UHASSELT

KNOWLEDGE IN ACTION

Faculty of Business Economics

Master of Management

Master's thesis

Evaluating the Comprehensibility of Control Flow Models Mined from the Sepsis Event Log

Killian Amunkeng Zelefac

Thesis presented in fulfillment of the requirements for the degree of Master of Management, specialization Business Process Management

SUPERVISOR :

Prof. dr. Niels MARTIN

UHASSELT

KNOWLEDGE IN ACTION

www.uhasselt.be
Universiteit Hasselt
Campus Hasselt:
Martelarenlaan 42 | 3500 Hasselt
Campus Diepenbeek:
Agoralaan Gebouw D | 3590 Diepenbeek

2021
2022

Faculty of Business Economics

Master of Management

Master's thesis

Evaluating the Comprehensibility of Control Flow Models Mined from the Sepsis Event Log

Killian Amunkeng Zelefac

Thesis presented in fulfillment of the requirements for the degree of Master of Management, specialization Business Process Management

SUPERVISOR :

Prof. dr. Niels MARTIN

Evaluating the Comprehensibility of Control Flow Models Mined from the Sepsis Event Log

Zelefac Killian Amunkeng
Master of Management: Business Process Management
Faculty of Business Economics, Hasselt University,
Martelarenlaan 42, 3500 Hasselt, Belgium

Supervisor: Prof. dr. Niels MARTIN

Abstract

To tackle the challenge of tightening budgets and at the same time achieve high quality of care standards, hospitals are now conscious of the necessity to comprehend their processes in order to optimize them. In this regard, a promising solution to unearth the actual behavior of an executed process information arising from the event log collected by a health information system (HIS) might be the use of process mining. The research question of this study is "How are the outputs of control-flow discovery algorithms evaluated by end-users in terms of comprehensibility when applied to the sepsis event log?". To answer the research question, an experiment was conducted with 19 students of the Master of Management: Business Process Management at Hasselt University. In the first phase of this experiment, the models were mined from the raw log using PM4PY and bupaR as the process mining tools. In the second phase, the mined models were later evaluated and compared by the students concerning their comprehensibility in an online survey using qualtrics. In the last phase, data was gathered via the online survey for analysis. Four findings emerged from this study. Firstly, the findings reveal that the students perceived the control-flow of the Process tree model as the easiest to comprehend. Secondly, they perceived the notation of the Causal net as the most suitable for presenting healthcare processes in a comprehensible manner. Thirdly, the students find it appealing to use all three notations for presenting healthcare processes. Lastly, Friedman's test shows no significant difference across the various control-flow models concerning the end-users perceived comprehensibility.

Keywords - *bupaR, Causal net, Comprehensibility, Friedman's test, PM4PY, Petri net, Process tree,*

Acknowledgements:

At this stage, I wish to show my appreciation to everyone who inspired, supported, and motivated me through this journey.

First of all, I would start by thanking all the Master of Management (Business Process Management) students at Hasselt university who took part in my study. Without you, this thesis would not have been completed.

Secondly, I would like to thank my supervisor, Prof. dr. Niels MARTIN, for his directions and pointers towards the accomplishment of this thesis. Thank you for pushing me to my limits. Through you, I have learned a lot about conducting and reporting research. I look forward to improving these skills.

Lastly, I give big thanks to my entire family and my friends back home in Cameroon. You have shown me that distance is not a barrier. Your words of encouragement and motivation served as my strength during the process of my research work.

1 Introduction

Nowadays, the healthcare domain is faced with profound challenges [1]. The most remarkable challenge is the conflict between budgets constraint and increasing demand for care services as a result of the aging population [2, 3]. To tackle these challenges and at the same time achieve high quality of care standards, hospitals are now conscious of the necessity to comprehend their processes in order to optimize them [4]. A process is a group of interconnected activities, decisions, and events that creates value for the organization and its customers [5]. Therefore, a healthcare process is a series of activities targeted at diagnosing, curing, and preventing any disease in order to boost a patient's health [6].

In literature, a distinction is typically made between *medical treatment processes* and *organizational (or administrative) processes* [7–9]. The former involves the processes for administering patients, including activities stretching from diagnosis to measures for healing the patients [10]. The latter are collective process arrangements that aid medical treatment processes as a whole [9]. They are not customized for a particular task but strive to organize medical treatment between various individuals and units (for example, patient registration and diagnosis requests) [9].

Healthcare processes as a whole and medical treatment processes specifically, bear some distinct features compared to normal business processes, for example, an order to cash process [9]. Healthcare processes can be regarded as loosely structured and knowledge-based [11, 12]. On the one hand, a loosely structured process can be executed in a broad, but limited and pre-conceived, number of different ways [12]. This has to do with the observation that healthcare processes usually display some degree of variability [9]. On the other hand, the knowledge-based nature means that the accomplishment of healthcare processes are highly dependable on skilled personnel (for instance, doctors) and the knowledge-based decisions they make [11]. The complexity of these decisions are based on a broad range of criteria, including clinical know-how, patient-related features, and healthcare professionals' expertise [9, 11, 13]. Moreover, the multidisciplinarity of the healthcare sector implies that doctors have to work together across their specializations and departments, which adds to the complexity [9]. Apart from the complexity and multidisciplinarity, healthcare processes evolve due to changes in medical know-how, administrative procedures, or technological developments in healthcare [14].

To improve healthcare processes, a healthcare institution has to gain an extensive understanding of the process at stake [1]. To collect information on how the process is carried out, domain experts who are knowledgeable about the process can be mobilized for a dialogue. This dialogue can be centered on the development of process models captivating process insights (for example, the order of activities in a medical treatment process), which constitute a foundation for a process assessment [1]. Nevertheless, this procedure is time-consuming, and the constructed process model usually presents an idealized view that might have little association with reality [15].

A promising solution to unearth the real behavior of an executed process information arising from the data collected by a process aware information system (PAIS) might be the use of process mining (PM) (Figure 1). A PAIS is an information system that controls and carries out operational processes encompassing individuals, applications, and/or information sources derived on process models [16]. Examples of PAIS are health information systems (HIS), workflow management systems, case handling systems, enterprise information systems, and so forth [16]. By obtaining data found in the data bases of HIS, for example, an event log can be created consisting of a detailed process execution data for the relevant healthcare process [1]. As such the event is composed of real-life data about the activities that have been executed, the time they were executed, who executed them and for whom (for instance, for which patient) [15, 17].

PM is a novel discipline that lies in between, on the one hand machine learning and data analysis and, on the other hand, process model-driven techniques [18]. Three types of PM methods exist: discovery, conformance checking, and enhancement. Van der Aalst [18] explained how process discovery techniques allow a process model to be extracted from event log data; how conformance checking techniques monitor deviations by comparing the event log and the a priori model; and how enhancement improves an existing process model using information about the actual process recorded in the event log.

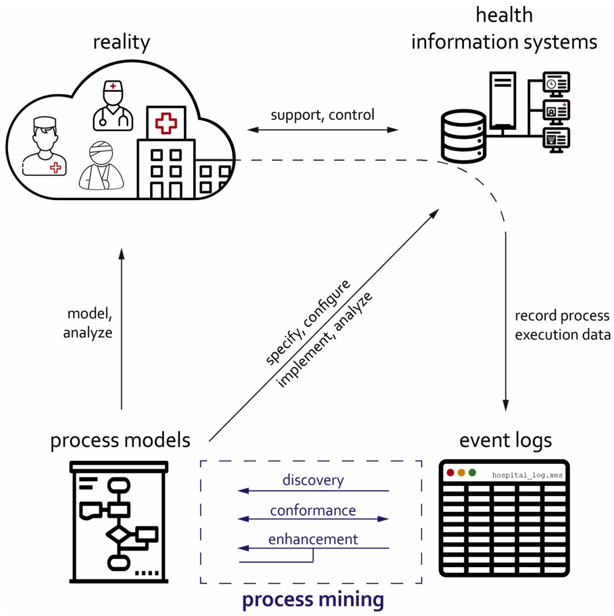


Figure 1: Position of Process Mining in Healthcare [1]

PM with healthcare data has used diverse algorithms for control-flow model discovery. According to Benevento et al. [19], the most promising methods are Heuristic Miner [20], Fuzzy Miner [21], Split Miner [22], and Inductive Miner [23]. Despite the fact that these algorithms can handle noisy and incomplete event log [6, 24], they suffer from two weaknesses when applied to real-life event logs [25]: (1) they generate large and spaghetti-like models (highly complex and lack structure, hence hindering their comprehensibility) [26], and (2) they generate process models that do not find the right trade-off between simplicity, fitness, generalization, and precision [27]. Consequently, if these models are challenging to comprehend, they fail to serve the purpose they are meant for (such as process assessment, process optimization, etc.) [27].

The assessment of control-flow discovery algorithms is mostly carried out by utilizing real-life event logs (for instance, real-life healthcare logs), where the mined control-flow models are evaluated using diverse metrics [27]. This is because evaluations of these real-life logs would provide important information regarding the practical application of certain control-flow discovery algorithms in general [28]. Nevertheless, the control-flow models are barely appraised by end-users or domain experts [27]. In this regard, this thesis aims to discover control-flow models of a real-life log (sepsis event log [29]) in the healthcare domain. In discovering the control-flow models, the focus will be to compare the models produced by three control-flow discovery algorithms (Alpha Miner; α Miner, Inductive Miner; IM, and Heuristic Miner; HM) with respect to the Comprehensibility (or Understandability) dimension. Comprehensibility refers to the features that a control-flow model possesses which can easily be comprehended by a reader of that model, from which correct conclusions are derived [30]. Since comprehensibility is essential if an institution (such as healthcare institution) would like to use the control-flow model for process improvement purposes or communication purposes, the following research question is formulated:
How are the outputs of control-flow discovery algorithms evaluated by end-users in terms of comprehensibility when applied to the sepsis event log?

To answer the above research question, this thesis reports the results of an experiment involving students of the master of business process management, aimed at assessing control-flow models with respect to their perceived comprehensibility (perceived ease of use and intention to use). As such, the contribution of this thesis is a comparative assessment of the outputs of control-flow discovery algorithms from the students' (end-users) point of view.

The remainder of this thesis is structured as follows. Section 2 presents the related literature review. Section 3 discusses the methodology and and experimental design. Section 4 presents the results. Section 5 discusses the results and limitations of the study. The thesis ends with Section 6, which concludes and points out some recommendations for future research.

2 Literature Review

This section provides an overview of the related literature. Section 2.1 presents some control-flow discovery algorithms with respect to their modeling notation. Section 2.2 explains the dimensions for evaluating the quality of control-flow models. Finally, Section 2.3 explains how the comprehensibility of control-flow model is operationalized.

2.1 Control-Flow Discovery Algorithms

As mentioned in Section 1, PM comprises three types (discovery, conformance checking, and enhancement). Process discovery (PD) generates process models out of an event log [1]. On the one hand, an event log is a collection of events related to a case, for instance a patient's visit to the hospital [1]. One event denotes a 'thing' which occurred during the process, which induces a transformation in the HIS and is recorded [31]. In many instances, an event refers to the realization of a medical or non-medical activity (for example, the start of a medical checkup for a specific patient or the termination of a patient's registration) [1]. On the other hand, a process model refers to the complete life cycle of a case [15]. The process models mined from such event log are represented in a known process modeling notation, such as: Business Process Modeling Notation (BPMN) [32], Petri nets [15], Workflow nets [15], Causal nets (or C-net) [18], and Process tree [15], and so on.

A critical part of PD is control-flow discovery, which extracts and visualizes the order of activities based on an event log [15]. An overview (the list is not exhaustive) of some control-flow discovery algorithms (ranging between 2012 and 2018) is presented in Table 1.

Table 1: Overview of Some Control Flow Discovery Algorithms (Adapted from [33])

Techniques	Main References	Year	Model Notation
HK	Huang and Kumar [34]	2012	Petri net
AIM	Carmona and Cortadella [35]	2013	
Prom-D	Song et al. [36]	2015	
Alpha $	Guo et al. [37]	2016	
RegPFA	Breuker et al. [38]	2016	
TAU Miner	Li et al. [39]	2016	
CoMiner	Tapia-Flores et al. [40]	2016	
Decomposeed Process Miner	Verbeek et al. [41]	2017	
Hybrid ILP Miner	Van Zelst et al. [42]	2018	
Competition Miner	Redlich et al. [43]	2014	BPMN
DGEM	Molka et al. [44]	2015	
BPMNMiner	Conforti et al. [45]	2016	
Split Miner	Augusto et al. [22]	2017	
Fodina	Vanden Broucke and De Weerdt [46]	2017	
Heuristic Miner	Augusto et al. [47]	2018	
CNMining	Greco et al. [48]	2015	Causal net
ProDiGen	Vázquez-Barreiros et al. [49]	2015	
Maximal Pattern Mining	Liesaputra et al. [50]	2016	
Proximity Miner	Yahya et al. [51]	2016	
Stage Miner	Nguyen et al. [52]	2017	
Inductive Miner:Infrequent	Leemans et al. [23]	2013	Process Tree
Evolutionary Tree Miner	Vanden Broucke et al. [53]	2014	

A few of these miners in Table 1 are described in the following paragraphs. For an extensive overview of existing miners for control-flow discovery, the reader is referred to the literature reviews by Soo [54] and Augusto et al. [33].

Petri net Models

To overcome the problem that most Petri net miners faced when confronted with large event logs (that is, logs with more than 50 activities), an approach for tackling this problem was proposed by Verbeek et al. [41]. This technique aims at dividing the event logs into diverse sub-logs. All the models mined from these small logs are then merged to produce a final model. With a view to mine sound Petri nets from incomplete event logs (in other words, logs that do not contain all the executed processes), Tapia-Flores et al. [40] propounded a technique called CoMiner. This method is based on continuous occurrence between activities that generate a set of activities termed conjoint occurrence classes [40]. These conjoint occurrence classes are then used to understand the causal and coexisting behaviors (to be specific, process executions) not seen in the event log.

BPMN Models

A state-of-the-art miner that aims at generating BPMN models from an event log using five different phases is called *Split miner* [22]. In the first phase, it creates a directly-follows graph (DFG) [55]. The DFG is not immediately filtered by the split miner but assesses it to discover self-loops and short loops [22]. A loop describes the repetition of an activity [15]. As an example, a self-loop can be recognized because it has an arc to itself. Because these loops are known to create problems in the DFG method, they are removed and only re-instated at the end to produce BPMN models. In the second phase, the split miner detects the concurrency relation between two pairs of activities. This concurrency relation shows up as two arcs. For example, one arc from activity a to activity b and another arc from activity b to activity a, resulting in a conflict between causality and concurrency [15]. To address this conflict, the corresponding arcs are removed from the DFG, and it is termed *pruned DFG* (PDFG). The PDFG consists of the third phase. In the third phase, the PDFG adopts a special filter algorithm. This filtering algorithm guarantees a balance between fitness and precision (see Section 2.2 for the explanation of fitness and precision) while preserving a low control-flow complexity. Split gateways and join gateways are discovered in the fourth and fifth phases to generate a BPMN model from the DFG.

On the one hand, a split gateway has one incoming arc and several outgoing arcs. On the other hand, a join gateway has several incoming arcs and only one outgoing arc. In conclusion, the split miner is powerful compared to other miners as it generates sound BPMN models with no deadlocks [56]. A deadlock is a situation where by a combination of incompatible splits gateways and join gateways prevent the process from progressing [57].

Causal net Models

Different approaches have been developed to mine causal nets from event logs. *CNMining* [48] is a method for generating causal nets. The central concept behind this approach is that it uses information from the event log in connection with previous knowledge as constraints over the structure of the model in order to discover a Causal net [54]. To discover Causal nets composed of an ideal set of patterns capable of covering all the traces [15] by exclusively making use of event types [54], the Maximal Pattern Mining was developed by Liesaputra et al. [50]. This method guarantees the soundness of the mined control flow models and it addresses the problem of noise in event logs by employing user specified threshold for the frequency of event [54]. Noise is a situation whereby the event log contains exceptional behavior (that is, exceptional process executions) that should not be spontaneously integrated in the model [15].

Process Tree Models

A whole family of approaches (inductive miner) exists to mine a process tree from an event log. The inductive miner family uses frequency-based heuristics [54] to generate process trees from an event log. To address the problem of infrequent behavior found in event logs, the inductive miner infrequent [23] was developed. It employs a frequency-based approach to address the problem of infrequent behavior found in event logs, and it guarantees the soundness (at least 80%) of the mined control-flow model [54]. A control-flow model is sound if it is free from certain anomalies [58]. Apart from the family of inductive miners, the evolutionary tree miner (ETM) [59] is another approach to discover process trees. ETM as a genetic algorithm [60], starts by producing a population of irregular process trees. ETM calculates the total fitness value for every tree in the population and applies changes to a subset of that population for every iteration. The miner undergoes several iterations until a stop criterion is satisfied and generates the process tree with the maximum overall fitness.

2.2 Dimensions for Assessing the Quality of Control-Flow Models

According to De Weerdt et al. [25], the quality of control-flow discovery algorithms can be evaluated based on two high-level dimensions (Figure 2): ***accuracy and comprehensibility***.

Accuracy examines how well a control-flow discovery algorithm can generate control-flow models that completely describe the behavior in an event log, thereby finding the balance between underfitting and overfitting [25]. Underfitting is a problem that arises when a model over-generalizes information depicted in the log. That is, more behaviors (additional process executions) are shown even when there are no signs in the event log that suggest this additional behaviors [15]. A model is overfitting if it does not generalize and only reproduces the exact behavior found in the log [15]. Three perspectives make up the overall accuracy of a control-flow model (***fitness, precision, and generalization***) [61].

Comprehensibility as a dimension for evaluating control-flow models assesses the models based on their structuredness and simplicity [25]. For simplicity, Van der Aalst [15] recommends that the Occam's Razor principle [62] should be used to aid PD. In the perspective of PD, simplicity means that the best control-flow model is the simplest model capable of explaining the behavior depicted in the event log [15]. Therefore, simplicity refers to the number of control-flow constructs found in the control-flow model [61]. It is often calculated using the size of the control-flow model [18]. Other simplicity measures are discussed in the study by Lieben et al. [63]. Structuredness describes the magnitude to which models are constructed by merging blocks of the corresponding split and joining gateways [64]. It stipulates that split gateways always have a joining gateway and are merged with each other [64].

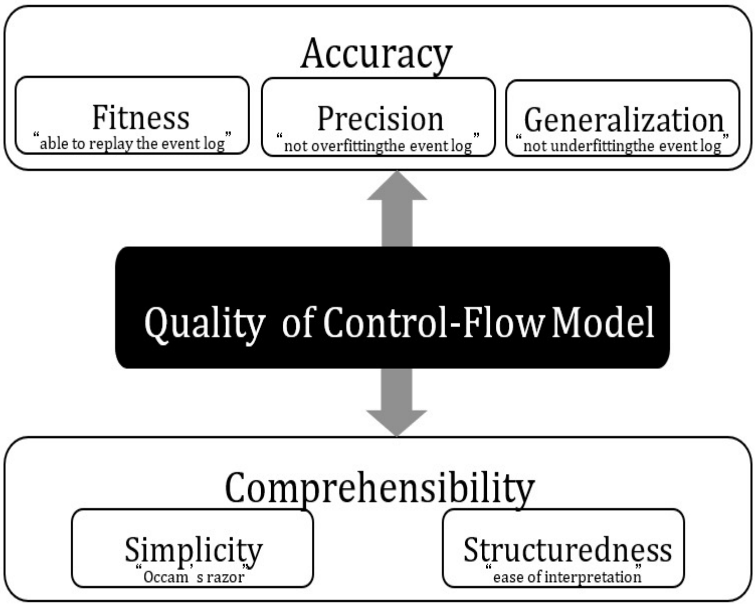


Figure 2: Overview of Quality Dimensions for Assessing Control-Flow Models (Adapted from [25, 61])

As seen in Figure 2, the accuracy of control-flow models can be evaluated based on three dimensions. Firstly, fitness is seen as the fundamental accuracy perspective because it describes how much behavior found in the event log is captured by a model [61]. That is to say, fitness assesses how well the control-flow model can represent the event log [65]. According to De Weerdt et al. [25], it is essential to have models with good fitness since representing the control-flow pattern in an event log is the primary goal of any control-flow discovery algorithm.

Secondly, precision postulates that the control-flow models should control the execution of unseen or unwanted behavior (that is, it should not underfit the data) [61]. In other words, precision measures the proportion of the behavior permitted by the control-flow model which is not observed in the event log [66].

Finally, generalization as a dimension to measure the accuracy of control-flow models has received much debate in the literature. Generalization reckons that a log is a mere sample of the actual behavior of the process system [67]. A system here represents the recorded, the standard, and the actual behavior of a process [67]. According to Maggi et al. [68], generalization assesses how a model is open to behaviors that reflect reality but have not been detected in the system. Nevertheless, a problem exists since the actual behavior of a process is concealed [67]. Janssenswillen et al. [69] identified that the behavior of the process system acts as log with concealed traces. This enables the interaction of the system with the control-flow model by adapting both the fitness and precision metrics to the system in place of the event log [67]. That is, the system's fitness and precision gauge the proportion of the system's behavior covered by the model and the proportion of the model behavior covered by the system, respectively [67].

2.3 Measuring Comprehensibility of Control-Flow Models

To clearly understand how the comprehensibility of control-flow models is measured, there are three points to note: (1) it is important to have a well-defined theoretical concept of the comprehensibility of control-flow models (see Section 2.2), (2) understand the manner in which comprehensibility is operationalized in the literature, and (3) have an idea of empirical studies that employ experimental techniques involving subjects to investigate the effects of a series of factors believed to influence the comprehensibility of control-flow models.

2.3.1 Indicators of Control-Flow Model Comprehensibility

In the field of conceptual modeling, task effectiveness and task efficiency are regular indicators applied to evaluate the comprehensibility of a control-flow model [70, 71]. Conceptual modeling refers to the action of correctly describing some factors of the physical and social world surrounding us for the motives of comprehension and communication [72]. These two indicators (task effectiveness and task efficiency) are the most widely used in the research domain in operationalizing control-flow model comprehensibility [64]. In the literature review by Dikici et al. [64], these indicators are grouped into two clusters (1) two objective indicators for comprehensibility and (2) four subjective (or perceived) indicators for comprehensibility.

The objective indicators of comprehensibility include (1) ***Task Effectiveness Comprehensibility*** and (2) ***Task Efficiency Comprehensibility***. Studies using these indicators provide test subjects with a series of comprehensibility tasks or questions regarding the control flow model [64]. On the one hand, to derive the task effectiveness comprehensibility indicator, the number of right answers (or adequately completed tasks) is divided by the overall number of questions in a comprehensibility task [73]. On the other hand, to derive the task efficiency comprehensibility, the number of correctly answered comprehensibility questions is divided by the time taken to complete the question [64].

The perceived indicators of comprehensibility include (1) ***Mental effort (Cognitive load)***: based on the cognitive load theory, one's cognitive system can cope with only a small number of elements at a particular time [74]. Once the volume of information to be treated surpasses this capacity, comprehensibility is negatively influenced [64]. In addition to the comprehensibility of task effectiveness and efficiency, Zugal et al. [75] propounded the cognitive load indicator to shed more light on the comprehensibility of control-flow models. Cognitive load refers to the mental efforts needed to tackle a problem and is measured by the end-user's rating (a subjective measure) [64]. (2) ***Perceived Ease of Use (for comprehending)***: perceived ease of use is defined by the Technology Acceptance Model (TAM) [76] as the magnitude to which an individual believes using a specific system (in the context of this study, the system is mapped to control-flow models) would be effortless. TAM is characterized by how end-users approve and employ control-flow models [76]. (3) ***Perceived Usefulness***: denotes an individual's subjective likelihood that utilizing a specific control-flow model would improve their job productivity [77]. (4) ***Intention to Use***: is described as the level to which an individual intends to utilize a specific control-flow model [77].

It should be noted that indicators (2), (3), and (4) of subjective indicators of comprehensibility are dependent on TAM. Based on TAM, the perceived usefulness and ease to use are considered to be significant factors of the end-users' intention to use a control-flow model [64]. The general operationalization for these indicators comprises a set of statements with choices to choose from in Likert scale format [64].

2.3.2 Control-Flow Model Comprehensibility Factors

In contrast to the indicators of control-flow model comprehensibility which measure the comprehensibility of control flow models, experiments and surveys have investigated several factors influencing the comprehensibility of control-flow models. Dikici et al. [64] differentiate between two major categories of control-flow comprehensibility factors: (1) factors that are associated with the control-flow model itself and (2) personal factors inherent to the model reader. On the one hand, the model-related factors are characterized by the structure of the control-flow elements and their mutual relationship [78]. Another interesting part of model-related factors is the syntax of control-flow models, which focuses on the graphical representation of the control-flow models [79]. On the other hand, Reijers and Mendling [80] in exploring the factors that influence the comprehensibility of control-flow models, focused their attention on the significant role of personal factors because comprehensibility requires a human interpretation.

Both factors (model-related and personal) are usually considered in most experimental studies as independent (or exogenous) variables [80].

On the one hand, an example of a control-flow model factor that influences the comprehensibility of the model is the visual layout [81]. Research on cognitive psychology has shown that the appearance of a control-flow model significantly influences end-users' comprehension of the control-flow model [82]. As such, the layout of a control-flow model is key to achieving its objectives (effectively describing the supposed process, guaranteeing the comprehension of the model, and facilitating the improvement of the control-flow model) [81]. On the other hand, an example of a personal factor that affects the comprehensibility of a control-flow model is the modeling expertise [83]. According to Bandara et al. [83], modeling expertise refers to the necessary skills, knowledge, and experience possessed by a modeler. As asserted by Petre [84], experts have different views and implement different techniques. In this regard, user characteristics such as individual cognitive talents, styles, and expertise with control-flow models are usually included as independent variables in various research to assess the comprehensibility of control flow models [85, 86].

For a detailed description of all these factors, the reader is referred to Dikici et al. [64] and Figl [87].

2.3.3 Empirical studies on Measuring the Comprehensibility of Control-Flow Models

This section presents some related studies that have measured and quantified the comprehensibility of control-flow models in terms of their modeling notation (model factor). Studies that also evaluate the comprehensibility of control-flow models with respect to some personal factors (knowledge of process modeling notation and professional domain) are equally presented.

In the existence of diverse process modeling notations, for example, Event-Driven Process (EPC) [88], BPMN [32], Unified Modeling Language-Activity Diagram (UML-AD) [89], Unified Modeling Language Class Diagram (UML-CD) [89] and Petri nets [15], it cannot be denied that some of these notations are more practical to convey information to individuals than others [80]. A couple of studies have compared notations in parallel with respect to their comprehensibility. Sarshar and Loos [90] assessed EPC and Petri nets, focusing their experiment on representing the control-flow aspects of the models. They employed 50 students with business and economy education for the evaluation of the two control-flow models with respect to their comprehensibility. The control-flow model comprehensibility was measured using task effectiveness comprehensibility, perceived ease of use, and intention to use. All the dependent variables measured sided with EPC notation. Nonetheless, the work acknowledges the chosen subjects in the experiment as a limitation since students' enthusiasm and learning approach might not be representative of the population of the actual users of the process models in practice.

When comparing BPMN with EPC with respect to their comprehensibility, Recker and Dreiling [91] employed 69 post-graduate students in Information Systems. The students were separated into two groups. One group received a control flow model presented in a notation they were familiar with (i.e., EPC) and the other group received a control flow model presented in an unknown notation (i.e., BPMN), with the expectation that task effectiveness comprehensibility, task efficiency comprehensibility and perceived ease of use will be higher for the EPC group. They found out that participants presented with the BPMN notation achieved similar results as the EPC group in the experiment. Hence, they arrived at the conclusion that end-users with training in any modeling notation perform well in comprehending other process models. Hence, they arrived at the conclusion that BPMN is intuitive as students performed equally well with BPMN (which the did not know) as with EPC (which they did know).

Another empirical study carried out by Purchase et al. [92], is based on the comparison between five distinct notational variations for UML class diagrams . The aim of this experiment was to evaluate whether there was any difference in comprehensibility between the distinct notational variations. The authors recruited 34 students in their second and third year studies of computer science and information systems at the University of Queensland. Five domain experts who were workers of the Distribution Systems Technology Center in Brisbane were equally recruited for the experiment. The students were paid 15 dollars for their time as a motivation to concentrate on the experiment. The domain experts participated as volunteers. The participants were asked to perform comprehension tasks (task effectiveness, task efficiency, and intention to use) on a series of equivalent UML class diagrams which vary according to the notations adopted. The results from this experiment show a significant effect of notational variations on task effectiveness comprehensibility, task efficiency comprehensibility, and intention to use. The authors concluded that the best performing notation could depend on the task for which it is employed.

Looking at the studies above, it is worthy to note that there is a lack of benchmark studies on the comprehensibility of control-flow models. Moreover, there seems to be little to no study that have investigated the comprehensibility of Process tree and Causal net notations. In this study, the comprehensibility of Petri net, causal net, and process tree notations will be evaluated. As such, this study will contribute to the literature on the comprehensibility of control-flow models.

With respect to personal factors influencing the comprehensibility of control-flow models, the knowledge on process modeling is regarded as one of the factors. This factor describes an individual's theoretical knowledge of the overall process modeling notations and specific modeling notations used Dikici et al. [64]. A well-known procedure to quantify this factor is to ask experimental subjects to evaluate their theoretical knowledge (as in Reijers and Mendling [80]). Nonetheless, some experiments use more trustworthy procedures and give experimental subjects a short test to assess their theoretical knowledge of modeling and notation (as in Mendling et al. [86]). In their study, Mendling et al. [86] employed a test with 12 questions to assess participants' level of theoretical knowledge. The study revealed a positive effect of knowledge on modeling and notation on both task effectiveness comprehensibility and task efficiency comprehensibility.

Another personal factor considered to influence the comprehensibility of control-flow models is the professional background. In literature on control-flow model comprehensibility, professional background covers a broad scope, generally employed to characterize participants' working or educational domain [64]. Reijers and Mendling [80] assessed this factor as a categorical variable describing the home institution of the participants. The experiment revealed that students with diverse backgrounds obtain significantly different results with respect to control-flow model comprehensibility. Nonetheless, the authors admit that the participants took distinct process modeling courses.

3 Methodology

The objective of this study is to assess and compare the control-flow models of three different control flow discovery algorithms concerning their perceived comprehensibility. To the best of my knowledge, there seems to be little to no study (with the exception of Petri net) that have investigated the comprehensibility of Process tree and Causal net notations. In this regard, this is an exploratory study. This section is structured as follows: Section 3.1 describes the experimental design while Section 3.2 presents the method for statistical analysis.

3.1 Experimental Design

The goal of this experiment was to find out how end users evaluate the control flow models mined from the sepsis event log with respect to their perceived comprehensibility.

3.1.1 Chosen Algorithms

Evaluating all the miners explained in the literature review would not be feasible due to the diverse nature of inputs required and outputs produced. Three miners (Alpha miner, Heuristics miner, and Inductive miner) were selected. These miners produce three distinct control-flow models (Petri net, Causal net, and Process tree). As such, they satisfy the goal of this experiment.

Alpha (α Miner)

Since this experiment aimed at comparing control-flow models with respect to their comprehensibility, of which one of the dimensions of comprehensibility is simplicity, as described in Section 2.2, the α miner [93] was chosen as one of the miners. According to Buijs et al. [66], the family of control-flow discovery miners that focuses on the simplicity of the mined models is the family of alpha miners [94–96]. These methods generally produce simple control-flow models, although, they produce low fitness and precision values [66]. The α miner is one of the most known control-flow discovery algorithms that was proposed by Van der Aalst et al. [93]. The output of this miner is a Petri net (Figure 3). According to Van der Aalst [15], a Petri net is composed of places that can be marked by tokens. A token is a black dot found in a place. In between two places, there exist transitions. Transitions facilitate the change of a place. Transitions (activities) and places are linked by arcs constituting the flow of activities.

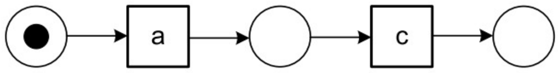


Figure 3: A Petri net model [15]

Besides the α miner, the Heuristic miner [97] and the Inductive miner [23] were selected. These miners are widely used on real-life healthcare event log because they can cope with noisy and incomplete data [19]. Real-life healthcare event logs usually face some data quality problems (e.g., missing events) [4]. Since the control-flow models were mined from the sepsis event log, a healthcare data, it was reasonable to select them. Moreover, the α miner forms the basis of the development of the inductive miner and heuristics miner [15]. Thus, it was also reasonable compare their outputs.

Heuristics Miner (*HM*)

Weijters et al. [97] developed the HM. HM is an extension over the α miner [15]. HM differs from the α Miner in that it can cope with noise and incompleteness of the event log [15]. It mines a process model for the provided event log in the form of a Causal net (Figure 4) [15]. A Causal net (or C-net) is a chart with nodes as activities and arcs as causal dependencies [18],

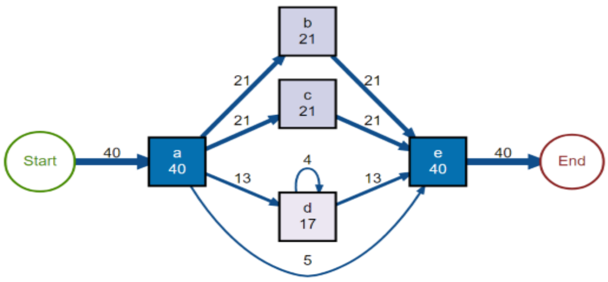


Figure 4: A Causal net Model; Source: `https://cran.r-project.org/web/packages/heuristicsmineR/readme/README.html`

Induvtive Miner (IM)

IM [23] is an extension over the α miner and HM [15]. Apart from facilitating the exploration of an event log; capable of coping with infrequent behavior and noise, it equally generates sound models [15]. The output of this miner is a process tree (Figure 5). A process tree is a hierarchical control-flow model where the nodes represent operators such as sequence and choice, and the leaves represent activities [15]

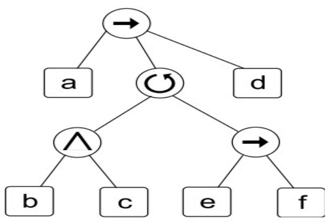


Figure 5: A Process tree Model [15]

3.1.2 Event Log

The event log used in this experiment was from a Dutch hospital originating from the (4TU. Center for Research Data, 2020). It is a real-life event log that contains events of sepsis cases from a hospital. The dataset includes 1050 patients generating a total of 15214 events that were recorded for 16 different hospital activities, with each case reflecting a patient's journey through the hospital [29]. The event log covers the time period from November 7, 2013 to June 5, 2015 [29].

Figure 6 shows a sample of the process variants in the event log (the y-axis represents the traces and the x-axis represents the order of actvities). According to [98], a process variant is a distinct path from the beginning to the end of the process. A process variant can be seen as a particular order of activity, like a "trace" [15] within the process, from the start to the end. In the context of the sepsis event log, a process variant is a specific activity sequence followed by a single patient or by multiple patients. The frequency of patients taking a specific path in the hospital is important because it reveals the frequent patterns in the process. Based on Figure 6, we can count 34 process variants (i.e., the most frequent paths made up of 20% of the log), with a maximum of 14 activities and a minimum of 3 activities followed by a specific patient or multiple patients.

A motivation for using this event log was because it has been used in other unrelated studies [99, 100] before and meets the characteristics [1] for control flow discovery to take place.

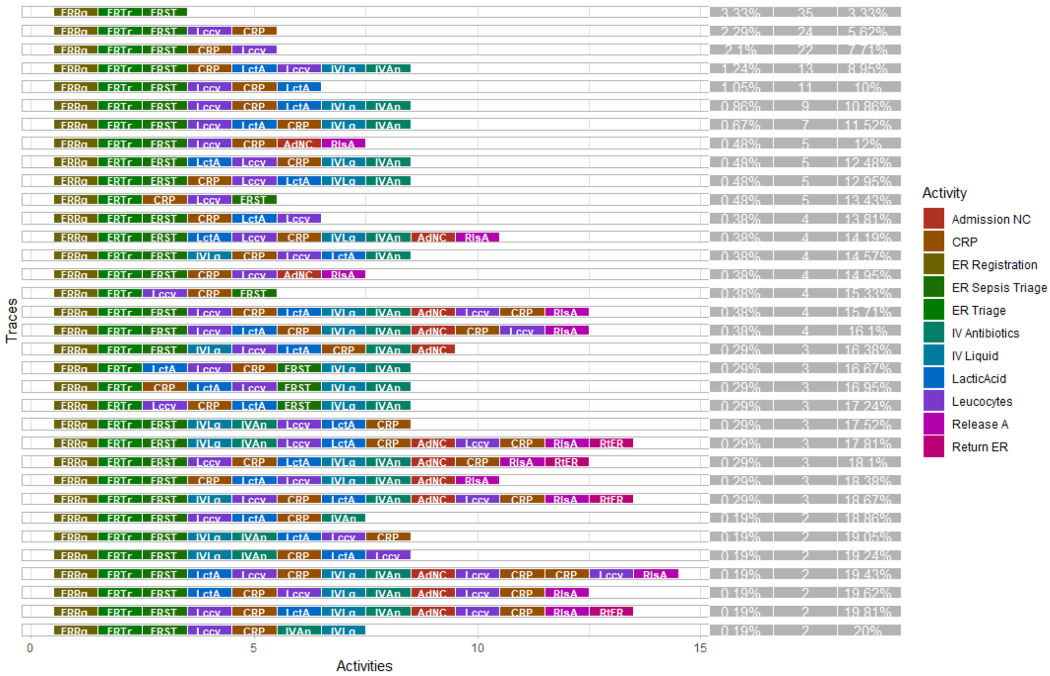


Figure 6: Process Variant of the sepsis event log (generated in bupaR)

3.1.3 Chosen Tool

bupaR [101] and PM4Py framework [102] were selected as tools for this study. As process mining evolves, statistics and data science techniques are beginning to support its improvement. Process mining tools such as PM4PY and bupaR are becoming popular. PM4PY is a library for process mining found in Python. bupaR is a process mining ecosystem that aids process mining in the statistical language R [103]. Compared to other academic tools used for process mining such as PROM, RAPIdProm, etc., bupaR and PM4PY ecosystems can (1) easily incorporate process mining techniques, and data science techniques, (2) bridge the barrier for algorithmic expansion and personalization for process mining techniques, and (3) can allow practitioners and researchers to easily share their work with process mining enthusiasts [104].

Firstly, bupaR was employed to generate the Causal net miner. For an overview of the bupaR packages, the reader is referred to Janssenswillen and Depaire [105]. Secondly, PM4PY was employed to generate Petri net miner and Process tree miner. Using the PM4PY framework as an additional tool was because it was impossible to mine a process tree in bupaR, as the implementation is currently unavailable.

3.1.4 Hypotheses Formulation

Knowing and stating in a clear and formal manner what we seek to examine in an experiment is a crucial part of experiments. Based on the research question and motivated by the study of Sarshar and Loos [90], who assessed the comprehensibility of EPC and Petri net, the following hypotheses were formulated:

Hypothesis 1 (H1): *The end-users perceived ease of use of the control flow is higher for one of the control-flow models.*

An important difference between the three control-flow models is the order in which activities are performed (i.e., their control-flow representation). The control-flow models being compared use different combinations of elements to represent their control-flow. We hypothesized that the perceived ease of use of the control flow of one model will be higher based on the control flow representation.

Hypothesis 2 (H2): *The end-users perceived ease of use of one of the notation is higher.*

If the perceived ease of use of a control-flow is higher for a particular control-flow model, then the perceived ease of use of notation of that model will equally be higher. There exist differences among the three control flow models in terms of their constructs and rules for combining them. This is expected to influence the mental effort of the end-users, rendering it easier (or difficult) to use the notation (s) with fewer (or higher) constructs and simple (or difficult) rules.

Hypothesis 3 (H3): *The end-users intention to use one of the notation is higher.*

Hypothesis 3 was formulated based on the assertion that the end-users' intention to use a notation is influenced by the end users' perceived ease of use of that notation [76]. Regarding this, if the end-users' perceived ease of use of a notation is higher for one of the control-flow models, then the end-users intention to use that particular notation will be higher.

Hypothesis 4 (H4): *There is a significant difference in the end users' overall perceived comprehension of the control-flow models.*

This hypothesis was formulated to test the overall perceived comprehensibility of the control-flow models. Perceived comprehensibility can be seen as an umbrella of the perceived ease of use of a notation and the intention to use a notation [64]. The perceived ease of use and intention to use all map to the user acceptance of a notation defined by TAM [76]. Hence, if the end-users' perceived ease of use of a notation and intention to use a notation are higher, we expect that the overall perceived comprehensibility of that notation will be significantly different from the others.

To test the above hypotheses, an experiment was conducted with students. Based on previous studies [90, 106] three dependent (perception based or subjective) variables and one independent variable were identified. Because of free will or intentionality in human behavior, perceptions have a significantly larger influence on judgments about whether to utilize a certain method (control-flow model) [107]. According to TAM, whether or not individuals will adopt a control-flow model is determined by how effective they believe it is. Hence, for evaluating the models, the following three perception-based variables were established:

- D1: Perceived Comprehensibility.
- D2: Perceived ease of use.
- D3: intention to use.

D1 was measured using the subjects' ratings, using a number from 1 to 7 (See appendix, Question 4). D2 and D3 were measured through statements relating to the control-flow models whereby the end-users gave subjective evaluations on a 4 point Likert scale (see appendix, Task statements 1 to 3). The Likert scale is now one of the most widely used and trusted methods of assessing perceptions, attitudes, and opinions. It allows respondents to express their feelings by selecting one of the available response alternatives [108]. A motivation for choosing the 4 point Likert scale was because we wanted to force the opinion of the end-users without being neutral. Hence, the level of importance is represented using a 4 point Likert scale (Figure 7)

The only independent (or exogenous) variable in this experiment was the type of process modeling notation.

3.1.5 Procedure

This experiment was carried out in three steps. Firstly, the models were mined from the raw log and were labeled by letters (e.g., Petri net= Model A, Causal net= Model B, and Process tree= Model C) as descriptors (See Appendix-Mined Control-Flow Models). The motivation for naming the models with letters was to eradicate the potential impact of domain knowledge.

An explanatory table of the key elements of the tested notations was attached to each control-flow model (See Appendix-Mined Control-Flow Models). This was designed to teach participants the meaning of each symbol and covered everything they needed to know to complete the experimental tasks. Secondly, Section 3.1.6 was carried out using an online tool (Qualtrics). This option was chosen due to the fact that it was not feasible to mobilize the targeted participants together and also more convenient for the participants, because they could fill out the questionnaire whenever it suited them. It was also possible to randomized the models and questions to avoid “learning effects."

Finally, a pilot test was carried out with a different group of students prior to the actual survey in order to learn and further fine tune the task questions. A pilot test inquires whether something can be done to improve the experiment and, if so, how the researcher should proceed. It is mostly conducted with a different group of subjects who are not part of the real experiment [109]. Based on this pilot study, it was observed that the students could not complete all the tasks. They complained it was time-consuming, resulting in a decrease in motivation as they had to evaluate three different control-flow models. The insights gathered from this pilot study led to a reduction in the number of comprehensibility tasks. Objective tasks were deleted from the experiment as they required more mental effort than the subjective tasks. For the final study, participation was voluntary and participants were allowed to fill out the survey within one week. A reminder to carry out Section 3.1.6 was sent to the participants after 4 days. Finally, the data was retrieved from Qualtrics as .CSV file for analysis.

3.1.6 Experimental Task

The experimental task was divided into two parts. Part A (see Appendix) of the online survey was to obtain information about the subjects knowledge and experience with process modeling.

Part B (see Appendix) of the online survey aimed at subjectively assessing the knowledge of the subjects regarding the three control flow models. For the 3 models, subjective questions/statements were identified [27, 80, 90] relating to the dependent variables. Participants were asked to use the experimental process models provided and respond to a question on the survey assessing their perceived comprehension based on their ratings (e.g., How do you assess process model A? [27, 80]). Participants were equally asked to give their opinions whether they agreed or disagreed with specific statements with respect to the perceived ease of use of and the intention to use the control-flow model (for example, I had to think about a decision situation for a long time before I could understand which path (or paths in the process is (or are to be) pursued further [90]). The questions/Likert scale statements used in this experiment were based on previous studies [27, 30, 90]. The Likert scale statements (Task B- Statements 1 to 3 of appendix Task Statements/Questions) came from [90] . Question 4 of the questionnaire came from Mendling and Strembeck [30] and Maggi et al. [27].

3.1.7 Participants

Convenience sampling was used in selecting the participants for this experiment. A convenience sample is a non-probability sample in which the researcher uses the participants who are closest to them and available to take part in the study [110]. The participants were 19 students following the master’s program in Business Process Management at Hasselt University, Belgium. The participants were chosen as the target audience because they were/had taken the course on business process modeling. Another reason for selecting the participants was because they will be the future users of business process models.

The participants were adequately informed about the purpose of the study, which was my master's thesis dissertation, and that their participation would significantly contribute to the completion of my thesis. Clear instructions were given about the specific tasks: (1) that they will receive three control-flow models discovered from a real-life event log. For each model, they will be asked to judge several statements related to the comprehensibility of the model and (2) participation is voluntary, and responses are anonymous.

3.2 Statistical Analysis Approach

To investigate hypotheses 1, 2 and 3, the procedure used in Sarshar and Loos [90] was followed. That is, conclusions were based on the mode (frequency) of students who responded to the 4-point Likert-type scale statements from Section 3.1.6. Since the variables used to verify these hypotheses were based on TAM, our aim was to get the number of participants that agreed or disagreed with a particular statement. The aim was not to test the significance as in hypothesis 4. If we were interested to test the significance, we would have adopted the procedure described for hypothesis 4.

Figure 7: Operationalzation of the 4 Point Likert-Scale

If there was more than one statement to gauge a variable (for example, hypothesis 1 and hypothesis 3), the frequencies of students who responded to the substatements were divided by the total number of statements and then summed to reflect the overall frequency of students for the respective variable. This exact procedure was employed by [27].

In contrast to hypotheses 1, 2 and 3, hypothesis 4 aimed to discover if there was a significant difference across the control-flow models with respect to their perceived comprehension. To test this hypothesis, a procedure described in Demšar [111] was followed. The Friedman test [112] was conducted which is a non-parametric equivalent of the Analysis of Variance (ANOVA) [113], a parametric test. The motivation for using the Friedman test over the ANOVA test was because participants had to give ratings, which are examples of ordinal scale measurement, rendering the data unsuitable for a parametric test [114]. Another motivation for using this test was due to the small sample size of our study. The Friedman test statistic is defined as:

$$\chi_F^2 = \frac{12P}{K(K+1)}\left[\sum_{j=1}^{k} R_j^2 - \frac{K(K+1)^2}{4}\right]$$

Where:

- R_j is the mean rank of the algorithms (in the context of this study, we map it to average rank of the perceived comprehensibility of the control flow models) from j = 1, 2 . . . k over P data sets.
- K is the number of algorithms (in the context of this study, we map it to the number control-flow models being compared) and P is the number of dataset (which maps to the number of participants in the context of this study).

Under the null hypothesis (There is no significance difference in the end-users overall perceived comprehensibility of the control flow models), the Friedman test statistic follows a χ^2 distribution with K-1 degrees of freedom when P and K are large enough (as a rule of thumb, P>10 and K>5). Suppose the null hypothesis is rejected by the Friedman test, all the pairs of control flow models will be compared using a Nemenyi Post-hoc tests [115]. The Nemenyi test is a pos-hoc analysis that aims to determine the difference among the pair of groups (in this study, we map the pair of groups to the control-flow models) after the null hypothesis has been rejected by the Friedman's test. The Nemenyi test then conducts a pairwise test to determine which control-flow models differ. The analysis was carried out in R studio.

4 Results

This section describes the results of the assessment. Firstly, the results from the demographic data obtained from the survey are presented. Secondly, a summary of the hypotheses results obtained from the analysis is presented. The real names of the outputs of the control-flow models are used: Model A (Petri net), Model B (Causal net), and Model C (Process tree).

4.1 Subjects' Demographic Data

In this subsection, the summary of the participants is presented in Table 2.

Table 2: Summary of Participants

Number of Students	Took part in the experiment	Completed	Partially completed
	19	15	4
	100%	78.95%	21.05%
Age	Minimum	Average	Maximum
	23	29	40
Gender	Male	Female	
	04	11	
	26.67%	73.33%	
Education	Degree	Number	Specialization
	Bachelor	11 (73.33%)	Economic Field
	Master	3 (20%)	/
	PhD	1 (6.67%)	Environmental Chemistry
Semesters completed	Minimum	Average	Maximum
	1	3	6
Theoretical Knowledge	Weak	Average	Strong
	2	9	4
	13.33%	60%	26.67%
Practical Knowledge	Yes	No	
	8	7	
	53.33%	46.67%	
Knowledge of other Modeling Notation	None		

Among the 19 participants who opted to participate in the survey, 15 students (78.95%) completed the entire survey. For further analysis, the data of the other four students were discarded due to the incompleteness of the task. The average age of the participants was 29, the minimum 23, and the maximum 40. 11 students (73.33%) had bachelor's degrees in an economic field (economics, accounting, management, entrepreneurship, and marketing). Three students (20%) had master's degrees, and one had a Ph.D. in environmental chemistry.

With regards to the theoretical knowledge of process modeling, 9 participants (60%) had average theoretical knowledge, 4 participants (26.67%) had a strong theoretical knowledge, and 2 participants (13.33%) had a weak theoretical knowledge. 8 participants (53.33%) have practical knowledge of business process modeling. Finally, no student could name any other process modeling language other than BPMN.

4.2 Hypotheses Testing

This section presents the results for the hypotheses investigations.

4.2.1 Hypothesis 1

Table 3: **The end-users perceived ease of use of the control flow is higher for one of the control-flow models.**

	Petri net model		Causal net model		Process tree model	
	Frequency of Students	Percent	Frequency of Students	Percent	Frequency of Students	Percent
Strongly Agree	5	33.33	5	33.33	3	20
Agree	6	40	5	33.33	6	40
Disagree	3	20	2	13.33	4	26.67
Strongly Disagree	1	6.67	3	20	2	13.33
Total	15	100	15	100	15	100

Table 3 is the summary results for hypothesis 1, measured through statements 1.1-1.3 (See Task B of Appendix C). Based on the 4 Likert Scale operationalization in Figure 7, we gauge statements 1.1-1.3 of the questionnaire based on the orange region in the table. From the table, the perceived ease of use of the control flow of the Process tree is higher than that of Causal net and Petri net. That is 6 students (40%) perceived the control-flow of the Process tree model as the easiest to comprehend compared to 4 students (26.67%) for the Petri net model and 5 students (33.33%) for the Causal net model.

4.2.2 Hypothesis 2

Table 4: **The end-users perceived ease of use of one of the notation is higher.**

	Petri net model		Causal net model		Process tree model	
	Frequency of Students	Percent	Frequency of Students	Percent	Frequency Of Students	Percent
Strongly Agree	5	33.33	5	33.33	5	33.33
Agree	5	33.33	3	20	5	33.33
Disagree	4	26.67	4	26.67	4	26.67
Strongly Disagree	1	6.67	3	20	1	6.67
Total	15	100	15	100	15	100

Table 4 presents the summary results for hypothesis 2. This was gauged by statement 2 (See Task B of Appendix C). Based on Figure 7, we gauge statement 2 of the questionnaire based on the orange region in the table. As depicted in the Table 4, 5 students (33.34%) each found both the Petri net and Process tree notations to be unsuitable for presenting processes in a comprehensible manner. Compared to both the Petri net and the Process tree notations, 7 students (46.67 %) found the Causal net notation suitable for presenting business processes in a comprehensible manner.

4.2.3 Hypothesis 3

Table 5: **The end-users intention to use one of the notation is higher.**

	Petri net model		Causal net model		Process tree model	
	Frequency of Students	Percent	Frequency of Students	Percent	Frequency of Students	Percent
Strongly Agree	4	26.67	3	20	3	20
Agree	6	40	7	46.67	7	46.67
Disagree	5	33.33	4	26.67	4	26.67
Strongly Agree	0	00	1	6.67	1	6.67
Total	15	100	15	100	15	100

Table 5 presents the summary results for hypothesis 3. This was measured by statements 3.1-3.2 (See Task B of Appendix C). Based on Figure 7, we gauge statements 3.1-3.2 of the questionnaire based on the green region in the table. As we can see from the table, the students' intentions to use all the notations were equal (i.e., 10 students (66.67%) across all three notations).

4.2.4 Hypothesis 4

Table 6: **There is a significant difference in the end users' overall perceived comprehension of the control-flow model.**

Friedman rank sum test	
Friedman Chi-square (χ^2_F)	5.4522
Degree of freedom (df)	2
P.Value	0.244
Alpha value (significance level)	0.01

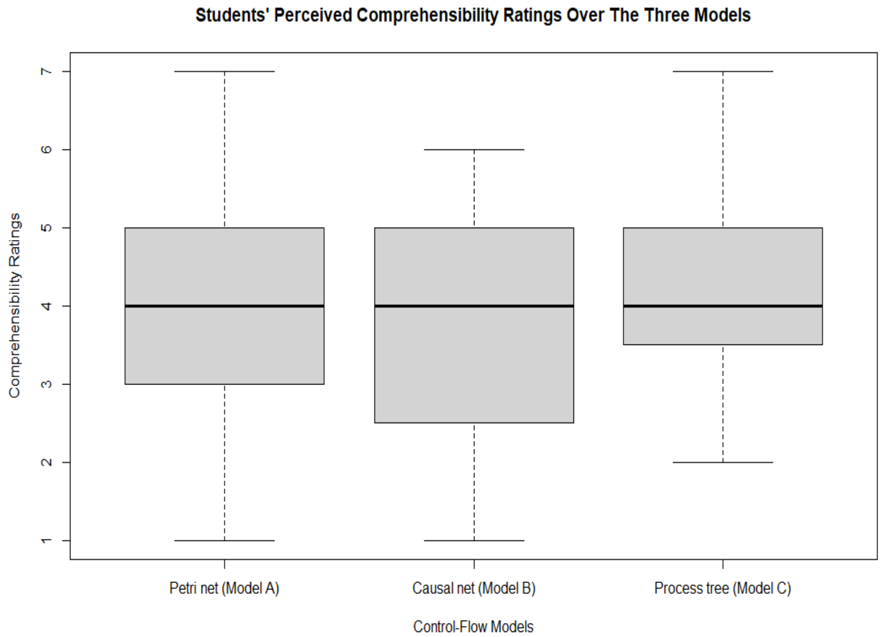


Figure 8: Box plot for Perceived Comprehensibility of Process Models

Table 6 shows the summary statistics of the Friedman's test and Figure 8 is the box plot showing the median perceived comprehensibility ratings across all three control-flow models. The y-axis in the box plot represents students' ratings on the comprehensibility of the control flow models while the x-axis represents the various control-flow models. Question 4 (See Task B of Appendix C) was used to investigate this hypothesis. In statistics, a box plot is a diagnostic tool for the data in a box displaying the median [116]. Both the box plot and the Friedman's test were used to test hypothesis 4.

Firstly, to evaluate whether the models differ significantly from each other with respect to their perceived comprehensibility, a box plot (Figure8) was used to investigate the difference across models with respect to the end users' overall perceived comprehension. The box plot shows no difference in the median ratings of the models (the median perceived comprehensibility rating across all three control-flow models is 4).

Secondly, the Friedman's test was employed to further verify if the control-flow models were significantly different from each other with respect to their perceived comprehensibility. The Friedman test statistic (5.4522) and the degrees of freedom (2) are reported in Table 6. The test is insignificant (χ^2_F= 5.4522, P-Value= 0.244 > 0.001). In statistics, the P-value is a measure used in assessing a hypothesis based on the observed data [116]. Once the P-value is greater than the significance level (i.e., the threshold value used to evaluate the statistical significance of a test statistic), we don't reject the null hypothesis given the data (i.e., there is no significance difference in the end users' overall perceived comprehension of the control-flow models) and vice versa. In Table 6, the P-value is 0.244 and the significance value (or alpha value) = 0.001. Since the P-Value (0.244) > significance level (0.001), there is insufficient proof to reject the null hypothesis.

5 Discussion of Results and Limitations

This section presents the discussion and explanation of the results. Section 5.1 discusses the results of the hypothesis and Section 5.2 discusses the limitations of the study.

5.1 Discussion of results

As highlighted in Section 1, Comprehensibility is vital if an organization (e.g., healthcare organization) would like to use the control-flow model for process improvement, optimization, and communication purposes. This study evaluated the perceived comprehensibility of three control-flow models expressed in Petri net, Causal net, and Process tree. As such, four hypotheses were investigated. To the best of my knowledge, this is the first experiment to evaluate the comprehensibility of the Petri net, Causal net, and Process tree notations jointly. As a result, there is a lack of benchmark studies to which the findings of this study can be compared to.

Firstly, based on the results of hypothesis 1 displayed in Table 3, we can see that the students perceived ease of use of the control flow of the process tree model is higher than that of the Causal net and the Petri net models. This result implies that the students perceived the control-flow representation of the Process tree more comprehensible to use for explaining processes. That is, the students were able to comprehend the flow of activities and could discern which paths in the process are to be pursued further. The reason for this result might be because the representation (or hierarchy) of the process tree permits one to easily comprehend the essential control flow of the process [117]. In a study by van Zelst and Leemans [117], the authors used a real-life event log and showed the relative simplicity with which an individual can infer the executed processes of a Process tree. Moreover, the Process tree is free from deadlocks, thereby facilitating the ease of use of the control flow.

Secondly, the result of hypothesis 2 presented in Table 4, shows that the students perceived ease of use of the Causal net notation is higher than that of the Process tree and the Petri net notations. Hypothesis 2 is based on the assumption that if the perceived ease of use of the control-flow is higher in a particular control-flow model, then the perceived ease of use of the notation will be higher. However, the result of hypothesis 2 contradicts this assumption. In hypothesis 1, the control-flow of the Process tree model was perceived by the students as the easiest to use. In hypothesis 2, the perceived ease of use of the Process tree notation was expected to be higher. A possible reason for this result might be because Causal nets provide a better representational bias for control-flow discovery than other conventional modeling notations (in this study, Petri net and Process tree) [58]. Moreover, It is put forward that participants in the first instance seek the start of a control-flow model [56]. According to Bühler [56], those modeling notations with a start and end symbol are simpler to comprehend. Hence, the Causal net notation used in this study satisfies this assertion.

Thirdly, hypothesis 3 was formulated based on the assertion that the end users' intention to use a particular notation is influenced by the perceived ease of use of that notation [76]. Based on this assertion, one would have expected that the students' intention to use the Causal net notation will be higher than that of the Petri net and Process tree notations. However, Table5 shows that the end users' intention to use all three notations are equal. This result implies that the students find the three notations more appealing to use for explaining healthcare processes.

Finally, hypothesis 4 intended to find out if there was a significant difference in the overall perceived comprehensibility of the control-flow models. This hypothesis was investigated using Figure 8 and later verified using Friedman's test statistic displayed in Table 6. One would have anticipated significant differences across the three control-flow models based on hypotheses 1 and 2. However, the box plot and Friendman's test show no significant difference across the three models concerning their perceived comprehensibility. In other words, it doesn't matter which control-flow model they use to present healthcare processes in a comprehensible manner. Based on the data used in this study, one can say that the students' perceived comprehensibility of a control-flow model is influenced by the students' intention to use the notation. A possible reason for this result might be due to the participants' professional backgrounds. Their professional backgrounds could be a factor that caused them not to perceive a difference across the control-flow models concerning their comprehensibility. Most of the study's participants have an accounting/business background, with just 26.67% of the overall participants having a strong theoretical knowledge of process modeling. Therefore, it can be argued that the students do not have a higher average capacity to deconstruct these notations (Causal net, Petri net, and Process tree) given their background, as they are more familiar with BPMN.

5.2 Limitations

This study is subject to limitations that might have influenced the obtained results. Firstly, one can state that the study carried out is an exploratory experiment (in the sense that the Process tree and the Causal net notations have not been evaluated before for comprehensibility). The aim has not been to benchmark previous studies but to obtain novel insights. The number of participants employed in this study is a limiting factor, preventing the obtention of meaningful conclusions.

Secondly, the subjects of this experiment are students. The fact that students evaluated control-flow models mined from a real-life healthcare event log strongly influences the generalization of these results. However, all the subjects were students from the master of business process management with some knowledge on process modeling, which is a factor that neutralizes this effect.

Thirdly, the event log used in this study is a healthcare event log. Healthcare event logs possess some variations compared to other business sectors. These variations result from the complexity of the healthcare sector explained in Section 1. To evaluate and compare control-flow models mined from the sepsis event log, it was necessary to do some preprocessing of the log (e.g., filtering some infrequent activities or flows). The preprocessing of this log was not done in this experiment. As such, this might greatly influence the comprehensibility of the control-flow models. Moreover, one of the selected miners (α miner), though appealing because of its simplicity, it is not applicable to real-life event logs because it assumes that the log is free from noise and is sensitive to infrequent behavior. This might negatively influence the comprehensibility of the model it generates.

Lastly, another pertinent limitation of this study relates to the comprehensibility task questions. No content-related questions (objective measures) were asked regarding the models (the reason for this is explained in Section 3.1.5). As asserted by Moody [106], from a scientific standpoint, objective measures provide more convincing evidence than subjective measures. Asking participants content-related questions regarding the control-flow models would provide more evidence of how comprehensible the models actually are.

6 Conclusion

This study aimed at evaluating the comprehensibility of control flow models mined from the sepsis event log. The comprehensibility of control-flow models is very important if an institution (e.g., healthcare institution) would like to use the control-flow model for process improvement purposes, communication purposes, etc. An experiment was conducted to answer the research question of "How are the outputs of control-flow discovery algorithms evaluated by end-users in terms of comprehensibility when applied to the sepsis event log" In this regard, the control-flow models were evaluated and compared by students concerning their perceived comprehensibility (perceived ease of use and intention to use).

The control-flow models (Petri net, Causal net, and Process tree) were mined from the raw log (Sepsis event log). The sepsis event log was fed into two process mining tools (bupaR and PM4PY), where the mined control-flow models were saved. These control-flow models were then assessed for comprehensibility in the end-user study. This study was carried out with 19 students pursuing the master of business process management at Hasselt University. Each participant was presented with the three control-flow models in an online survey via Qualtrics. For each model, the participants had to judge several statements concerning the perceived ease of use and intention to use. They equally had to rate the models with respect to their perceived comprehensibility.

Based on the findings of this study, the end-users perceived the control-flow of the Process tree model as the easiest to comprehend. However, they perceived the notation of the Causal net as the most suitable for presenting healthcare processes in a comprehensible manner. It was defined in this study that the end-users perceived ease of use of a particular notation would influence the intention to use that particular control-flow model. However, this was contradicted, as the end-users find all three control-flow models appealing to use for presenting healthcare processes. With the help of Friedman's test, it was shown that there was no statistical difference across the models with respect to their perceived comprehensibility. Based on the data, we may conclude that the overall perceived comprehensibility of the control-flow models is influenced by the end-users intention to use them for presenting healthcare processes.

Despite these interesting findings, this study is subject to some limitations (especially the number of subjects and the subjective measures of comprehensibilty). Finally, directions for future research are presented to take this investigation further. It is important to explore this study further with more content-related questions on the control-flow models regarding their comprehensibility. Asking content-related questions where participants can answer with a Yes or No (e.g., can ER registration and ER Tirage be executed at the same time?) would provide evidence on how comprehensible the control-flow models actually are. Moreover, to avoid a decrease in motivation from the part of the participants during experiments of this nature, it would be vital to carry out such an experiment with a large sample size consisting of three groups, where each group evaluates a particular control-flow model with respect to its comprehensibility. A more robust statistical test like the ANOVA test can then be employed to investigate the difference in comprehensibility across the models.

References

[1] Niels Martin, Jochen De Weerdt, Carlos Fernández-Llatas, Avigdor Gal, Roberto Gatta, Gema Ibáñez, Owen A. Johnson, Felix Mannhardt, Luis Marco-Ruiz, Steven Mertens, Jorge Munoz-Gama, Fernando Seoane, Jan Vanthienen, Moe Thandar Wynn, David Baltar Boilève, Jochen Bergs, Mieke Joosten-Melis, Stijn Schretlen, and Bram B. Van Acker. Recommendations for enhancing the usability and understandability of process mining in healthcare. *Artificial Intelligence Medicine*, 109:101962, 2020. doi: 10.1016/j.artmed.2020.101962. URL `https://doi.org/10.1016/j.artmed.2020.101962`.

[2] Ronny S Mans, Wil MP Van der Aalst, and Rob JB Vanwersch. *Process mining in healthcare: evaluating and exploiting operational healthcare processes*. Springer, 2015.

[3] Kathrin Kirchner, Nico Herzberg, Andreas Rogge-Solti, and Mathias Weske. Embedding conformance checking in a process intelligence system in hospital environments. In *Process Support and Knowledge Representation in Healthcare*, pages 126–139. Springer, 2012.

[4] Niels Martin. *Data quality in process mining*, pages 53–79. Springer International Publishing, Cham, 2021. ISBN 978-3-030-53993-1. doi: 10.1007/978-3-030-53993-1_5. URL `https://doi.org/10.1007/978-3-030-53993-1_5`.

[5] Marlon Dumas, Marcello La Rosa, Jan Mendling, and Hajo A. Reijers. *Fundamentals of Business Process Management, Second Edition*. Springer, 2018. ISBN 978-3-662-56508-7. doi: 10.1007/978-3-662-56509-4. URL `https://doi.org/10.1007/978-3-662-56509-4`.

[6] Eric Rojas, Jorge Munoz-Gama, Marcos Sepúlveda, and Daniel Capurro. Process mining in healthcare: A literature review. *Journal of Biomedical Informatics*, 61:224–236, 2016. doi: 10.1016/j.jbi.2016.04.007. URL `https://doi.org/10.1016/j.jbi.2016.04.007`.

[7] Marlon Dumas, Wil M Van der Aalst, and Arthur H Ter Hofstede. *Process-aware information systems: bridging people and software through process technology*. John Wiley & Sons, 2005.

[8] Uzay Kaymak, Ronny Mans, Tim Van de Steeg, and Meghan Dierks. On process mining in health care. In *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1859–1864. IEEE, 2012.

[9] Álvaro Rebuge and Diogo R. Ferreira. Business process analysis in healthcare environments: A methodology based on process mining. *Information Systems*, 37(2):99–116, 2012. doi: 10.1016/j.is.2011.01.003. URL `https://doi.org/10.1016/j.is.2011.01.003`.

[10] Eric Rojas, Michael Arias, and Marcos Sepúlveda. Clinical processes and its data, what can we do with them. In *Proceedings of the International Conference on Health Informatics (HEALTHINF 2015), Lisbon, Portugal*, pages 12–15, 2015.

[11] Claudio Di Ciccio, Andrea Marrella, and Alessandro Russo. Knowledge-intensive processes: characteristics, requirements and analysis of contemporary approaches. *Journal on Data Semantics*, 4(1):29–57, 2015.

[12] Steven Mertens, Frederik Gailly, and Geert Poels. Towards a decision-aware declarative process modeling language for knowledge-intensive processes. *Expert Systems Applications*, 87:316–334, 2017. doi: 10.1016/j.eswa.2017.06.024. URL `https://doi.org/10.1016/j.eswa.2017.06.024`.

[13] Richard Lenz, Mor Peleg, and Manfred Reichert. Healthcare process support: achievements, challenges, current research. *International Journal of Knowledge-Based Organizations (IJKBO)*, 2(4), 2012.

[14] Payam Homayounfar. Process mining challenges in hospital information systems. In *2012 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 1135–1140. IEEE, 2012.

[15] Wil MP Van der Aalst. *Process mining: data science in action*. Springer, 2016.

[16] Wil MP Van der Aalst. Process-aware information systems: Lessons to be learned from process mining. In *Transactions on Petri Nets and other Models of Concurrency II*, pages 1–26. Springer, 2009.

[17] Adam Rule, Michael F. Chiang, and Michelle R. Hribar. Using electronic health record audit logs to study clinical activity: a systematic review of aims, measures, and methods. *Journal of American Medical Informatics Association*, 27(3):480–490, 2020. doi: 10.1093/jamia/ocz196. URL `https://doi.org/10.1093/jamia/ocz196`.

[18] Wil M.P Van der Aalst. *Process mining: discovery, conformance and enhancement of business processes*. Springer, Berlin, 2011. ISBN 9783642193446;3642193447;.

[19] Elisabetta Benevento, Prabhakar M Dixit, Mohammadreza Fani Sani, Davide Aloini, and Wil MP van der Aalst. Evaluating the effectiveness of interactive process discovery in healthcare: A case study. In *International Conference on Business Process Management*, pages 508–519. Springer, 2019.

[20] AJMM Weijters and Joel Tiago S Ribeiro. Flexible heuristics miner (fhm). In *2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, pages 310–317. IEEE, 2011.

[21] Christian W Günther and Wil MP Van Der Aalst. Fuzzy mining–adaptive process simplification based on multi-perspective metrics. In *International Conference on Business Process Management*, pages 328–343. Springer, 2007.

[22] Adriano Augusto, Raffaele Conforti, Marlon Dumas, and Marcello La Rosa. Split miner: Discovering accurate and simple business process models from event logs. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 1–10. IEEE, 2017.

[23] Sander JJ Leemans, Dirk Fahland, and Wil MP Van Der Aalst. Discovering block-structured process models from event logs containing infrequent behaviour. In *International Conference on Business Process Management*, pages 66–78. Springer, 2013.

[24] R. S. Mans, Helen Schonenberg, Minseok Song, Wil M. P. van der Aalst, and Piet J. M. Bakker. Application of process mining in healthcare - A case study in a dutch hospital. In Ana L. N. Fred, Joaquim Filipe, and Hugo Gamboa, editors, *Biomedical Engineering Systems and Technologies, International Joint Conference, BIOSTEC 2008, Funchal, Madeira, Portugal, January 28-31, 2008, Revised Selected Papers*, volume 25 of *Communications in Computer and Information Science*, pages 425–438. Springer, 2008. doi: 10.1007/978-3-540-92219-3_32. URL `https://doi.org/10.1007/978-3-540-92219-3_32`.

[25] Jochen De Weerdt, Manu De Backer, Jan Vanthienen, and Bart Baesens. A multi-dimensional quality assessment of state-of-the-art process discovery algorithms using real-life event logs. *Information Systems*, 37(7):654–676, 2012.

[26] Edgar Batista and Agusti Solanas. Skip miner: Towards the simplification of spaghetti-like business process models. In *2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA)*, pages 1–6. IEEE, 2019.

[27] Fabrizio Maria Maggi, Andrea Marrella, Fredrik Milani, Allar Soo, and Silva Kasela. A user evaluation of automated process discovery algorithms. *arXiv preprint arXiv:1806.03150*, 2018.

[28] Jochen De Weerdt, Manu De Backer, Jan Vanthienen, and Bart Baesens. A robust f-measure for evaluating discovered process models. In *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2011, part of the IEEE Symposium Series on Computational Intelligence 2011, April 11-15, 2011, Paris, France*, pages 148–155. IEEE, 2011. doi: 10.1109/CIDM.2011.5949428. URL `https://doi.org/10.1109/CIDM.2011.5949428`.

[29] Felix Mannhardt. Sepsis cases-event log. 4tu. researchdata. dataset, 2016.

[30] Jan Mendling and Mark Strembeck. Influence factors of understanding business process models. In *International Conference on Business Information Systems*, pages 142–153. Springer, 2008.

[31] Dan Russler. Clinical event., 2009.

[32] Richard C Simpson. An xml representation for crew procedures. *NASA Summer Faculty Fellowship Program 2004, Volumes 1 and 2*, 2005.

[33] Adriano Augusto, Raffaele Conforti, Marlon Dumas, Marcello La Rosa, Fabrizio Maria Maggi, Andrea Marrella, Massimo Mecella, and Allar Soo. Automated discovery of process models from event logs: review and benchmark. *IEEE Transactions on Knowledge and Data Engineering*, 31(4):686–705, 2018.

[34] Zan Huang and Akhil Kumar. A study of quality and accuracy trade-offs in process mining. *INFORMS Journal on Computing*, 24(2):311–327, 2012.

[35] Josep Carmona and Jordi Cortadella. Process discovery algorithms using numerical abstract domains. *IEEE Transactions on Knowledge and Data Engineering*, 26(12):3064–3076, 2013.

[36] Wei Song, Hans-Arno Jacobsen, Chunyang Ye, and Xiaoxing Ma. Process discovery from dependence-complete event logs. *IEEE Transactions on Services Computing*, 9(5):714–727, 2015.

[37] Qinlong Guo, Lijie Wen, Jianmin Wang, Zhiqiang Yan, and Philip S Yu. Mining invisible tasks in non-free-choice constructs. In *International Conference on Business Process Management*, pages 109–125. Springer, 2016.

[38] Dominic Breuker, Martin Matzner, Patrick Delfmann, and Jörg Becker. Comprehensible predictive models for business processes. *Mis Quarterly*, 40(4):1009–1034, 2016.

[39] Chuanyi Li, Jidong Ge, Liguo Huang, Haiyang Hu, Budan Wu, Hongji Yang, Hao Hu, and Bin Luo. Process mining with token carried data. *Information Sciences*, 328:558–576, 2016.

[40] Tonatiuh Tapia-Flores, Edelma Rodríguez-Pérez, and Ernesto López-Mellado. Discovering process models from incomplete event logs using conjoint occurrence classes. In *ATAED@ Petri Nets/ACSD*, pages 31–46, 2016.

[41] HMW Verbeek, Wil MP van der Aalst, and Jorge Munoz-Gama. Divide and conquer: a tool framework for supporting decomposed discovery in process mining. *The Computer Journal*, 60(11):1649–1674, 2017.

[42] Sebastiaan J Van Zelst, Boudewijn F van Dongen, Wil MP van der Aalst, and HMW Verbeek. Discovering workflow nets using integer linear programming. *Computing*, 100(5): 529–556, 2018.

[43] David Redlich, Thomas Molka, Wasif Gilani, Gordon S Blair, and Awais Rashid. Scalable dynamic business process discovery with the constructs competition miner. In *SIMPDA*, pages 91–107. Citeseer, 2014.

[44] Thomas Molka, David Redlich, Marc Drobek, Xiao-Jun Zeng, and Wasif Gilani. Diversity guided evolutionary mining of hierarchical process models. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pages 1247–1254, 2015.

[45] Raffaele Conforti, Marlon Dumas, Luciano García-Bañuelos, and Marcello La Rosa. Bpmn miner: Automated discovery of bpmn process models with hierarchical structure. *Information Systems*, 56:284–303, 2016.

[46] Seppe KLM Vanden Broucke and Jochen De Weerdt. Fodina: a robust and flexible heuristic process discovery technique. *Decision Support Systems*, 100:109–118, 2017.

[47] Adriano Augusto, Raffaele Conforti, Marlon Dumas, Marcello La Rosa, and Giorgio Bruno. Automated discovery of structured process models from event logs: the discover-and-structure approach. *Data & Knowledge Engineering*, 117:373–392, 2018.

[48] Gianluigi Greco, Antonella Guzzo, Francesco Lupia, and Luigi Pontieri. Process discovery under precedence constraints. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 9(4):1–39, 2015.

[49] Borja Vázquez-Barreiros, Manuel Mucientes, and Manuel Lama. Prodigen: Mining complete, precise and minimal structure process models with a genetic algorithm. *Information Sciences*, 294:315–333, 2015.

[50] Veronica Liesaputra, Sira Yongchareon, and Sivadon Chaisiri. Efficient process model discovery using maximal pattern mining. In *International Conference on Business Process Management*, pages 441–456. Springer, 2016.

[51] Bernardo Nugroho Yahya, Minseok Song, Hyerim Bae, Sung-ook Sul, and Jei-Zheng Wu. Domain-driven actionable process model discovery. *Computers & Industrial Engineering*, 99:382–400, 2016.

[52] Hoang Nguyen, Marlon Dumas, Arthur HM ter Hofstede, Marcello La Rosa, and Fabrizio Maria Maggi. Mining business process stages from event logs. In *International Conference on Advanced Information Systems Engineering*, pages 577–594. Springer, 2017.

[53] Seppe KLM Vanden Broucke, Jan Vanthienen, and Bart Baesens. Declarative process discovery with evolutionary computing. In *2014 IEEE Congress on Evolutionary Computation (CEC)*, pages 2412–2419. IEEE, 2014.

[54] Allar Soo. Automated process discovery: A literature review and a comparative evaluation with domain experts. *Master Thesis, University of Tartu*, 2017.

[55] Wil M. P. van der Aalst. A practitioner's guide to process mining: Limitations of the directly-follows graph. In Maria Manuela Cruz-Cunha, Ricardo Martinho, Rui Rijo, Emanuel Peres, and Dulce Domingos, editors, *CENTERIS 2019 - International Conference on Enterprise Information Systems / ProjMAN 2019 - International Conference on Project MANagement / HCist 2019 - International Conference on Health and Social Care Information Systems and Technologies 2019, Sousse, Tunisia*, volume 164 of *Procedia Computer Science*, pages 321–328. Elsevier, 2019. doi: 10.1016/j.procs.2019.12.189. URL `https://doi.org/10.1016/j.procs.2019.12.189`.

[56] Jana Bühler. *The empirical analysis of the comprehensibility of process models created by process mining*. PhD thesis, Ulm University, 2021.

[57] Haleh Moattar, Wasana Bandara, Udo Kannengiesser, and Michael Rosemann. Control flow versus communication: comparing two approaches to process modelling. *Business Process Management Journal*, 2022.

[58] Wil M. P. van der Aalst, Arya Adriansyah, and Boudewijn F. van Dongen. Causal nets: A modeling language tailored towards process discovery. In Joost-Pieter Katoen and Barbara König, editors, *CONCUR 2011 - Concurrency Theory - 22nd International Conference, CONCUR 2011, Aachen, Germany, September 6-9, 2011. Proceedings*, volume 6901 of *Lecture Notes in Computer Science*, pages 28–42. Springer, 2011. doi: 10.1007/978-3-642-23217-6_3. URL `https://doi.org/10.1007/978-3-642-23217-6_3`.

[59] Joos CAM Buijs, Boudewijn F van Dongen, and Wil MP van der Aalst. Quality dimensions in process discovery: The importance of fitness, precision, generalization and simplicity. *International Journal of Cooperative Information Systems*, 23(01):1440001, 2014.

[60] Joos CAM Buijs, Boudewijn F van Dongen, and Wil MP van der Aalst. A genetic algorithm for discovering process trees. In *2012 IEEE Congress on Evolutionary Computation*, pages 1–8. IEEE, 2012.

[61] Seppe KLM Vanden Broucke. Advances in process mining: Artificial negative events and other techniques. 2014.

[62] AJM Garrett. Ockham's razor. In *Maximum Entropy and Bayesian Methods*, pages 357–364. Springer, 1991.

[63] Jonas Lieben, Toon Jouck, Benoît Depaire, and Mieke Jans. An improved way for measuring simplicity during process discovery. In *Workshop on Enterprise and Organizational Modeling and Simulation*, pages 49–62. Springer, 2018.

[64] Ahmet Dikici, Oktay Turetken, and Onur Demirors. Factors influencing the understandability of process models: A systematic literature review. *Information and Software Technology*, 93:112–129, 2018.

[65] Anne Rozinat, AK Alves De Medeiros, Christian W Günther, AJMM Weijters, and Wil MP Van der Aalst. Towards an evaluation framework for process mining algorithms. *BPM Center Report BPM-07-06, BPMcenter. org*, 123:142, 2007.

[66] Joos C. A. M. Buijs, Boudewijn F. van Dongen, and Wil M. P. van der Aalst. On the role of fitness, precision, generalization and simplicity in process discovery. In Robert Meersman, Hervé Panetto, Tharam S. Dillon, Stefanie Rinderle-Ma, Peter Dadam, Xiaofang Zhou, Siani Pearson, Alois Ferscha, Sonia Bergamaschi, and Isabel F. Cruz, editors, *On the Move to Meaningful Internet Systems: OTM 2012, Confederated International Conferences: CoopIS, DOA-SVI, and ODBASE 2012, Rome, Italy, September 10-14, 2012. Proceedings, Part I*, volume 7565 of *Lecture Notes in Computer Science*, pages 305–322. Springer, 2012. doi: 10.1007/978-3-642-33606-5_19. URL `https://doi.org/10.1007/978-3-642-33606-5_19`.

[67] Daniel Reißner, Abel Armas-Cervantes, and Marcello La Rosa. Generalization in automated process discovery: A framework based on event log patterns. *Computing Research Repository*, abs/2203.14079, 2022. doi: 10.48550/arXiv.2203.14079. URL `https://doi.org/10.48550/arXiv.2203.14079`.

[68] Fabrizio M Maggi, RP Bose, and Wil MP van der Aalst. Efficient discovery of understandable declarative process models from event logs. In *International Conference on Advanced Information Systems Engineering*, pages 270–285. Springer, 2012.

[69] Gert Janssenswillen, Toon Jouck, Mathijs Creemers, and Benoît Depaire. Measuring the quality of models with respect to the underlying system: An empirical study. In Marcello La Rosa, Peter Loos, and Oscar Pastor, editors, *Business Process Management - 14th International Conference, BPM 2016, Rio de Janeiro, Brazil, September 18-22, 2016. Proceedings*, volume 9850 of *Lecture Notes in Computer Science*, pages 73–89. Springer, 2016. doi: 10.1007/978-3-319-45348-4_5. URL `https://doi.org/10.1007/978-3-319-45348-4_5`.

[70] Andrew Gemino and Yair Wand. A framework for empirical evaluation of conceptual modeling techniques. *Requirements Engineering*, 9(4):248–260, 2004.

[71] Andrew Burton-Jones, Yair Wand, and Ron Weber. Guidelines for empirical evaluations of conceptual modeling grammars. *Journal of the Association for Information Systems*, 10 (6):1, 2009.

[72] Andrew Gemino and Yair Wand. A framework for empirical evaluation of conceptual modeling techniques. *Requirements Engineering*, 9(4):248–260, 2004. doi: 10.1007/s00766-004-0204-6. URL https://doi.org/10.1007/s00766-004-0204-6.

[73] Elias Mturi and Paul Johannesson. A context-based process semantic annotation model for a process model repository. *Business Process Management Journal*, 2013.

[74] Paul A Kirschner. Cognitive load theory: Implications of cognitive load theory on the design of learning, 2002.

[75] Stefan Zugal, Jakob Pinggera, Hajo Reijers, Manfred Reichert, and Barbara Weber. Making the case for measuring mental effort. In *Proceedings of the Second Edition of the International Workshop on Experiences and Empirical Studies in Software Modelling*, pages 1–6, 2012.

[76] Fred D Davis. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, pages 319–340, 1989.

[77] Daniel L Moody. The method evaluation model: a theoretical model for validating information systems design methods. 2003.

[78] Marcello La Rosa, Petia Wohed, Jan Mendling, Arthur H. M. ter Hofstede, Hajo A. Reijers, and Wil M. P. van der Aalst. Managing process model complexity via abstract syntax modifications. *IEEE Transactions on Industrial Informatics*, 7(4):614–629, 2011. doi: 10.1109/TII.2011.2166795. URL https://doi.org/10.1109/TII.2011.2166795.

[79] Philipe De Meyer and Jan Claes. An overview of process model quality literature - the comprehensive process model quality framework. *Computing Research Repository*, abs/1808.07930, 2018. URL http://arxiv.org/abs/1808.07930.

[80] Hajo A Reijers and Jan Mendling. A study into the factors that influence the understandability of business process models. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 41(3):449–462, 2010.

[81] Vered Bernstein and Pnina Soffer. Identifying and quantifying visual layout features of business process models. In Khaled Gaaloul, Rainer Schmidt, Selmin Nurcan, Sérgio Guerreiro, and Qin Ma, editors, *Enterprise, Business-Process and Information Systems Modeling - 16th International Conference, BPMDS 2015, 20th International Conference, EMMSAD 2015, Held at CAiSE 2015, Stockholm, Sweden, June 8-9, 2015, Proceedings*, volume 214 of *Lecture Notes in Business Information Processing*, pages 200–213. Springer, 2015. doi: 10.1007/978-3-319-19237-6_13. URL https://doi.org/10.1007/978-3-319-19237-6_13.

[82] Daniel L. Moody. The x201cphysicsx201d of notations: Toward a scientific basis for constructing visual notations in software engineering. *IEEE Transactions Software Engineering*, 35(6):756–779, 2009. doi: 10.1109/TSE.2009.67. URL https://doi.org/10.1109/TSE.2009.67.

[83] Wasana Bandara, Guy G Gable, and Michael Rosemann. Factors and measures of business process modelling: model building through a multiple case study. *European Journal of Information Systems*, 14(4):347–360, 2005.

[84] Marian Petre. Why looking isn't always seeing: readership skills and graphical programming. *Communications of the ACM*, 38(6):33–44, 1995.

[85] Jan Recker and Alexander Dreiling. The effects of content presentation format and user characteristics on novice developers' understanding of process models. *Communications of the Association for Information Systems*, 28(1):6, 2011.

[86] Jan Mendling, Mark Strembeck, and Jan Recker. Factors of process model comprehension—findings from a series of experiments. *Decision support systems*, 53(1):195–206, 2012.

[87] Kathrin Figl. Comprehension of procedural visual business process models. *Business & Information Systems Engineering*, 59(1):41–67, 2017.

[88] August-Wilhelm Scheer. *ARIS—business process modeling*. Springer Science & Business Media, 2000.

[89] Charles Ashbacher. "the unified modeling language reference manual, second edition", by james rumbaugh. *Journal of Object Technology*, 3(10):193–195, 2004. doi: 10.5381/jot.2004.3.10.r1. URL `https://doi.org/10.5381/jot.2004.3.10.r1`.

[90] Kamyar Sarshar and Peter Loos. Comparing the control-flow of epc and petri net from the end-user perspective. In *International Conference on Business Process Management*, pages 434–439. Springer, 2005.

[91] Jan Recker and Alexander Dreiling. Does it matter which process modelling language we teach or use? an experimental study on understanding process modelling languages without formal education. *ACIS 2007 Proceedings*, page 45, 2007.

[92] Helen C Purchase, Linda Colpoys, Matthew McGill, and David Carrington. Uml collaboration diagram syntax: an empirical study of comprehension. In *Proceedings First International Workshop on Visualizing Software for Understanding and Analysis*, pages 13–22. IEEE, 2002.

[93] Wil Van der Aalst, Ton Weijters, and Laura Maruster. Workflow mining: Discovering process models from event logs. *IEEE Transactions on Knowledge and Data Engineering*, 16(9):1128–1142, 2004.

[94] Wil M. P. van der Aalst. *Process Mining - Discovery, Conformance and Enhancement of Business Processes*. Springer, 2011. ISBN 978-3-642-19344-6. doi: 10.1007/978-3-642-19345-3. URL `https://doi.org/10.1007/978-3-642-19345-3`.

[95] Boudewijn Frans van Dongen. Process mining and verification. 2007.

[96] Lijie Wen, Wil M. P. van der Aalst, Jianmin Wang, and Jiaguang Sun. Mining process models with non-free-choice constructs. *Data Mining and Knowledge Discovery*, 15(2): 145–180, 2007. doi: 10.1007/s10618-007-0065-y. URL `https://doi.org/10.1007/s10618-007-0065-y`.

[97] AJMM Weijters, Wil MP van Der Aalst, and AK Alves De Medeiros. Process mining with the heuristics miner-algorithm. *Technische Universiteit Eindhoven, Tech. Rep. WP*, 166 (July 2017):1–34, 2006.

[98] Anne Rozinat. How to understand the variants in your process. `https://fluxicon.com/blog/2012/11/how-to-understand-the-variants-in-your-process/`, n.d.

[99] Felix Mannhardt and Daan Blinde. Analyzing the trajectories of patients with sepsis using process mining. In Jens Gulden, Selmin Nurcan, Iris Reinhartz-Berger, Wided Guédria, Palash Bera, Sérgio Guerreiro, Michael Fellmann, and Matthias Weidlich, editors, *Joint Proceedings of the Radar tracks at the 18th International Working Conference on Business Process Modeling, Development and Support (BPMDS), and the 22nd International Working Conference on Evaluation and Modeling Methods for Systems Analysis and Development (EMMSAD), and the 8th International Workshop on Enterprise Modeling and Information Systems Architectures (EMISA) co-located with the 29th International Conference on Advanced Information Systems Engineering 2017 (CAiSE 2017), Essen, Germany, June 12-13, 2017*, volume 1859 of *CEUR Workshop Proceedings*, pages 72–80. CEUR-WS.org, 2017. URL `http://ceur-ws.org/Vol-1859/bpmds-08-paper.pdf`.

[100] Renee M Hendricks. Process mining of incoming patients with sepsis. *Online Journal of Public Health Informatics*, 11(2), 2019.

[101] Gert Janssenswillen, Benoit Depaire, Marijke Swennen, Mieke Jans, and Koen Vanhoof. bupar: Enabling reproducible business process analysis. *Knowledge-Based Systems*, 163: 927–930, 2019.

[102] Alessandro Berti, Sebastiaan J Van Zelst, and Wil van der Aalst. Process mining for python (pm4py): bridging the gap between process-and data science. *arXiv preprint arXiv:1905.06169*, 2019.

[103] R Ihaka and R Gentleman. R: A language for data analysis and graphics. journal of computational and graphical statistics 5: 299. *doi: 10.2307/1390807*, 1996.

[104] Alessandro Berti, Sebastiaan J van Zelst, and Wil MP van der Aalst. Pm4py web services: Easy development, integration and deployment of process mining features in any application stack. In *BPM (PhD/Demos)*, pages 174–178, 2019.

[105] Gert Janssenswillen and Benoît Depaire. bupar: Business process analysis in r. In *BPM (Demos)*, 2017.

[106] Daniel L Moody. Comparative evaluation of large data model representation methods: The analyst's perspective. In *International Conference on Conceptual Modeling*, pages 214–231. Springer, 2002.

[107] Martin Fishbein and Icek Ajzen. Belief, attitude, intention, and behavior: An introduction to theory and research. *Philosophy and Rhetoric*, 10(2), 1977.

[108] N Jovancic. Likert scale: How to create your own survey. *Lead Quizzes. Retrieved March*, 25:2020, 2020.

[109] Junyong In. Introduction of a pilot study. *Korean Journal of Anesthesiology*, 70(6):601, 2017.

[110] Ashley Crossman. Convenience sampling. Technical report, Retrieved 1/20/2018, from https://www. thoughtco. com/convenience-sampling . . . , 2017.

[111] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30, 2006.

[112] Milton Friedman. A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics*, 11(1):86–92, 1940.

[113] Chunpeng Fan and Donghui Zhang. On power and sample size of the anova-type rank test. *Communications in Statistics-Simulation and Computation*, 46(4):3224–3241, 2017. doi: 10.1080/03610918.2015.1082585. URL `https://doi.org/10.1080/03610918.2015.1082585`.

[114] Gregory W Corder and Dale I Foreman. *Nonparametric statistics: A step-by-step approach*. John Wiley & Sons, 2014.

[115] Peter Bjorn Nemenyi. *Distribution-free multiple comparisons.* Princeton University, 1963.

[116] Ronald E Walpole, Raymond H Myers, Sharon L Myers, and Keying Ye. *Probability and statistics for engineers and scientists*, volume 5. Macmillan New York, 1993.

[117] Sebastiaan J. van Zelst and Sander J. J. Leemans. Translating workflow nets to process trees: An algorithmic approach. *Algorithms*, 13(11):279, 2020. doi: 10.3390/a13110279. URL `https://doi.org/10.3390/a13110279`.

Appendices

A bupar and PM4PY codes to Mine Process Models

A.1 bupaR-Code for causal net

```
#####Importing the log###########################
Sepsis <- read_xes("D://Thesis//Sepsis Cases - Event Log.xes")
summary(Sepsis)
################Heuristic Miner##################
m <- precedence_matrix_absolute(Sepsis)
as.matrix(m)
dependency_matrix(Sepsis, threshold = .7) %>% render_dependency_matrix()
causal_net(Sepsis, threshold = .7) %>% render_causal_net()
#####Process Variants#######
Sepsis %>%
  trace_explorer()
```

A.2 PM4PY Codes for Petri net and Process Tree

```
############Importing the log######################

import pm4py

Sepsis = pm4py.read_xes('D:/Thesis/Sepsis Cases - Event Log.xes')

print(Sepsis)


###################Alpha Miner###################

from pm4py.algo.discovery.alpha import algorithm as alpha_miner

net, initial_marking, final_marking = alpha_miner.apply(Sepsis)

from pm4py.visualization.petri_net import visualizer as pn_visualizer

gviz = pn_visualizer.apply(net, initial_marking, final_marking)

pn_visualizer.view(gviz)


#################Inductive Miner################

from pm4py.algo.discovery.inductive import algorithm as inductive_miner

from pm4py.visualization.process_tree import visualizer as pt_visualizer

tree = inductive_miner.apply_tree(Sepsis)

gviz = pt_visualizer.apply(tree)

pt_visualizer.view(gviz)
```

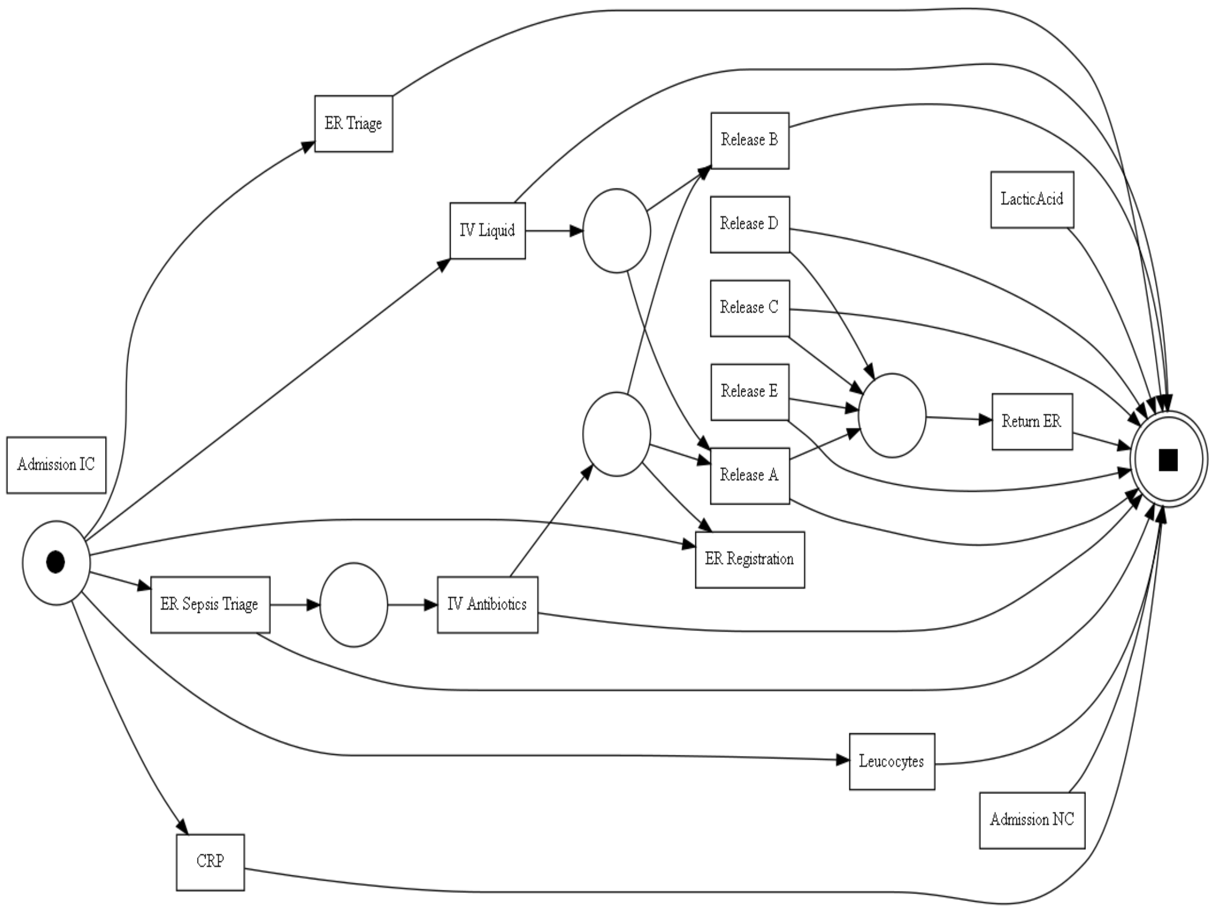
A.3 R codes for Friedman's test and Box Plot

```
#####Importing Ratings Data ###########################
Ratings <- read.csv("D://Thesis//Master's Thesis-Ratings of Models.csv", header=T, sep=',')
attach(Ratings)
head(Ratings)
median(Model.A.Ratings)
median(Model.B.Ratings)
median(Model.C.Ratings)
#####BOXPLOT####
boxplot(Model.A.Ratings, Model.B.Ratings, Model.C.Ratings,
```

```
        main= "Students’ Perceived Comprehensibility Ratings Over The Three Models",
        xlab = "Process Models", ylab= " Comprehensibility Ratings",
        names=c("Petri net (Model A)","Causal net (Model B)","Process tree (Model C)"))
###Carrying out friedman’s test#################
Sample<-matrix(c(Model.A.Ratings,Model.B.Ratings,Model.C.Ratings),ncol=3)
friedman.test(Sample)
```

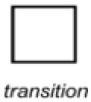
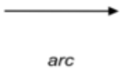
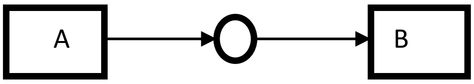
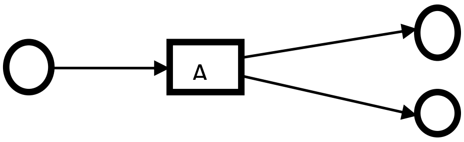
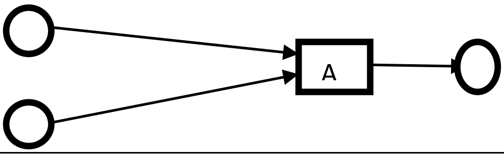
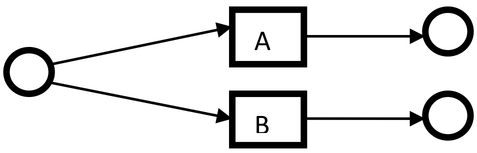
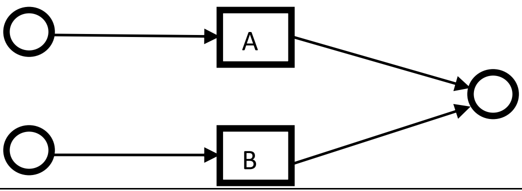
B Mined Control-Flow Models

B.1 Model A (Petri net)

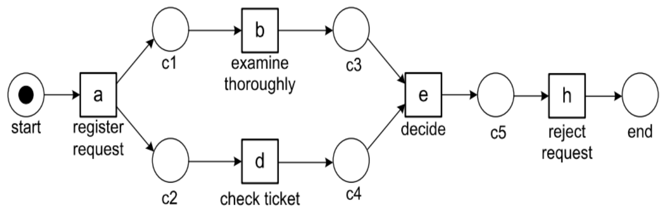


B.1.1 Cheatsheet Model A

Cheat sheet (Model A)

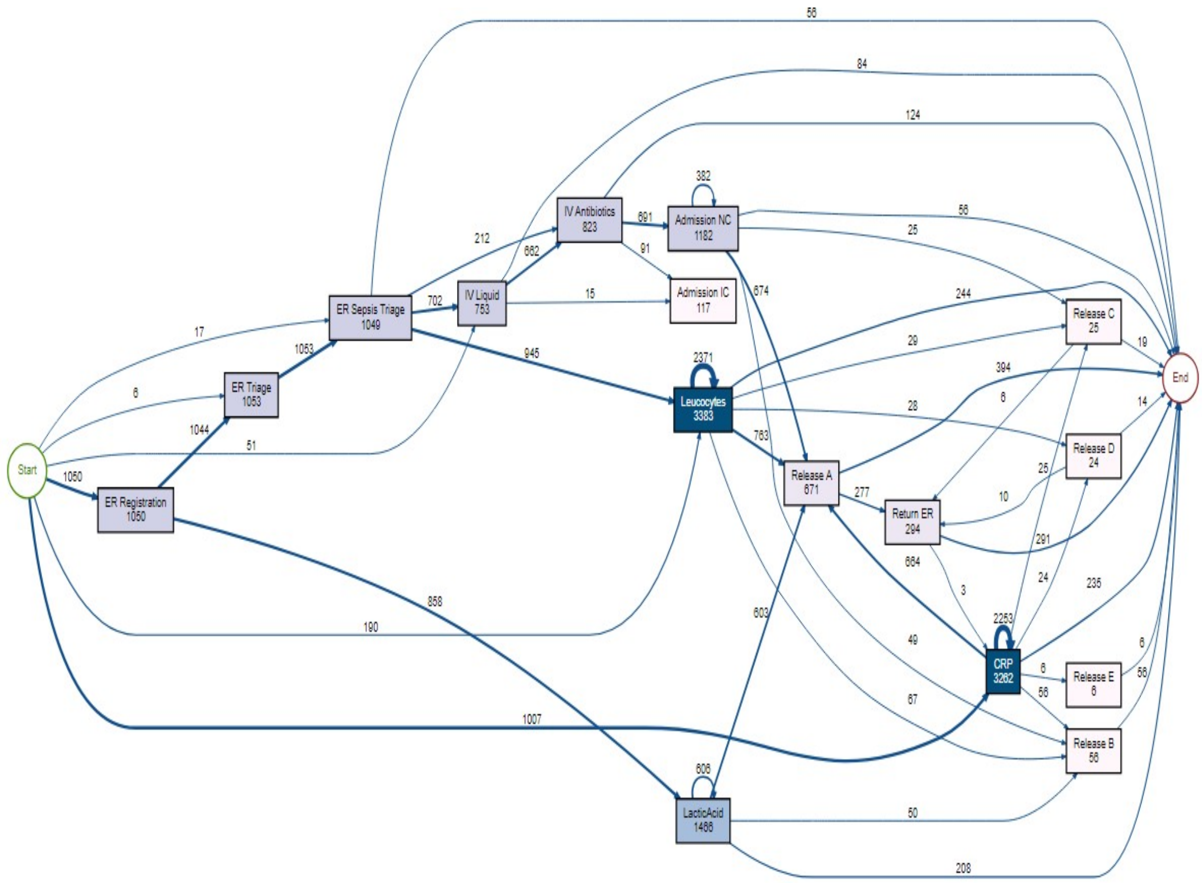
Elements	**Description**
	Places symbolise states, conditions, or resources that need to be met/be available before an action can be carried.
	A Transitions represents an activity in a process model.
	An Arc is a directed connection between a place and a transition. It shows the interaction between places and transitions, including the workflow of the process.
	- A token is a black dot. It often represents objects throughout the process and can only be found in places. Once a token is put in the end place, the process ends. - A token on a place means that the corresponding condition is fulfilled or that a resource is available.
	Activity is executed in sequence.
	"AND" Split (Parallel split) A parallel split occurs when one transition is connected to multiple output places
	"AND" join (Parallel join) A parallel join occurs when multiple input places are connected to a single transition.
	*"XOR"* Split An exclusive split occurs when one input place is connected to two transitions
	*"XOR" Join* *An exclusive join occurs when two transitions are connected to one output place*

Example Model and Interpretation



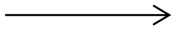
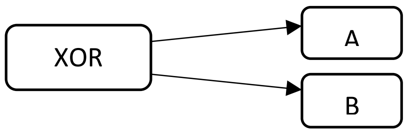
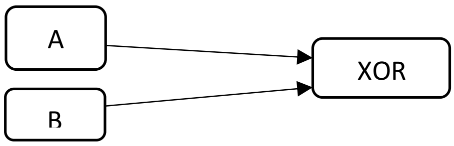
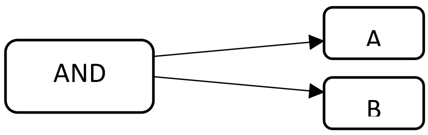
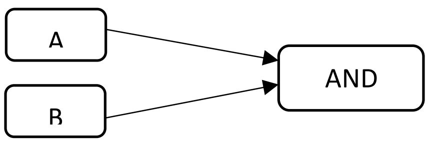
The process starts with register request, the activities examine thoroughly and check ticket are executed in parallel. Finally, the process ends when request is rejected.

B.2 Model B (Causal net)



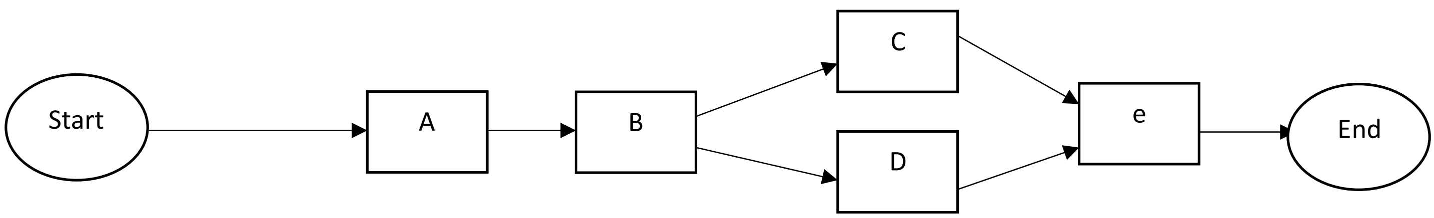
B.2.1 Cheatsheet Model B

Cheat Sheet-Model B

Elements	Description
 node	Nodes Represent activities. Each activity has a set of possible input bindings and a set of possible output bindings.
 arc	An arc represents causal dependencies between nodes (activities). Arcs may be annotated with frequency (how often a certain relation between activities is observed) and/or confidence/certainty.
 Loop operator	An activity can be repeated
	XOR-Split. Separates two or more alternative activities
	XOR-join to merge two or more alternative activities that may have previously been separated with an XOR-split.
	AND-Spit. Crete parallel flows with activities that can be executed concurrently.
	AND-join. Merges parallel activities

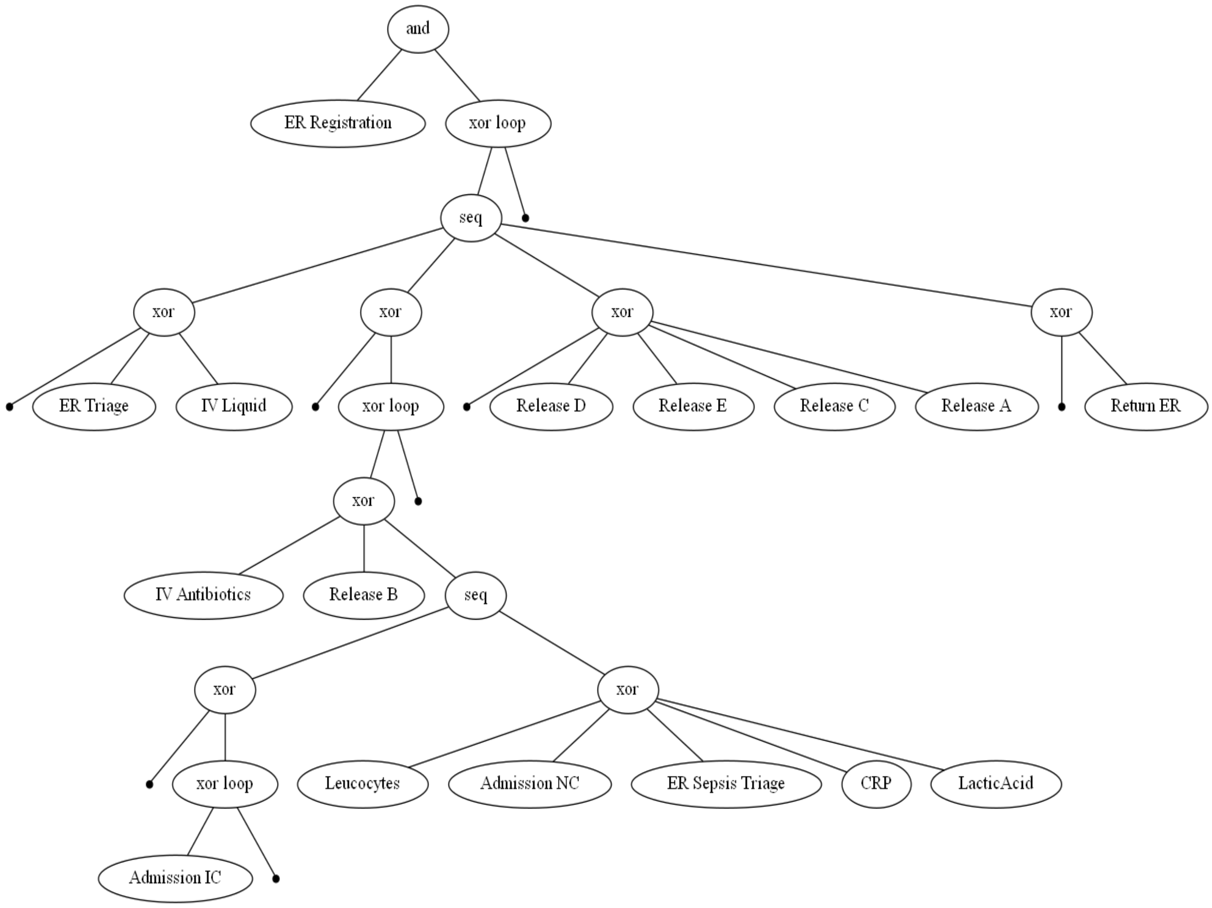
NB: there is no specific detection mechanism for the XOR (exclusive split and join) and the AND (parallel split and join) constructs. However, they can be identified by looking at the frequency of activities.

Example Model



Activity A is executed, followed by B. When B is executed, both C and D are executed parallelly. E is activated once C and D are completed. The process ends when E is completed.

B.3 Model C (Process Tree)

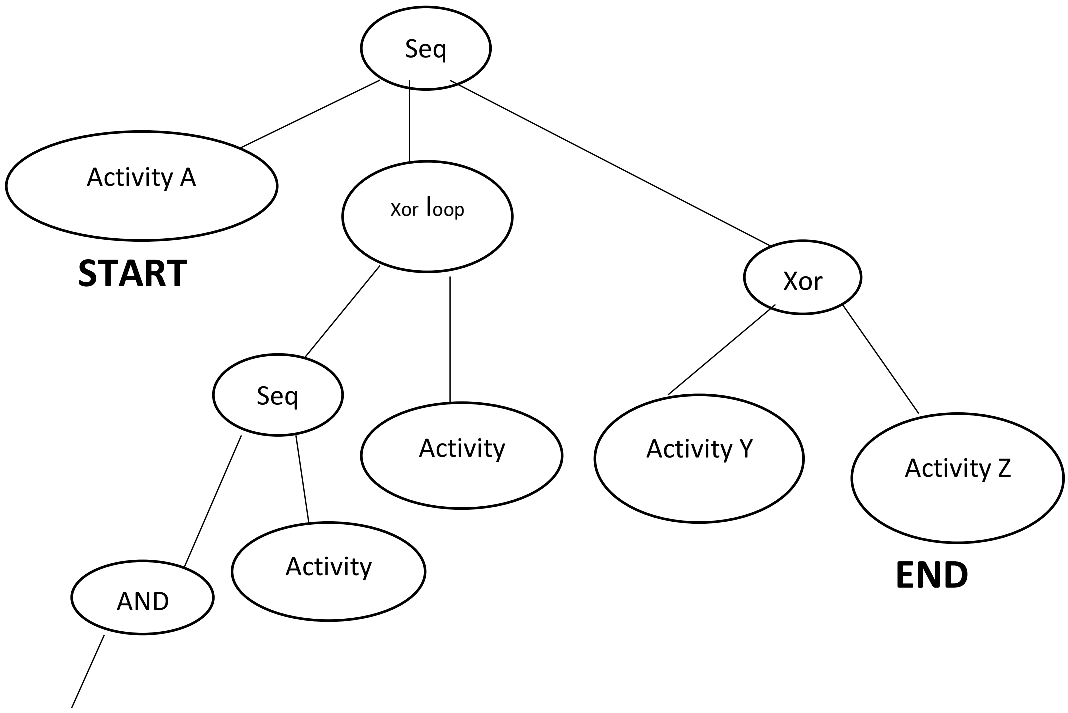


B.3.1 Cheatsheet Model C

Cheat sheet-Model C

Elements	Description
and — Parallel split	Activities are executed concurrently
Xor loop — Loop	Activity can be repeated
Xor — Exclusive choice	Choice between activities
Seq — Sequence	Activities are executed in sequence
Leaf node	Each leaf node represents activity from the collection of activities

Example model



This model is read from top left coming downward, then top right coming downward. Hence, the process starts with activity A and ends with activity Z.

C Task Statements/Questions

C.1 Task A

1. What is your age?
2. What is your gender?
3. What is your educational Background (Previous degree and specialization)?
4. How many semester(s) have you completed so far in the Master's of Business Process Management?
5. How do you assess your theoretical knowledge of process modeling?

- I have very weak theoretical knowledge
- I have weak theoretical knowledge
- I have average theoretical knowledge
- I have strong theoretical knowledge
- I have very strong theoretical knowledge

6.a. Do you have any practical experience with business process modeling?

- Yes
- No

6.b. If Yes, can you elaborate a bit on your experience?
7.a. Do you know any other process modeling language(s) other than BPMN?

- Yes
- No

7.b. If Yes, please list them and provide your experiences with them.

C.2 Task B

1.1. I have trouble understanding the flow of activities in the process.
1.2. I would understand the process better if there were some "display mechanism" that would show me step by step where the process is at a point in time.
1.3. I had to think about a decision situation for a long time before I could understand which path (or paths in the process is (or are to be) pursued further.
2. In principle, I consider this method unsuitable for presenting processes in an understandable manner for inexperienced users.
3.1. I would use this method in a company to present business processes and workflows in a way that employees can understand.
3.2. I find this method suitable for experts to plan processes.
Task 1 to 3 were assessed using the following 4 point Likert scale

- I strongly agree.
- I agree.
- I disagree.
- I strongly disagree.

4. How do you evaluate the following process model?
Rate the above question using a number from 1 to 7 according to the labels shown below:
1= "Extremely difficult to understand the model" 2= "Very difficult to understand the model" 3= "A bit difficult to understand the model" 4= "Neither difficult nor easy to understand the model" 5"= "Quite easy to understand the model" 6= "Very easy to understand the model" 7= "Extremely easy to understand the model"

Closing Question
Any recommendations to improve this model's clarity?