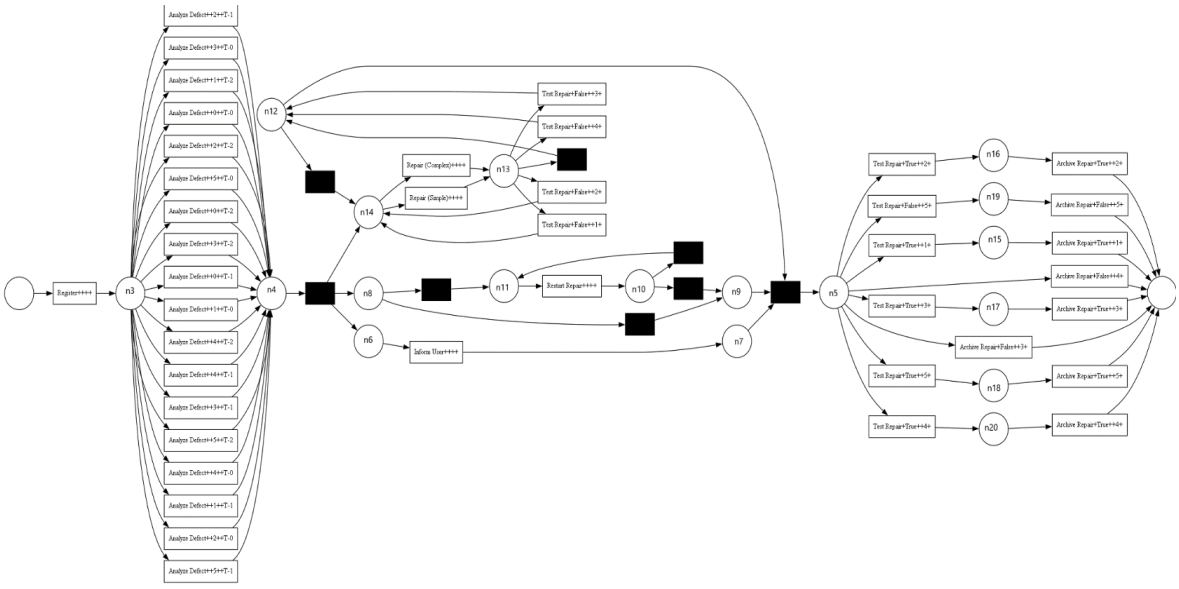

Figure 5.11. Multi-modal Petri net with control-flow and data perspectives

In token based playback, depending on the presence of undesirable behavior, it is possible that tokens remain in places or symbols are added to places. Although the fault type of the undesirable behavior varies, it always depends on the phone type and the new phone type is T-3. The case-based fitness values for token based playback are shown in Table 5.8.

Table 5.8. Control-flow and data mode, token based replay nonfitting cases between old model new log

case_id	**is_fit**	**trace_fitness**	**missing**	**remaining**	**produced**	**consumed**
case_54	FALSE	0.57	3	3	7	7
case_92	FALSE	0.85	3	1	12	14
case_15	FALSE	0.57	3	3	7	7
case_78	FALSE	0.57	3	3	7	7
case_85	FALSE	0.88	3	1	15	17
case_13	FALSE	0.93	3	1	26	28
case_30	FALSE	0.57	3	3	7	7
case_61	FALSE	0.57	3	3	7	7
case_60	FALSE	0.57	3	3	7	7
case_66	FALSE	0.57	3	3	7	7
case_84	FALSE	0.91	3	1	21	23
case_74	FALSE	0.57	3	3	7	7
case_44	FALSE	0.85	3	1	12	14
case_48	FALSE	0.85	3	1	12	14
case_83	FALSE	0.57	3	3	7	7
case_10	FALSE	0.85	3	1	12	14
case_8	FALSE	0.93	3	1	26	28
case_43	FALSE	0.57	3	3	7	7
case_68	FALSE	0.85	3	1	12	14
case_40	FALSE	0.89	3	1	16	18
case_72	FALSE	0.93	3	1	26	28
case_50	FALSE	0.85	3	1	12	14

Predictions can be made based on these values and unwanted activities. These predictions are responsed by the values seen in the Table 5.9 from the alignment based replay and becoming evident in the same cases.

Table 5.9. Control-flow and data mode, alignment based replay nonfitting cases between old model new log

case_id	**fitness**	**is_fit**
case_54	0.82	FALSE
case_92	0.86	FALSE
case_15	0.82	FALSE
case_78	0.82	FALSE
case_85	0.88	FALSE
case_13	0.91	FALSE
case_30	0.82	FALSE
case_61	0.82	FALSE
case_60	0.82	FALSE
case_66	0.82	FALSE
case_84	0.9	FALSE
case_74	0.82	FALSE
case_44	0.86	FALSE
case_48	0.86	FALSE
case_83	0.82	FALSE
case_10	0.86	FALSE
case_8	0.91	FALSE
case_43	0.82	FALSE
case_68	0.86	FALSE
case_40	0.88	FALSE
case_72	0.91	FALSE
case_50	0.86	FALSE

The values in the table vary depending on the case and the main reason is that the calculations are different from each other. In this mode, in token based playback, the fitness value is less than 1 and the resulting undesirable behaviors are used as predictors. Response is taken based on the decrease in fitness values for similar cases with alignment based playback.

For the precision value, when looking at the token based approach and the alignment based approach in this mode, the same value is taken as the old log compared to the old model. In case of using the new log and the old model, the token based approach is slightly inferior due to the presence of new behaviors in the new event log. Similar results can be obtained with the created enhanced model.

Similar to the other mode, the generalization metric is higher for alignment based replay due to the lower penalty imposed by alignment based calculation for non-compliance.

In the mode formed by adding control-flow and data perspectives, the simplicity metric is similar to the use of these values in the observed model, since the basic arc degrees are similar between the old event log and the new event log. However, since the new model found is the modeled network itself, the value goes up slightly.

A summary of the metric values can be seen in Table 5.10.

Table 5.10. Metrics for control-flow and data mode

Model and log	**Approach**	**Fitness mode**	**Precision mode**	**Simplicity mode**	**Generalization mode**
Old model old log	token based	1.0	0.52	0.95	0.63
	alignment based	0.99	0.52	0.95	0.99
Old model new log	token based	0.96	0.54	0.95	0.62
	alignment based	0.96	0.57	0.95	0.99
Enhanced model new log	token based	1.0	0.55	0.97	0.61
	alignment based	0.99	0.55	0.97	0.99

Replay with control and time perspectives will not be detailed. However, the metric results related to this mode are in Table 5.11 is also located.

Table 5.11. Metrics for control-flow and time mode

Model and log	**Approach**	**Fitness mode**	**Precision mode**	**Simplicity mode**	**Generalization mode**
Old model old log	token based	1.0	0.96	0.80	0.90
	alignment based	1.0	0.96	0.80	0.99
Old model new log	token based	1.0	0.96	0.80	0.90
	alignment based	1.0	0.96	0.80	0.99

6. DISCUSSION

Two research questions asked within the scope of this thesis can be described as major. These questions are addressed as RQ3 and RQ4. According to RQ3, the feasibility of detecting new behavior with the multi-perspective explorer is questioned. With RQ4, it is asked how to detect the new behavior with root causes and metrics, with the idea that this analysis can be done with a multi-modal approach and processes that improve with attribute effective representation. Additionally, it is questioned how the existing studies in the literature and the designed structure will be discussed in the context of concepts such as virtual-smart factory.

As a result of the examinations, since decision tree-based rules are deduced at the guard points with the multi-perspective explorer and these rules do not cover all the features that the tree takes as input, the detection of new behavior cannot be achieved (i.e., by multi-perspective inputs). Since the multi-perspective explorer does not provide token-based replay, variant analysis can be used to achieve the response purpose of root cause analysis with multi-modal input. An important emphasis at this point is that, in this study, the response approach for root cause analysis is provided with the accepted alignment-based approach instead of stakeholder or decision tree-based one. In addition, for cases with the same service times, it will be able to get response from control and resource mode about the prediction that is related to time to be made with control and data mode. For cases similar to the study within the scope of the thesis, event logs enriched with the time between activities will be able to support the response. Variants are filtered with query support, and a quick response can be received. The assumption here is that the times are calculated properly. No additional analysis will be needed. In this state, it can be said that root cause analysis has become somewhat hybrid.

With the help of other perspective data added to the control flow perspective, the results obtained after replay through multi-modal event logs and petri nets. Root cause detection can be made with the help of metrics. The interpretation about the fitness metric is that the behavior seen in the system and log but not included in the model can be achieved with model repair. By viewing the parts that perform the virtual production of event logs as a system, the representation of the process is improved with the new behavior. The general

comment made within the scope of generalization is that when the behavior in the system and log is partially provided by the model, it is described as noise. However, in our case, it may be new behavior rather than noise and it may reduce the generalization slightly.

Within the scope of G4 the main contribution can be seen as the artifact itself given in accordance with the DSR and G1. If the contribution is the artifact itself, RQ2 can also be included in the major question category. With the extension of Kafka, the messaging system will be seen as an important foundation in the literature, on both sides (i.e., producer and consumer), as a contribution in terms of expansion in accordance with the foundations. As a critical evaluation method, the review methodology that set the ground for the development process is another contribution.

Due to the library used on the generator side, event logs related problems may arise that the production cannot be achieved at the desired level. Additionally, there may be cases where convergence does not occur directly after divergence with the XOR gate. For these cases, the Python language can be used for multi-modal event logging production purposes for updating and testing only with the presence of the relevant libraries. The use of the ProM tool for similar purposes only with the relevant plugins is abstracted from the framework. We have tried to show that it can evolve from a production-based constraint to a contribution with a feature added to the artifact.

For the event log studied within the scope of the thesis, the occurrence of events is obtained instantly. In case the events do not occur instantly and service times vary, the difference between event times can be estimated with inductive visual miner visualization via ProM. The response to the prediction can be received with variant analysis visualization. This treatment of time brings the solution partially closer to multi-perspective variant analysis. Depending on changing event logs, the analysis may differ depending on the scope of the root cause analysis. We think that in addition to service time, waiting time plus service time (i.e., sojourn time) is also added to the analysis. This analysis can be shaped both only by behavior and other attributes added to it. On the server side, it is assumed that the pm4py framework and all libraries according to the analysis are included. When you want to perform analysis only with pm4py, an analysis based on both token and layout can be performed with the time perspective added to the control perspective as a result of time-related parsing operations.

Other constraints are that only the XES data format is given at this stage for framework event log generation, and only a sound algorithm can produce proper results. Additionally, when obtained through sequence flow control discovery for the BPMN model, no activity should be filtered.

The solution proposed provides research rigor in the context of DSR G5. The work itself must also be against excessive formalism. In accordance with this guide, metrics with a solid mathematical foundations are used. Interventions related to event record production or parsing operations can be viewed as abstract. These operations can be considered as a result of an effort to be mathematically rigorous. It can be assumed that these operations are performed properly.

In the introduction, the importance of increasing data as a result of the industrial revolutions was emphasized. It was also stated that PM gained importance with the 4th Industrial Revolution in order to make sense of process-oriented data. The types related to the virtual factory are the evolution of production from the process of emulating the factory ability to a complete emulation by incorporating employee behavior. This study may be seen as limited in that it only performs process-based operations. However, the work is expected to have a widespread impact, as it can be modeled as any production system, social system, or process in hardware and software. Therefore, any modeled system can be examined multi-modally rather than multi-perspectively.

With the study, the feature-based decision mechanism provided by the multi-perspective explorer can be seen with control and other perspective data added to the control perspective. Additionally, four metrics related to model quality are given. Fitness is the most important metric and directly affects decision analysis. With generalization, information-increasing clues about the system can be obtained for the changing behavior of the system. The simplicity metric supports the idea that a model with generalization should be simple. The precision metric may change even with event and case-based frequencies. This is also related to the generalization of the model.

7. CONCLUSION

The first research question that led this thesis asks what the current studies are. The scope of the relevant studies is expanded to include the multi-perspective structure, with the emergence of the expression multi-modal in the multi-perspective structure, as multi-modal studies are rarely found in the literature. Additionally, any study in the field definitely uses at least one perspective of the multi-perspective structure. Therefore, even if the analysis is examined multi-modally, multiple data from different perspectives are used. The second point of expansion is RCA. Because, in many studies using PM improvement methods, improvement is achieved through RCA. By using the divide and conquer approach in the literature review methodology, a synthesis is achieved with the divided subfields.

Within the scope of related studies, multi-modal analysis, which provides anomaly detection in data science-oriented processes that intersects with multi-modal analysis and PM, has been examined. It has been observed that the multi-modal definition has gained modularity in different areas along with other multi-modal studies. With predictive process monitoring, artificial intelligence-supported investigations, which can also be associated with the field of data science, are carried out. These data science-based studies are fault-tolerant.

The second research question aims to discuss the designed PM system in the context of digital transformation and virtual-smart factories. Accordingly, a framework has been designed in accordance with Design Science Rresearch approach. Framework can be called artifacts of artifacts by using separate artifacts together. The data production and producer parts for the provided artifacts can be seen as a virtual factory, and the consumer part, which produces a process representation different from the process representation it receives, can be seen as another virtual factory. Since these two virtual factories interact with each other, they can be considered as a smart factory. In this study, we propose a new term that we call virtual factory, which makes a virtual factory in a way that an event log can be produced from the one representation of the process and an another representation of the process can be produced from this log.

Additionally, unlike traditional messaging systems, our proposal appears to be fully compatible with distributed messaging. We are talking about using a distributed architecture, as opposed to only messaging between services on the server side. The framework proposed in this study provides isolation by dividing the consumer part in the context of data processing. This is expected to prevent possible data loss under heavy traffic.

In parallel with the third research question, two different investigations are carried out within the scope of the case study in order to show the difference between multi-perspective analysis and multi-modal analysis. Accordingly, when the model representation of the process and the event log representation differ from each other, behaviors that do not comply with the multi-perspective explorer are eliminated primarily based on the control-flow perspective-oriented fitness metric. Afterwards, by adding the decision tree created for the decision points to the petri net, the multi-perspective explorer cannot be used to detect new behavior due to the error made based on the decision tree. Performance evaluation for the two guard points of the decision tree is provided through confusion matrix and ROC curve.

Secondly, when we want to perform analysis with the multi-perspective explorer, with the event log and process representation evolving into a multi-modal structure by articulating the control-flow perspective with other perspective data, unfit behaviors are eliminated after the compliance check in the first stage, since the compliance control includes separate stages. Therefore, the analysis cannot continue.

In line with the fourth research question, multi-modal analysis is aimed to perform root cause analysis based on metrics. Regarding the fourth question, in this study, while performing conformance control based on metrics, other perspective data can be included in the analysis at the same time. As a result of playback with other perspective data included in the root cause analysis, prediction is made with the token based approach with the decrease in fitness metrics and the presence of undesirable behaviors. In the alignment based approach, a response is received when the alignment decreases for similar cases. Thus, misleading diagnoses in the root cause analysis made with the decision tree can be prevented. For other metrics, values between 0 and 1 are obtained and appear to be acceptable. The precision value may be lower in the advanced model. This is because paths with a low frequency of occurrence in the event log are included in the model. The generalization metric differs between token based and alignment based replays. While the number of executions is taken into account in the token based approach, other executions that are likely to be seen in the

alignment based approach are also included in the calculation. In order to present ideal values for the simplicity metric, the difference in arc degrees of the modeled and observed network structure is used in absolute value. As a result of the root cause analysis, the model, which is expanded with other perspective data, is improved. Thus, expansion and improvement seen within the scope of enhancement can be done together.

In addition, since statistics, as one of the other disciplines related to PM, is based on established hypotheses, the cost of a mistake can be very heavy. In data science, the algorithms developed are error-tolerant models, and especially at the point of classification, a result that belongs to a class that it does not belong to can be produced. This study prevents such errors. In the articles where the multi-perspective structure is first encountered, it is stated that, despite the perspectival structure, conformance checking is not used.

The definition of conformance checking given within the scope of this thesis can be reduced to three categories. First, deviations that occur within any perspective are not compatible with other data in the same perspective. Second, the aim is to detect deviant behavior by using other components of the multi-perspective structure with data science support. The third definition is the control of conformance with two different process representations over quality metrics. Originally, quality criteria and conformance checking are carried out together in the multi-modal structure by combining the components of the multi-perspective structure. In other words, there is an integrated application of two different conformance control definitions.

8. REFERENCES

[1] W.M.P. Van Der Aalst, Process Mining: Data Science in Action, **2012.** https://doi.org/10.1007/s00287-012-0641-4.

[2] M. Grieves, J. Vickers, Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems, Transdiscipl. Perspect. Complex Syst. New Find. Approaches (**2016**) 85–113. https://doi.org/10.1007/978-3-319-38756-7_4.

[3] S. Jain, N.F. Choong, K.M. Aye, Virtual factory: An integrated approach to manufacturing systems modeling, Int. J. Oper. Prod. Manag. 21 (**2001**) 594–608. https://doi.org/10.1108/01443570110390354.

[4] İ.E. EMRE, Ç. SELÇUKCAN EROL, Veri Analizinde İstatistik mi Veri Madenciliği mi?, Bilişim Teknol. Derg. (**2017**) 161–161. https://doi.org/10.17671/gazibtd.309297.

[5] J. Schalken, H. van Vliet, Measuring where it matters: Determining starting points for metrics collection, J. Syst. Softw. 81 (**2008**) 603–615. https://doi.org/10.1016/j.jss.2007.07.041.

[6] W. M. P. Van Der Aalst, B. van D. Carmona, Process Mining Handbook, **2022**. https://doi.org/10.1007/978-3-031-08848-3_9.

[7] T.G. Erdogan, A.K. Tarhan, Multi-perspective process mining for emergency process, Health Informatics J. 28 (**2022**). https://doi.org/10.1177/14604582221077195.

[8] F. Mannhardt, M. De Leoni, H.A. Reijers, The multi-perspective process explorer, (**2023**) 130–134.

[9] A. Rullo, A. Guzzo, E. Serra, E. Tirrito, A Framework for the Multi-modal Analysis of Novel Behavior in Business Processes, Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) 12489 LNCS (**2020**) 51–63. https://doi.org/10.1007/978-3-030-62362-3_6.

[10] A. Rebmann, A. Emrich, P. Fettke, Enabling the Discovery of Manual Processes Using a Multi-modal Activity Recognition Approach, Springer International Publishing, **2019**. https://doi.org/10.1007/978-3-030-37453-2_12.

[11] T. Bi, P. Liang, A. Tang, Architecture Patterns, Quality Attributes, and Design Contexts: How Developers Design with Them, Proc. - Asia-Pacific Softw. Eng. Conf. APSEC 2018-Decem (**2018**) 49–58. https://doi.org/10.1109/APSEC.2018.00019.

[12] J.P. Alan R. Hevner, Sudha Ram, Salvatore T. March, Design Science In Information System Research, AI Soc. 10 (**1996**) 199–217. https://doi.org/10.1007/BF01205282.

[13] S.J.J. Leemans, D. Fahland, W.M.P. Van Der Aalst, Process and deviation exploration with inductive visual miner, CEUR Workshop Proc. 1295 (**2014**) 46–50.

[14] A. Rebmann, J.R. Rehse, H. van der Aa, Uncovering Object-Centric Data in Classical Event Logs for the Automated Transformation from XES to OCEL,

Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) 13420 LNCS (**2022**) 379–396. https://doi.org/10.1007/978-3-031-16103-2_25.

[15] M. Chinosi, A. Trombetta, BPMN: An introduction to the standard, Comput. Stand. Interfaces 34 (**2012**) 124–134. https://doi.org/10.1016/j.csi.2011.06.002.

[16] F.A. Yasmin, Faculty of Electrical Engineering , Mathematics & Computer Science Enhancement in Process Mining Guideline for Process Owner and Process Analyst, (**2019**).

[17] B. Ekici, T.G.E. Erdogan, A.P. PhD. Ayça Kolukısa Tarhan, BPMN Data Model for Multi-Perspective Process Mining on Blockchain _ International Journal of Software Engineering and Knowledge Engineering, (n.d.).

[18] J.L. Peterson, Petri Nets, ACM Comput. Surv. 9 (**1977**) 223–252. https://doi.org/10.1145/356698.356702.

[19] W.M.P. Aalst, A. Adriansyah, A. Karla, A. de Medeiros, F. Arcieri, T. Baier, T. Blickle, R. Bose, P. Van Den Brand, R. Brandtjen, A. Guzzo, P. Harmon, A. Hofstede, J. Hoogland, J.E. Ingvaldsen, K. Kato, D. Malerba, R. Mans, A. Manuel, M. Mccreesh, M. Muehlen, J. Munoz-gama, L. Pontieri, J. Ribeiro, A. Rozinat, H.S. P, R.S. P, M. Sep, J. Sinur, P. Soffer, M. Song, Process Mining Manifesto, Lect. Notes Bus. Inf. Process. 99 (2012) 169–194. www.win.tue.nl/ieeetfpm/.

[20] O. Doğan, Overview of Process Mining: Alpha Algorithm for Process Flow Discovery, Pamukkale Univ. J. Eng. Sci. 26 (**2020**) 966–973. https://doi.org/10.5505/pajes.2020.57418.

[21] A.J.M.M. Weijters, W.M.P. van der Aalst, A.K.A. de Medeiros;, Process Mining with the HeuristicsMiner Algorithm, Beta Work. Pap. (**2006**).

[22] C.W. Günther, W.M.P. Van Der Aalst, Fuzzy mining - Adaptive process simplification based on multi-perspective metrics, Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) 4714 LNCS (**2007**) 328–343. https://doi.org/10.1007/978-3-540-75183-0_24.

[23] A. Adriansyah, J. Munoz-Gama, J. Carmona, B.F. van Dongen, W.M.P. van der Aalst, Measuring precision of modeled behavior, Inf. Syst. E-Bus. Manag. 13 (**2015**) 37–67. https://doi.org/10.1007/s10257-014-0234-7.

[24] A. Berti, W. Van Der Aalst, Reviving token-based replay: Increasing speed while improving diagnostics, CEUR Workshop Proc. 2371 (**2019**) 87–103.

[25] B.F.V.A.N. Dongen, C. Science, Quality Dimensions In Process Discovery: The Importance of Fitness, Precision, Generalization and Simplicity, (**2013**) 1–40.

[26] J.B. van D. Carmona, Conformance Checking, **2022**. https://doi.org/10.1007/978-3-030-96655-3_7.

[27] J.R. Rehse, L. Pufahl, M. Grohs, L.M. Klein, Process Mining Meets Visual Analytics: The Case of Conformance Checking, Proc. Annu. Hawaii Int. Conf. Syst. Sci. 2023-Janua (**2023**) 5452–5461.

[28] P.J.M.B. R.S. Mans, M.H. Schonenberg, M. Song, W.M.P. van der Aalst, Application of Process Mining in Healthcare – A Case Study in a Dutch Hospital, Commun. Comput. Inf. Sci. 25 (**2009**) 189–201. https://doi.org/10.1007/978-3-540-92219-3.

[29] F. Almira Yasmin, R. Bemthuis, M. Elhagaly, F. Wijnhoven, F. Allah Bukhsh, A Process Mining Starting Guideline for Process Analysts and Process Owners: A Practical Process Analytics Guide using ProM, (**2020**) 1–18. www.processmining.org.

[30] A. Rozinat, W.M.P. van der Aalst, Decision mining in business processes, BPM Cent. Rep. BPM-06-10, … 164 (**2006**) 16. http://www.processmining.org/_media/publications/beta_164.pdf.

[31] F. Mannhardt, D. Blinde, Analyzing the trajectories of patients with sepsis using process mining, CEUR Workshop Proc. 1859 (**2017**) 72–80.

[32] K. Gajowniczek, T. Ząbkowski, R. Szupiluk, Estimating the Roc Curve and Its Significance for Classification Models' Assessment, Quant. Methods Econ. XV (**2014**) 382–391.

[33] Online.Available https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc.

[34] C. Ferri, P. Flach, J. Hernández-Orallo, Learning decision trees using the area under the ROC curve, Int. Conf. Mach. Learn. (**2002**) 139–146. http://www2.cs.ust.hk/~qyang/537/Papers/flachICML02.pdf.

[35] J. Muñoz-Gama, Conformance Checking, Lect. Notes Bus. Inf. Process. 440 (**2022**) 327–354. https://doi.org/10.1007/978-3-030-96655-3_7.

[36] J. Carmona, Conformance Checking: Foundations, Milestones and Challenges, Springer International Publishing, **2022**. https://doi.org/10.1007/978-3-030-96655-3_7.

[37] W. Van der Aalst, A. Adriansyah, B. Van Dongen, Replaying history on process models for conformance checking and performance analysis, Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 2 (**2012**) 182–192. https://doi.org/10.1002/widm.1045.

[38] F.R. Blum, Metrics in process discovery, Tech. Rep. TR/DCC (2015) 1–21.

[39] J. Muñoz-Gama, J. Carmona, A fresh look at precision in process conformance, Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) 6336 LNCS **(2010**) 211–226. https://doi.org/10.1007/978-3-642-15618-2_16.

[40] F.A. Yasmin, F.A. Bukhsh, P. De Alencar Silva, Process enhancement in process mining: A literature review, CEUR Workshop Proc. 2270 (**2018**) 65–72.

[41] M. de Leoni, Foundations of Process Enhancement, Springer International Publishing, **2022**. https://doi.org/10.1007/978-3-031-08848-3_8.

[42] B. Stopford, Designing systems, **2003**. https://doi.org/10.4324/9781315642833-6.

[43] A. Burattin, PLG2: Multiperspective process randomization with online and offline simulations, CEUR Workshop Proc. 1789 (**2016**) 1–6.

[44] J. Walkenbach, Kafka the Definitive Guide, **2010**.

[45] J.A. Shaheen, Apache Kafka: Real Time Implementation with Kafka Architecture Review, Int. J. Adv. Sci. Technol. 109 (**2017**) 35–42. https://doi.org/10.14257/ijast.2017.109.04.

[46] W.M.P. van der A. Maikel L. van Eck(B), Xixi Lu, Sander J.J. Leemans, PM2: A Process Mining Project Methodology, Lect. Notes Comput. Sci. (Including Subser.

Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) 9097 (**2015**) 520–521. https://doi.org/10.1007/978-3-319-19069-3.

[47] S. Suriadi, M.T. Wynn, C. Ouyang, A.H.M. Ter Hofstede, N.J. Van Dijk, Understanding process behaviours in a large insurance company in Australia: A case study, Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) 7908 LNCS (**2013**) 449–464. https://doi.org/10.1007/978-3-642-38709-8_29.

[48] R. Andrews, F. Emamjome, A.H.M. Ter Hofstede, H.A. Reijers, An expert lens on data quality in process mining, Proc. - 2020 2nd Int. Conf. Process Mining, ICPM 2020 (2020) 49–56. https://doi.org/10.1109/ICPM49681.2020.00018.

[49] M.N. Tiftik, T. Gurgen Erdogan, A. Kolukisa Tarhan, A framework for multi-perspective process mining into a BPMN process model, Math. Biosci. Eng. 19 (2022) 11800–11820. https://doi.org/10.3934/mbe.2022550.

[50] T.G. Erdogan, A. Tarhan, A goal-driven evaluation method based on process mining for healthcare processes, Appl. Sci. 8 (**2018**). https://doi.org/10.3390/app8060894.

[51] K. Goel, S.J.J. Leemans, M.T. Wynn, A.H.M. ter Hofstede, J. Barnes, Improving PhD Student Journeys with Process Mining: Insights from a Higher Education Institution, CEUR Workshop Proc. 3112 (**2021**) 39–49.

[52] S. Aguirre, C. Parra, J. Alvarado, Combination of Process Mining and Simulation Techniques for Business Process Redesign: A Methodological Approach, Lect. Notes Bus. Inf. Process. 162 (**2013**) 24–43. https://doi.org/10.1007/978-3-642-40919-6_2.

[53] S. Lagraa, R. State, Process mining-based approach for investigating malicious login events, Proc. IEEE/IFIP Netw. Oper. Manag. Symp. 2020 Manag. Age Softwarization Artif. Intell. NOMS 2020 2 (**2020**). https://doi.org/10.1109/NOMS47738.2020.9110301.

[54] A. Meincheim, C.D.S. Garcia, J.C. Nievola, E.E. Scalabrin, Combining process mining with trace clustering: Manufacturing shop floor process-an applied case, Proc. - Int. Conf. Tools with Artif. Intell. ICTAI 2017-Novem (**2018**) 498–505. https://doi.org/10.1109/ICTAI.2017.00082.

[55] I. Ketykó, F. Mannhardt, M. Hassani, B. van Dongen, What Averages Do Not Tell -- Predicting Real Life Processes with Sequential Deep Learning, (**2021**). http://arxiv.org/abs/2110.10225.

[56] J. Lahann, Multimodal Process Prediction (Extended Abstract), CEUR Workshop Proc. 3299 (**2022**) 32–36.

[57] A. Nguyen, S. Järvelä, C. Rosé, H. Järvenoja, J. Malmberg, Examining socially shared regulation and shared physiological arousal events with multimodal learning analytics, Br. J. Educ. Technol. 54 (**2023**) 293–312. https://doi.org/10.1111/bjet.13280.

[58] R. Akhavian, A.H. Behzadan, Knowledge-Based Simulation Modeling of Construction Fleet Operations Using Multimodal-Process Data Mining, J. Constr. Eng. Manag. 139 (**2013**) 1–11. https://doi.org/10.1061/(asce)co.1943-7862.0000775.

[59] T.O.F. Content, E. Summary, B. Of, T.H.E. Village, D. Of, L. Use, Enterprise Integration Patterns, (**2000**) 1–26.

[60] A. Telli, T.G. Erdogan, A. Kolukisa, Detecting Novel Behavior and Process Enhancement with Multimodal Process Mining, 4th Int. Informatics Softw. Eng. Conf. - Symp. Program, IISEC 2023 (**2023**). https://doi.org/10.1109/IISEC59749.2023.10391012.

[61] M. Mattsson, J. Bosch, Framework composition: Problems, causes and solutions, Proc. Conf. Technol. Object-Oriented Lang. Syst. TOOLS (**1997**) 203–214. https://doi.org/10.1109/tools.1997.654724.

[62] C. Wanzeller, Association for Information Systems Comparative Analysis of Process Mining Tools Polytechnic Institute of Viseu , estgv15362@alunos.estgv.ipv.pt, (**2022**).

[63] G. Tu, Verification of WF-nets, **2023**. https://doi.org/10.6100/IR577287.