Master Thesis

Combining Process Mining and Queueing Theory for ICT Ticket Resolution Process at LUMC

Ayşe Aslan

UNIVERSITY OF TWENTE.

LUMC

Supervised by
Prof. Dr. Richard Boucherie
Dr. Maartje van de Vrugt

August 2017
This Thesis Is Submitted to University of Twente
For the Degree of Master of Science in Applied Mathematics

Preface

This thesis is the product of the final project which is a substantial part of the master program in applied mathematics in University of Twente, in Netherlands. I conducted this project at the ICT department in Leiden University Medical Center (LUMC), in Netherlands. In this project, I investigated the ticket resolution process of the ICT department. Academically, this project is supervised by Dr. Maartje van de Vrugt and Prof. Dr. Richard Boucherie.

My journey of research is started when we formulated the research questions together with my supervisor Willem van Duyvenvoorde at the ICT department in LUMC and academic supervisors at University of Twente to conduct a project which both answers the needs of the ICT department and constitutes a firm and sound work of applied mathematics.

I started the project in February 2017 and ended in August 2017. During that time, my supervisor Willem van Duyvenvoorde was always pleased to answer my queries to understand the ticket resolution process of the ICT department. Firstly, I would like to thank Willem van Duyvenvoorde for his time and support.

I would like to thank my supervisors Dr. Maartje van de Vrugt and Prof. Dr. Richard Boucherie for their valuable comments, guidance and support throughout the project.

Finally, I would like to thank my family and friends for being helpful and supportive.

Ayşe Aslan

25 August 2017, Enschede

Summary

The research in this thesis analyzes and supports the ticket resolution process of ICT department in LUMC. The research aims to investigate the performance of the ticket resolution process and provide predictions for the resolution times of the tickets to support the ticket resolution process. The main purpose of the research is combining process mining and queueing theory techniques for the ticket resolution process. The purpose of utilizing process mining is to gain insights of the ticket resolution process from the historical data. This research aims to accomplish the mission of combining process mining and queueing theory by making use of the process insights that process mining provides to build a queueing network model of the ticket resolution process. This research aims to take a queueing theory perspective in providing resolution time predictions for tickets. This research proposes a stochastic Petri net approach which incorporates mean queueing performances of the queueing network model of the ticket resolution process to provide queueing theory perspective predictions.

Thesis is composed of 9 chapters each of them dealing with different aspects of the research. Chapter 1 is introductory. Chapter 1 is subdivided into 5 sections. Section 1.1 describes the management process of incidents that are reported as tickets at the ICT department. Section 1.2 describes the research subject of ticket resolution process at the ICT department. Section 1.3 explains the research motivations. Section 1.4 describes the research objectives and questions. Section 1.5 describes the outline of the thesis.

Chapter 2 gives a brief literature review on the topics that relate to the research questions of the thesis. Chapter 2 is subdivided into 4 sections. Section 2.1 gives a literature review on process mining which explores the extent of the insights about processes that can be discovered with process mining. Section 2.2 gives a literature review on ticket resolution. Section 2.3 gives a review of literature which utilizes both process mining and queueing theory techniques. Section 2.4 presents a review of literature on the techniques of predicting time in processes.

Chapter 3 introduces process mining to the readers. Chapter 3 is subdivided into 5 sections. Section 3.1 briefly describes process mining. Section 3.2 states the questions that process mining can answer about processes. Section 3.3 describes the tools to perform process mining. Section 3.4 gives an illustration of performing process mining to a sample event log and discovering information. Section 3.5 describes quality criteria to assess the quality of discovered process models.

Chapter 4 outlines the research approach. Chapter 4 is subdivided into 3 sections. Section 4.1 states the goals of the research approach. Section 4.2 illustrates the overview of the research methodology. Section 4.3 explains how process mining and queueing theory are combined in the research.

Chapter 5 describes how process mining is applied to the ticket resolution process of

the ICT department in LUMC and reports the analyses conducted with several process mining tools on the process. Chapter 5 is subdivided into 4 sections. Section 5.1 explicates the event log preparation for the ticket resolution process. Section 5.2 concentrates on the efficiency analysis of the ticket resolution process with process mining tools. Section 5.3 describes the process discovery with Inductive Visual Miner process mining tool for the ticket resolution process. Section 5.4 presents a process mining tool to discover social relations among the operators of the ticket resolution process.

Chapter 6 contains the queueing network model of the ticket resolution process. Chapter 6 is subdivided into 5 sections. Section 6.1 explains the gained insights by process mining on the ticket resolution process that shape the queueing network model. Section 6.2 presents the queueing network model of the ticket resolution process. Section 6.3 explains the server structure of delay. Section 6.4 explains the server structure of operators. Section 6.5 describes the derivation of the model input parameters.

Chapter 7 provides the analysis of the model. Chapter 7 is subdivided into 4 sections. Section 7.1 provides the analysis of operators. Section 7.2 provides the analysis of delay. Section 7.3 provides the analysis of the network. Section 7.4 provides the obtained results from the implementation of the analyses.

Chapter 8 proposes the stochastic Petri net methodology to produce resolution time predictions and provides the results of implementing resolution time predictions. Chapter 8 is subdivided into 5 sections. Section 8.1 explains the resolution time prediction approach of the research. Section 8.2 describes how to represent the ticket resolution process with stochastic Petri nets. Section 8.3 presents the resolution time prediction method. Section 8.4 provides the results of implementing the resolution time prediction method. Section 8.5 gives the results of numerical experiments of investigating the performance improvements of the ticket resolution process.

Chapter 9 contains the drawn conclusions and recommendations. Chapter 9 is subdivided into 3 sections. Section 9.1 presents the conclusions drawn. Section 9.2 provides recommendations for the ICT department. Section 9.3 provides ideas on further research into the ticket resolution process.

Contents

List of Figures

List of Tables

1 Introduction

In this chapter, we introduce the subject of our research and state our motivations, objectives and research questions.

1.1 Incident Management Process

We describe the process of incident management at ICT department in LUMC. Incidents are the unplanned interruptions or reductions in the service quality of the ICT services. The process of incident management manages the life-cycle of all incidents.

With the ultimate aim of minimizing the negative effects to LUMC, fixing the disruptions as soon as possible constitutes the main goal of the incident management. There are several underlying goals of the incident management. In order to shorten the disruptions solving the incidents during the first contact with the service desk constitutes the first underlying goal. The second underlying goal is making use of the knowledge items in the knowledge system in the solution of common questions and service disruptions. The last underlying goal is making use of the standard solutions in the solution of common questions and service disruptions.

There are several variants of incident management process. The incidents that have great impact and/or urgency are major incidents and for them the variant of major incident process applies. For the incidents that involve security issues, the variant of security incident process applies.

1.1.1 Key Performance Indicators

Several key performance indicators (KPIs) are set by ICT department to determine if the goals of the incident management are met. In defining the KPIs, the following values are utilized.

- Number of incidents that are resolved in the first line
- Number of incidents that are resolved within the target resolution times
- Number of incidents that stay open longer than one month
- Number of major incidents vs. number of incidents
- Number of incidents which the knowledge system offers a solution

The KPIs of the incident management at the ICT department are listed below.

- KPI 1: Percentages of the incidents that are resolved in the first line
  Gives insight into what extent the hospital employees are helped directly and gives insights into what extent service desk employees are skilled.

- KPI 2: Percentages of the incidents that are resolved within the target resolution times
  Gives insight into the efficiency of the incident management process.

- KPI 3: Percentages of the incidents that have stayed open for longer than one month
  Identifies the possible bottlenecks in the resolution of the incidents.

- KPI 4: Percentages of the major incidents
  Gives insight into the extent to which the management is done in a good way.

- KPI 5: Percentages of the incidents that are resolved using the knowledge system
  Gives insights into the quality of the knowledge system.

1.1.2 Overall Process Flow

The overall process map of the incident management is given in figure 1.1. This diagram presents the handling of the incidents through the service desk employees at line 1 and the operators at line 2. Other than the service desk employees and the operators at line 2, the security officer is involved in the handling of the incidents that involve security issues.

The service desk employees try to relate incidents to the known incidents, make use of the knowledge system and the standard solutions in resolving the incidents. If a resolution is found at first line for an incident, the incident is not forwarded to the second line operators. The operators at second line can forward an incident to the other operators if necessary.

There are several sub-processes in the incident management process. An incident may relate to a major incident process, to a change management process, to a configuration management process and to a knowledge management process. In our research, we only focus on finding resolutions to the incidents. We do not include the sub-processes of the incident management. We do not extend our research to what happens to the incidents when they are recognized as major incidents and forwarded to the major incident process, or recognized as change requests and forwarded to the change management process, etc.

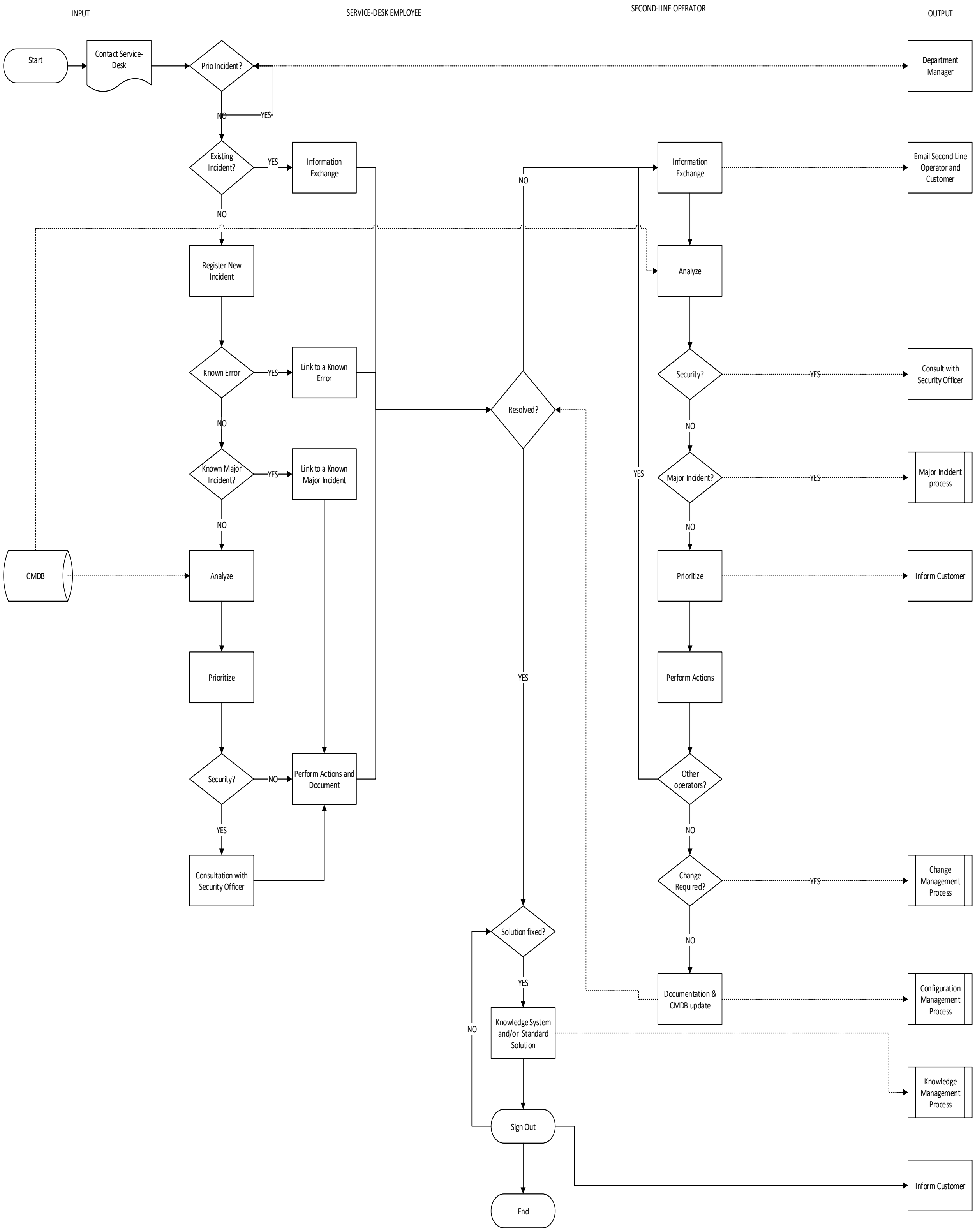


Figure 1.1: Process Map of Incident Management

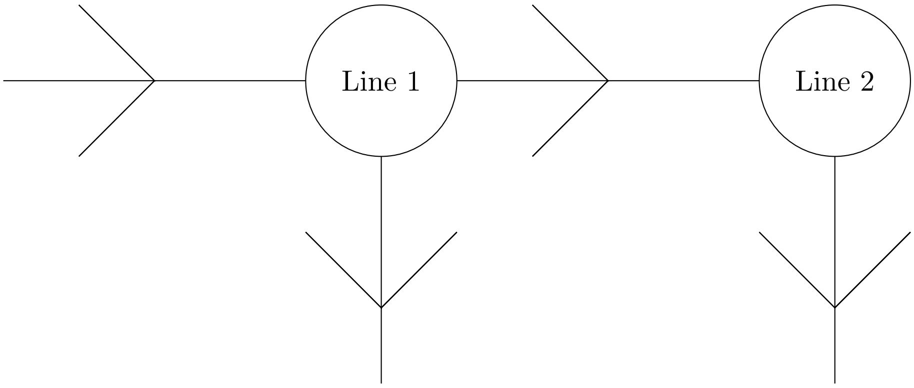


Figure 1.2: Line 1 and Line 2

1.2 ICT Ticket Resolution Process

In this section, we describe the process that we model and analyze in our research. This process is the ICT ticket resolution process which serves the ICT requests of the hospital employees at LUMC. The tickets report incidents and tickets are resolved by managing the incidents they relate by the ICT department. In our research, we refer incidents as tickets and refer incident handling/resolution as ticket handling/resolution.

Hospital employees at LUMC create tickets via an email, a call or a visit to the service desk. ICT serves the tickets of the hospital employees at two lines: first line and second line. First line is the first place that all tickets are registered and handled by the service desk employees, while the second line is the place where specialist operators and operator groups work. The tickets are escalated from first line to operator groups/operators at second line if the tickets could not be resolved in first line.

ICT operators can have roles as a first line operator or/and as a second line operator depending on their specializations. For instance, a service desk employee who has both line 1 and line 2 operator role, serves the tickets at first line as a first line operator, and at the same time serves the tickets at second line which belongs to a category that he/she is specialized in as a second line operator. ICT operators may not belong to any operator groups and may belong to more than one operator groups.

The escalation of a ticket from first line to second line can be directly to an operator or can be to an operator group at second line. If a ticket is directly escalated to an operator at second line, the resolution service of the ticket begins immediately, where in the case that the ticket is escalated to an operator group at second line, the resolution service of the ticket does not start immediately and the ticket waits to be assigned to an operator in the operator group. Each operator group at second line exercises different policies to assign the tickets to the operators within the group. Some of the operator groups let their operators choose the tickets that they would like to handle, while some of the operator groups directly assigns the tickets to the operators inside the group.

There are 3 main types of tickets: questions, service requests and malfunctions. When tickets are created, context information relating to tickets are recorded. Context infor-

Urgency	Description
High	One can not continue working
Middle	One can continue to work but there is a serious disruption of the normal process
Normal	One can continue to work with some disruption of the normal process

Table 1.1: Urgency Level Measurement

Impact	Description
High	The incident affects all LUMC
Middle	The incident affects one or more departments
Normal	The incident affects one or more persons

Table 1.2: Impact Level Measurement

mation is taken for the purposes of classifying and determining the priority levels of the tickets. The information about the object (software, hardware, machine, account, etc.) relating to the tickets is used to classify the category and subcategory of the tickets, whereas the information of urgency and impact level of the tickets are used to determine the priority levels of the tickets. Urgency is measured by the level of disruption in the work process, whereas impact is measured by the extent of the disruption. For instance, a ticket relating to security issues carries high urgency as it is vital to resolve the ticket immediately and a ticket which relates to a malfunction of a MR machine has a high impact level as this type of malfunction affects patients, doctors, nurses, etc. at the same time. Both the category information and the priority information plays an important role in the flow of the tickets. The category information plays an important role as if a ticket is escalated to second line, the ticket is transferred to an operator or operator group who is specialized in the category of the ticket. The priority information is important as the tickets carrying high priorities are prioritized by the operators. The priority levels of the tickets affect the resolution times of the tickets as ICT has target resolution times for the tickets by priority levels.

The tickets that are escalated to second line are routed between operators/operator groups until they are resolved. If an operator who deals with a ticket decides that he/she can not resolve the ticket or there is a permission, authority or knowledge required from other operators/ operator groups for the resolution of the ticket, the operator can transfer the ticket to other operators/operator groups. When a ticket is transferred from

Priority	Impact		
Urgency	High	Middle	Normal
High	1 Critic	2 High	3 Middle
Middle	2 High	3 Middle	4 Important
Normal	3 Middle	4 Important	5 Normal

Table 1.3: The Priority Table

Priority	First Line Resolution Time	Second Line Resolution Time
Critic	15 mins	2 hours
High	30 mins	1 day
Middle	1 hour	3 days
Important	1 day	1 week
Normal	1 week	1 week

Table 1.4: Target Resolution Times at Line 1 and Line 2

an operator/operator group to another operator/operator group, the ticket is not served until the operator/operator group who the ticket is transferred to responses, thus the ticket's resolution is delayed.

In our research, we consider KPI 2 as a measurement to assess the performance of the ticket resolution process and therefore we focus on the resolution times of the tickets. The resolution time of a ticket consists of the resolution service times at the operators who the ticket visits along its path to resolution and the delay times when the ticket is transferred from an operator group/operator to another operator group/operator.

1.3 Motivation

- **Motivation of the ICT Department in LUMC**

  The topdesk system, the platform where the tickets are registered and resolved by the operators, that serves the employees of LUMC has been keeping the recordings of the activities on the tickets since 2005. ICT department has the motivation to mine the data of historical activities on the tickets to investigate the ticket handling process and reach results that will improve their process so that they better serve the tickets of hospital employees.

- **Applying Process Mining to a Different Process Inside a Hospital**

  Process mining as a data science tool which bridges data mining and business process intelligence makes it possible to discover processes and extract knowledge from event logs. As it is already successfully in action in gaining insights about healthcare processes from relevant patient data and hospital records, carrying process mining to an other process in a hospital, to a service process in an ICT department inside a Hospital, constituted a motivation for us. Our motivation is to make use of process mining to understand what is actually happening in the ICT ticket resolution process in LUMC. Process mining outputs a process model that is extracted from the data, organizational relations between resources and mean performance measures by taking an event log as input. We aim to benefit from every aspect that process mining provides to better understand a process.

- **Supporting the Ticket Handling Process**

The tickets are prioritized before handling, and each priority level has a target resolution time at line 1 and line 2 that is set by ICT. The department reports that in many cases these targets aren't met. Therefore, our main motivation of this research is to make resolution time predictions for the tickets to support the ticket handling process by providing the ticket makers accurate resolution time predictions that will lead to customer satisfaction and prevent disappointment that occurs when targets are not met.

1.4 Objective

The main objective of this research is to produce resolution time predictions for the tickets by taking a queueing theory perspective on the process of ticket resolution of ICT department in LUMC. Our aim is to present a methodology which incorporates process mining and queueing theory techniques to provide support for the ticket resolution process. In our research, we reveal answers for the following research questions.

Main Research Question: How long will an arriving ticket given its category and priority stay open?

Research Sub-Questions: There are several sub-questions to be tackled in order to make resolution time predictions for tickets.

- How is the flow of the tickets among operators/operator groups?
- What are the routing probabilities between operators/operator groups?
- What is the probability that an operator can resolve a ticket?
- How long does an operator serve a ticket?
- How long does a ticket is delayed?
- How can the category and the priority of a ticket affect the above questions?
- On average how long does a ticket stay in the system?
- Can the ICT department resolve the tickets within targets?
- How to improve the performance of the ticket resolution process in terms of meeting targets?

1.5 Thesis Outline

The incident management process which handles the incidents that arrive at ICT department as tickets, the ticket resolution process which is the subject of our research, research questions, objectives and motivations are described in chapter 1. A literature review on the topics of process mining, ticket resolution, queueing theory with process mining and predicting times in processes is given in chapter 2. A brief introduction on process mining is given in chapter 3. The chapter 4 describes the research approach of the thesis. The analyses and the results by process mining the ticket resolution process are given in chapter 5. The queueing network model of the ticket resolution process is described in chapter 6. The queueing analysis of the queueing network model is conducted in chapter 7. The approach of predicting resolution times and implementation results are given in chapter 8. Conclusions and recommendations for the ticket resolution process are given in chapter 9.

2 Literature Review

In this chapter, we present the reviewed literature that we consider relevant to our research. In our research, we aim to analyze a ticket resolution process in an ICT department and provide resolution time predictors to support the process. Our main objective of this research is to incorporate queueing theory and process mining to analyze and support the ticket resolution process. Therefore, we reviewed the literature on process mining, ticket resolution, queueing perspective process mining and techniques to predict time in processes.

2.1 Process Mining

Process mining can be applied to a variety of processes in healthcare, business, industry, etc. when an event log which contains the recordings of process instances is available. Here we present the literature on the applicability of process mining in discovering information from event logs about processes.

The work by Aalst et al. [30] demonstrates an application of process mining to a real-life process in one of the provincial offices of the Dutch Public Works Department. With taking control-flow, organizational and case perspectives, the processing of invoices sent to an office is analyzed. To mine for a process perspective, authors use a heuristic approach to deal with the noise and incompleteness of the log.

Process mining is a high-potential tool for gaining insights of healthcare processes which are mostly too complex to comprehend due to their dynamic, multi-disciplinary and ad-hoc nature. However, due to complexities of healthcare processes, it is difficult to obtain good process models with process mining for healthcare processes. A process mining methodology which uses sequential clustering to pre-process event logs is developed in [19] to identify regular and exceptional medical cases of clinical work-flow.

Process mining is applied to a Dutch Hospital in [15] in ProM framework to gain knowledge about the care-flow of gynecological oncology patients. Related event logs are extracted from the hospital's information system and used to perform process mining with control-flow, organizational and performance perspective. For control-flow perspective mining, authors use Heuristic Miner to tackle the noisy data and to focus on main flow. Trace Clustering plug-in is applied to the log to split the data into several clusters. Social Network Miner is used to analyze relations between originators using handover of work metric. Dotted Chart and Basic Performance Analysis plug-ins are employed for performance perspective.

The study [22] constitutes an application of process mining in a healthcare process for conformance checking purposes. Here, process mining is used as a mediator between clinical reality that is seen in the event log and the clinical guidelines. The medical guidelines, de jure model, are repaired based on reality to obtain a de facto model with a methodology based on cross validation.

An another application of process mining is done in [23] for the test process of wafer scanners in ASML with ProM framework. Log inspection, filtering and process discovery are performed to analyze and suggest improvements for the test process.

Process mining is applied to provide on-line decision support in [18] for the running cases of a process by forecasting the future events. An event log which contains completed instances is used as a training set to learn a predictive clustering tree (PCT) off-line and the learned PCT-based process model is used to forecast the next events for running cases with an on-line manner.

In a study by Krinkin et al. [9] to predict traffic overload when a network topology changes in wireless mesh networks and to recommend redistributions, process mining is applied to extract information about the network topology.

In a study [14], the availability of medical data due to use of ICT tools in hospitals and the necessity of making use of this data to increase the efficiency of the healthcare processes are pointed out. Process mining is applied on two data sets, clinical data set and pre-hospital behavior data set, for stroke patients to gain meaningful insights about the processes. Process models of the treatment processes of two hospitals which show if the clinical guidelines are followed and where the process deviates from the clinical guidelines are extracted with process mining from the clinical data sets. Performance analysis plug-in in ProM is used to identify the bottlenecks in the pre-hospital behavior process. They conclude that process mining can provide interesting results regarding the processes within hospitals.

A study by Aalst et al. [34] demonstrates an application of process mining to detect security violations by analyzing audit trails. Alpha algorithm is used to mine a work flow (WF)-net from audit trails and 'token' game is played on the net for some cases to detect anomalies in the handling of the process. Conformance checking is done by comparing the fragments of the process with the extracted WF-net of the process. Authors conclude that process mining techniques such as α algorithm can be applied to check security.

In [24] it has been showed that process mining can be applied to obtain software process models from the Software Configuration Management Systems (SCM) which store information of the software processes. Software processes of a real project, ArgoUML, are mined with process mining techniques in ProM and some properties of the software processes are analyzed and verified.

Process mining is used in mining staff assignment rules in [13]. Decision tree learning method is used to learn staff assignment rules by using organizational model and historical data as input.

From the literature on the applications of process mining we find that the most common use of process mining is to discover process models from the event logs of executed instances of the processes. The discovered process models from the event logs show what is happening in reality. The main purpose of process mining the historical data of instances of the processes is found to be discovering meaningful insights regarding the processes. The one of the most meaningful insights about the processes is to find out if they comply with the reality. The process models that are mined with process discovery techniques are utilized to discover deviations from the pre-described processes and pre-defined guidelines.

In the literature, we find that ProM framework is the most common software for performing process mining. Apart from the process discovery and compliance checking plug-ins, performance plug-ins are commonly used in the literature to identify bottlenecks of the processes. In our research, we also use the process mining tools in ProM to perform process mining.

Process mining is applied in several healthcare processes inside hospitals to discover how the processes are actually executed, to learn where the processes deviate from clinical guidelines and to identify the bottlenecks of the healthcare processes. In our research, we aim to use process mining to gain insights of an ICT service process inside a hospital, the ticket resolution process at the ICT department in LUMC, from the historical data.

2.2 Ticket Resolution

Here we present the literature review on ticket resolution. We find that there are rare studies which analyze the ticket resolution processes of IT services and this literature mostly focuses on efficient routing of the tickets among groups. In this literature, Markov Models and Generative Models are used to model the routing of the tickets.

In a study [16], the problem of bouncing tickets among multiple expert groups is pointed out. A methodology to reduce bouncing of the tickets and thus improving the resolution time is presented. Using the content and the routing sequence of tickets, generative models that characterize the life-cycle of tickets are developed. From the content of the problems of the tickets that group solved previously for each expert group a resolution profile is built with Resolution Model. The transfer model that is presented in this work on the other hand considers ticket routing sequences to build the profiles of transfer of tickets between every two expert groups. Lastly, an optimized network model which uses both content and routing sequence of tickets is presented and a numerical approach is taken to get an approximate solution, the next expert group to route the ticket, for this model.

A hybrid methodology which uses both text content and routing sequence to make routing recommendations for new tickets based on the information gained in historical ticket data is developed in [29]. Here, improving the efficiency of ticket routing is measured in terms of mean number of steps until the resolution. Their methodology, first identifies the content of a new ticket and finds a set of existing tickets which are similar in content, and then a weighted Markov model is created from the routing sequence of the tickets in this set.

Another study on efficient ticket routing by Shao et al. [28] provides a Markov Model that captures the ticket routing decisions by mining the ticket resolution steps without using the content data of the ticket. A search algorithm is given to make recommendations based on the Markov Model.

Process mining is adapted to produce predictions for the performances of the ticket resolution instances in [6]. In this study, instead of relying on the typical activity labels of a ticket resolution process such as open/close a ticket, forward a ticket, or send a message, etc., or the individual workers involved in the resolution of the tickets, or the pre-defined groups of workers, authors discover data-driven groups which are the abstractions of the sequence of workers involved in the tickets from the log data. Their approach employs both resolution steps and context data of the tickets. The context data of the tickets are used to cluster cases. With clustered data-driven worker groups and clustered ticket cases, performance predictions are given for different scenarios of the ticket resolution process.

In our research, we do not consider the ticket content data, the description of the problem in the ticket and also the content of the activities that are performed by operators. We only make use of the resolution sequences, operators that a ticket visits on its path from start to resolution similarly to [28]. In all of the literature that we reviewed on ticket resolution, recommendations that aim efficient routing of the tickets are generated. However, making recommendations for ticket routing is out of the scope of our research, we only model the routing behavior and predict the resolution times of the tickets.

2.3 Queueing Theory and Process Mining

In our research, we aim to combine queueing theory and process mining. Here we review the literature which incorporates queueing theory and process mining to learn about the methods and approaches to incorporate these two concepts.

In a study [25] which presents methods for checking conformance and improving performance of scheduled multi-stage service processes, process mining is applied to discover a Fork/Join queueing network of the scheduled processes from the real data. Process mining techniques are incorporated in identifying the network structure, in estimating the routing mechanism, and in characterizing the server structures.

Process mining and queueing theory together are applied in [38] for a healthcare process. A method which uses process mining to discover a process model and queueing theory to analyze queue measurements such as queue length and waiting time per activity is described in this study. The process model is obtained by the Inductive Visual Miner tool in ProM. The methodology described in this study to extract waiting time for activities requires an event log with both start and end time of activities to obtain average service times from the process model. The average service time and the number of patients from the process model are used to obtain predictions for arrival rate, queue length and waiting time for each activity.

Online delay prediction problem is tackled in [27] with queueing perspective process mining using event log of a call-center. This study proposes three types of delay predictors: transition system predictors, queueing model predictors and snapshot predictors. The transition system approach considers queueing phase and service phase of an activity as two separate steps. In this approach, cases are assumed to be independent and the system load, number of customers in the queue, is included as context information. Queue length predictor and queue length Markovian predictor, queueing model predictors, are based on the G/M/n and G/M/n+M model, respectively. Two heavy traffic predictors are presented: last customer to enter service and head of line.

The study in [26] presents methodologies to extract queue lengths from an event log per activity when the event log misses timestamps such as enqueueing timestamps and/or service start timestamps of activities. They first partition the log based on duration of the activity with clustering, then loads are used to fit a phase type distribution to sojourn times by using Bayesian inference.

In studies [38], [27] and [26], process mining and queueing theory are integrated with an approach which uses process mining to discover a process model from an event log and regards each activity in the discovered process model as a queue. Mean queueing performance measures of the activity queues are used in [38] and [27] to provide predictions for the queue lengths, waiting time, etc, of the activities. We find out that incorporation of process mining and queueing theory in [25] is not as direct as in studies [38], [27] and [26]. In [25], process mining is used to shape and compose a queueing model of the subject process.

2.4 Predicting Time

Here we present the literature on predicting time in processes. In the literature we reviewed on prediction models for processes, we find methods which are based on transition systems, queueing theory, Markov models and regression analyses. Process mining is found to be a widely-used technique to obtain a process model of a process from the event log of executed instances. In some cases, machine learning techniques such as decision trees and support vector machines are incorporated to learn from process models to obtain likelihood measurements of executing a future activity. It has been observed that most of the studies concentrates on making predictions for running cases, not for arriving cases.

2.4.1 A Similar Process: Customer Contact Centers

The case process of our research is an ICT ticket resolution process. In this process, the service of an operator for a ticket consists of several text replies and this service is not continuous as a call. The processing of tickets by operators is close to the processing of emails by agents at customer contact centers in which customer contact is made via e-mails rather than calls. An arrived customer request via an e-mail to a customer contact center is handled among the agents at the contact center and the e-mail follows a path among the agents until an agent resolves the e-mail.

The processing of the incoming e-mails by the customers to a customer contact center is analyzed in [3] with a queueing network of agents who process the e-mails. With two-moment queueing approximations the network is analyzed and performance measures such as average number of e-mails in the system and average resolution time are obtained.

This study models the process as a queueing network of agents and gives the mean sojourn time in the system which can be regarded as a prediction for the stay of the arriving emails in the process. In our research, we can model the ticket resolution process as a queueing network of operators and obtain mean sojourn time of the system which we can provide as prediction for the resolution times of arriving tickets. The weakness of this prediction is that it does not consider the workload of the system.

2.4.2 Recent History Predictions

Recent delay history is used in [7] to make delay predictions for an arriving customer. The used estimators are: the delay of the last customer to enter service, the delay experienced by the customer at the head of the line, and the delay experienced by the customer to have arrived most recently among those who have already completed service are compared with the queue length estimator of delay.

The approach of using the recent history can be applied in producing predictions for the resolution time of the tickets by presenting the resolution time of the most recently resolved ticket as the prediction for the resolution time of an arriving ticket in an on-line manner. This approach considers the system workload by taking the recent history into account.

2.4.3 Process Mining Based Predictions

Process mining techniques are used in [31], [10] and [17] to produce prediction models that are trained with the historical data of the executed process instances. In these studies, remaining times of the running cases in the processes are predicted. In predicting the time perspective of the cases in processes, issues of predicting the future activities and predicting the activity durations are resolved with process mining and machine learning techniques.

When timestamp information exists in the event log, it is possible to take a time perspective in process mining. One of the first studies which focuses on time perspective is the study by Aalst et al. [31] which presents a transition system based methodology to predict remaining processing time for running cases. The transition system is generated from historical data of executed instances using event abstraction. A completion time prediction for a running case is made based on the average time to completion for the cases which are in the state which corresponds to the current state of the case.

In study [10], predictions for the likelihood of executing future tasks for a semi-structured business process are made from instance-specific probabilistic process models. Using a set of historical traces, a process model is mined. Decision tree learning, a machine learning technique, is used at every decision point in the process model. Instance-specific process model is constructed for a running process instance using the decision trees to compute one-step transition probabilities. The instance-specific probabilistic process model is then transformed into a Markov chain, and Markov methods such as absorption and first passage time probabilities are applied to obtain the likelihood of executing future tasks.

Several approaches which rely on process mining and machine learning are developed in [17] to predict the remaining processing time for running cases. The prediction methods are developed with simple regression, regression with contextual information and data-aware transition systems approaches. Event logs of past executed process instances are used for training, and prediction models are implemented in ProM framework as a plug-in.

2.4.4 A Stochastic Petri Net Prediction

An approach which relies on stochastic Petri nets and considers the passed time since the last event is presented in [20] to predict the remaining execution time. By replaying event logs on the Petri net, statistical information on the activity durations is collected to be able to fit statistic parametric and non-parametric distributions for the activity durations. This study presents a remaining time prediction for running cases for a given time based on past executed activities on the cases. The prediction algorithm starts with replaying a given running case on the Petri net to obtain the initial marking. For a pre-defined number of iterations, the algorithm gathers the remaining time simulation results and calculates the mean of all.

3 Process Mining

In this chapter, we introduce process mining to the readers. In the first sections, we introduce the types of process mining, the things that can be discovered by it and the tools to do it. An illustration of discovering information by process mining and quality criteria for the discovered process models are given at the last sections of the chapter.

3.1 What Is Process Mining?

Process mining as a new discipline is positioned between data mining and process modeling. The main aim of process mining is to extract information from event logs for discovery, monitoring and improving purposes for a real process [4]. Process mining shows what is actually happening in a process based on recorded instances of the process itself. The necessary ingredient for process mining is an event log which consists of cases, process instances, and events that are performed on cases. With the existence of event logs of a process, four types of process mining can be conducted: discovery, conformance, enhancement and operational support.

Discovery Process mining can be used to discover a process model only from the observed behavior in the event log. A process model describes the steps that are taken in the process. A common algorithm for discovery is the alpha-algorithm which produces a Petri net, a common process modeling language that allows to model concurrency, of the process from an event log. This algorithm is able to produce a Petri net without using any a-priori knowledge. In case the event log contains information about the resources who perform activities, social-networks which reflect how people work together can be found as well [4].

Conformance Process mining is also used for checking if the existing process model conforms with the event log of the process. Conformance checking can be used to detect, locate and explain deviations from the guidelines and rules that are described in the existing process model [4].

Enhancement Here process mining is used to improve the process model by projecting the information extracted from the log into the process model. The model is enhanced by making it closer to the reality [21].

Operational Support Differently from other 3 types of process mining, operational support is not done offline. Here, operational support can be given to running cases in an online setting through detecting deviations and generating alerts, predicting future and informing, and recommending activities or resources [33].

There are 4 process mining perspectives: control-flow perspective, organizational perspective, case prespective and time perspective.

- **Control-Flow Perspective** This perspective aims to produce a Petri net or some other notation that describes the control-flow of activities based on the flow of cases in a log. Focus is on the ordering of activities and the dependencies between activities [4].
- **Organizational Perspective** This perspective focuses on the resource, the person/department who executes the activity. When an event log contains resource information of the events, role of resources and relations between resources can be extracted with organizational perspective [4].
- **Case Perspective** Case perspective focuses on the case properties. A case can be characterized by its activity path or by the resources performing activities [4].
- **Time Perspective** When an event log includes information about the timestamp of events, with the time perspective one can discover bottlenecks of the process, analyze service times of activities and predict the remaining time for running cases [4].

3.2 What Can Process Mining Discover?

There are several questions about the processes that process mining can answer from event logs which possess at least a case identifier, an activity name, a timestamp, and a resource who performs the activity for each event. The listed questions in [36] are grouped by perspective as the following.

- **Control-Flow Perspective**
  - How are the cases actually being executed?
  - Which activities precede which other activities?
  - Are there concurrent activities?
  - Are there loops?
- **Organizational perspective**
  - How many people are involved in a specific case?
  - Who subcontracts work to whom?
  - Who work on the same activity?
  - How many transfers happen from one role to another role?
  - What is the communication structure among people?
- **Case perspective**
  - What are the most frequent paths?
  - What is the distribution of cases over the paths?

Plug-in Name	Function
Alpha Miner	Discovers a Petri net from an event log by using the α- algorithm
Heuristic Miner	Discovers a C-net
Fuzzy Miner	Discovers a fuzzy miner
Transition System Miner	Discovers a transition system from a given event log with chosen trace abstraction
Transition System Analyzer	Makes a prediction for the remaining time based on transition system
Log Filter	Filters the log based on a criterion
Dotted Chart Analysis	Represents all events in a dotted chart
Social Network Miner	Discovers social network between resources

Table 3.1: Plug-Ins in ProM

- **Time perspective**
  - What is the average, minimum, maximum throughput of cases?
  - What is the average service duration for an activity?
  - How much time is spent between two activities?

3.3 Tools

The ProM framework which is developed by Eindhoven University of Technology is an open-source standard process mining platform which includes all types of mining techniques. Due to its high functionality, ProM is an expert tool for process mining, however there exist other commercial process mining tools like Disco such that non-experts can perform mining too. ProM is a plug-able environment which uses MXML and XES as input format for event logs. It is also possible load csv files as there is a conversion tool which converts csv files to XES. The most widely used mining techniques which are plugged in ProM are: alpha-algorithm, heuristic miner, fuzzy miner, genetic miner, social-network miner, etc. Apart from mining plug-ins, there are other plug-in tools in ProM for loading, converting, filtering and splitting logs and also for visualizing logs [4]. Some of the present mining plug-ins in ProM 6 which are mentioned in [4] are given in table 3.1.

3.4 From Event Logs to Process Discovery

Event logs can be considered as the starting point of process mining. Basically, an event log is defined as the multi set of traces while the activities that are executed on a case define the trace of the case. Apart from the activities and the related case, often event logs contain information regarding the resources, persons who perform the activities, and the timestamp of the events, preferably the start and end time of the activities [33].

A sample event log is created to illustrate the application of process mining with ProM on extracting information from an event log. First thing we do is to apply alpha algorithm, one of the most common process model discovery algorithms, to obtain a Petri net of the process from the recorded process instances in the log.

case id	activity	timestamp	resource
1	A	2017-03-17 08:00:12	R.1
2	A	2017-03-17 10:20:13	R.2
1	B	2017-03-17 16:50:10	R.3
3	A	2017-03-17 12:12:23	R.1
2	C	2017-03-17 13:45:56	R.4
3	B	2017-03-17 14:16:78	R.3
1	D	2017-03-17 18:00:23	R.1
2	D	2017-03-17 18:05:31	R.1
3	D	2017-03-17 18:10:13	R.1

Table 3.2: A Sample Event Log

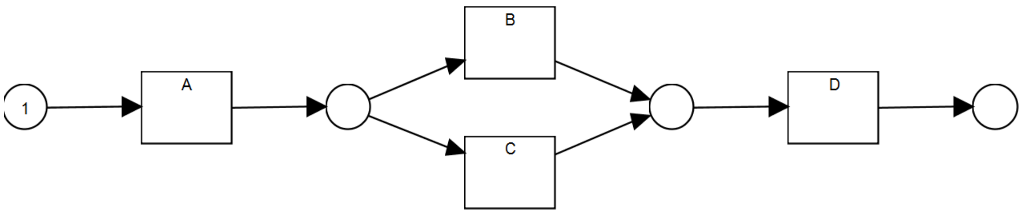


Figure 3.1: The Petri Net Obtained by Alpha Algorithm

Definition 3.4.1. A Petri net is a triple N=(P, T, F) where

- P is the finite set of places,
- T is the finite set of transitions,
- $F \subseteq (P \times T) \cup (T \times P)$ is the set of directed edges describing the flow relations.

The Petri net in figure 3.1 summarizes the behavior of the process that is observed in the log. In this process, all cases start with activity A, and then either activity B or activity C follows. Lastly, activity D is executed before the case closes.

The α algorithm, one of the first process discovery algorithms that is proposed by Aalst et al. [32] discovers a process model of a given log by discovering the ordering relations between the activities that are performed in the process. We describe how this algorithm discovers the Petri net in figure 3.1 from the sample event log in table 3.2.

α algorithm discovers relations between activities from the traces in a log by defining the following ordering relations between activities [4].

Let a, b be the activities in a log L.

- *Direct Succession*: $a >_L b$ if and only if there is a trace in which a is followed by b.

- *Causality*: $a \rightarrow_L b$ if and only if $a >_L b$ and not $b >_L a$, in some cases a is followed by b but b is never followed by a.

- *Parallel*: $a||_L b$ if and only if $a >_L b$ and $b >_L a$, in some cases a is followed by b and in some cases b is followed by a.

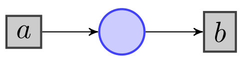
- *Choice*: $a\#_L b$ if and only if not $a >_L b$ and not $b >_L a$, a is never followed by b and b is never followed by a.

According to these relations we obtain the footprint of our log as the following.

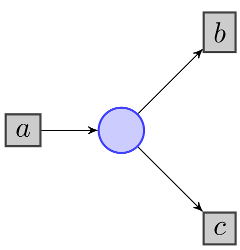
	A	B	C	D
A	$\#$	$\rightarrow$	$\rightarrow$	$\#$
B	$\leftarrow$	$\#$	$\#$	$\rightarrow$
C	$\leftarrow$	$\#$	$\#$	$\rightarrow$
D	$\#$	$\leftarrow$	$\leftarrow$	$\#$

Table 3.3: Footprint of the Sample Log

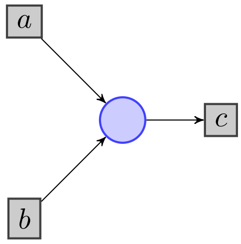
α algorithm uses the footprint of a log to discover process patterns that are required to construct a process model.



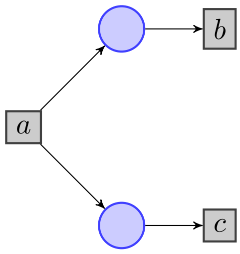
sequence pattern $a \rightarrow b$



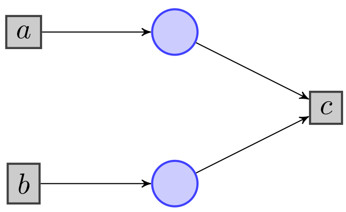
XOR-split pattern $a \rightarrow b$, $a \rightarrow c$, and $b\#c$



XOR-join pattern $a \rightarrow c$, $b \rightarrow c$, and $a\#b$



AND-split pattern $a \rightarrow b$, $a \rightarrow c$, and $b|||c$



AND-join pattern $a \rightarrow c$, $b \rightarrow c$, and $a||b$

From the footprint of the sample log, α algorithm discovers a XOR-split pattern among the activities a, b and c as $a \rightarrow b$, $a \rightarrow c$, and $b\#c$, and a XOR-join pattern among the activities b, c, and d as $b \rightarrow d$, $c \rightarrow d$, and $b\#c$ that result in a Petri net in figure 3.1.

We apply social network miner in ProM to answer questions relating to the social relations between the persons who perform the activities. We mine for a subcontracting of work and obtain the answer for the question of who subcontracts work to whom and we mine for the handover of work to understand who handovers work to who.

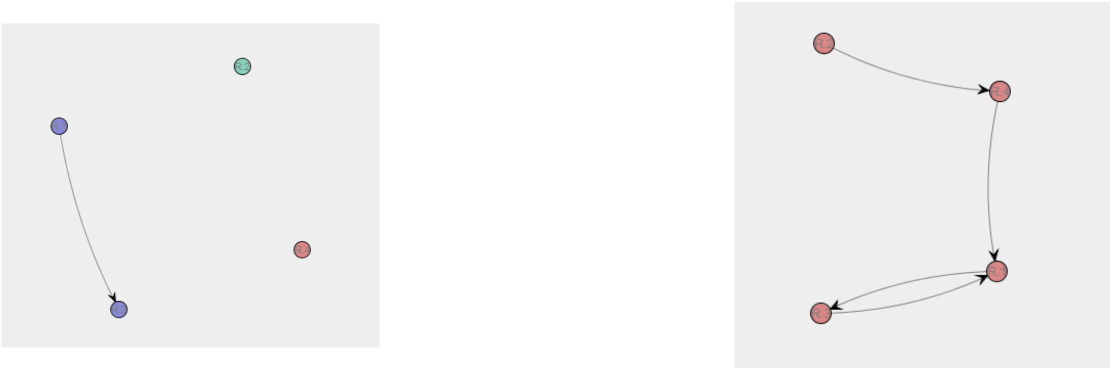


Figure 3.2: Subcontracting Work Network and Handover of Work Network

Process mining is also used for making performance analysis. We mine our event log with the inductive visual miner plug-in in ProM to find the frequencies of the activities.

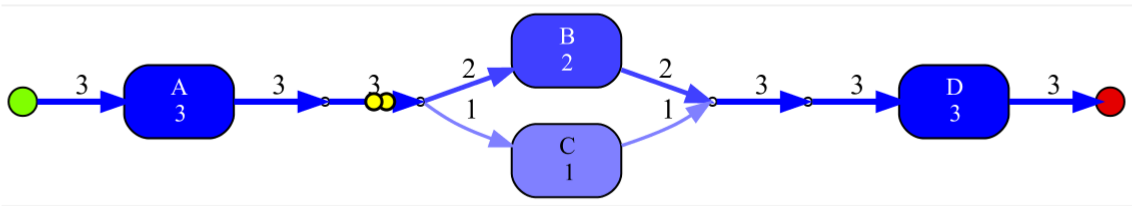


Figure 3.3: Activity Frequencies with Inductive Visual Miner

3.5 Quality in Process Model Discovery

In this section, we discuss the quality criteria of discovered process models from event logs with process discovery algorithms.

Let L be an event log of instances of a process. A process discovery algorithm maps L onto a process model M which represents the behavior seen in L. The quality of a discovered process model M is measured by how good M represents the behavior seen in L. The four main quality criteria for discovered process models that are described in [5] are the following.

Fitness: M allows the behavior seen in L. M is a good fitting process model if M can replay most of the cases in L.

Precision: M is precise such that it does not allow behavior that is completely unrelated to the behavior seen in L.

Generalization: M generalizes the behavior seen in L.

Simplicity: M is simple such that it is the simplest process model that can explain the behavior seen in L.

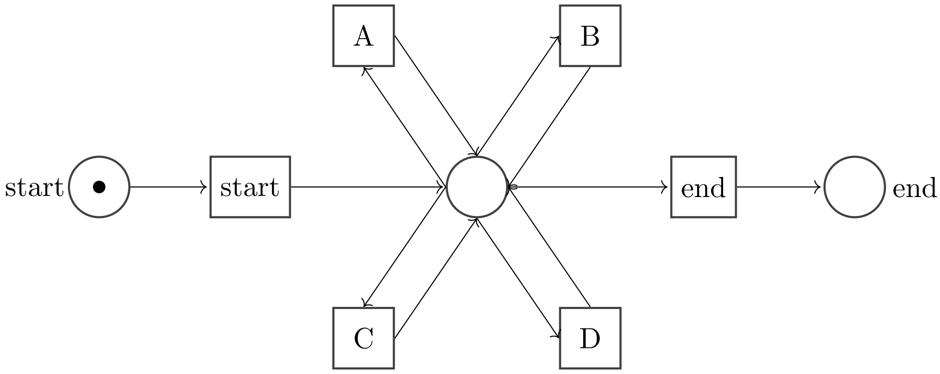


Figure 3.4: Flower Model of the Sample Log

A good process model which represents the behavior seen in an event log can be discovered if the event log contains the representative behavior. An event log which contains infrequent behavior or does not contain enough to describe the behavior is considered as less representative. Event logs which are not able to represent the behavior of the processes are a challenge for the process discovery techniques.

There is a trade-off between the quality criteria of discovered process models from event logs. It is difficult to balance these four quality criteria. For instance, a model which overly simplifies lacks in fitting and precision, and model which is perfect in fitness often lacks simplicity [5].

Let represent the sample log given in table 3.2 as $L = \{< A, B, D >^2, < A, C, D >^1\}$. We output a process model for L which allows every possible behavior among all activities seen in L and name it flower process model as the shape resembles a flower. The flower process model of the sample log is described in figure 3.4. We compare the process model which α algorithm discovers in figure 3.1 and the flower process model with these four quality criteria. The two instances seen in the log, $< A, B, D >$ and $< A, C, D >$, fit in both models, so both models are perfect in fitness. The model that α algorithm discovers is more precise than the flower process model as the flower model allows unrelated behaviors. Both models are simple. Both models are not good in generalization. The flower model over generalizes the behavior, basically any instance which involves activities A, B, C and D can be replayed. On the other hand, the model that α algorithm discovers under generalizes as the allowed behavior is restricted to the behavior seen in the log.

All four quality criteria can be quantified with metrics. In the literature, several metrics are proposed and used for assessing the quality criteria: fitness, precision, generalization and simplicity. An overview of the state-of-the-art metrics used to quantify quality criteria to asses discovered process models is given in [8].

Process discovery algorithms are successfully applied to extract process models which represent the behavior in event logs. Although, there are criteria described to assess the quality of discovered process models and several metrics are developed to quantify each criterion, there is a lack of a common framework which assesses the goodness of discovered process models [35]. The study [1] states the need for a common evaluation framework for assessing the outputted process models that are discovered with process mining algorithms and presents an evaluation framework that aims to benefit the process mining researchers by enabling to assess the results of process discovery algorithms and to discuss the validity of the results obtained by process mining techniques.

4 Research Approach

In this chapter, we state our research goals and describe our research methodology.

4.1 Research Goals

In our research, we aim to model the ticket resolution process at ICT department based on the historical data of ticket resolution activities, analyze the performance of the ticket resolution process in terms of resolution times and provide resolution time predictions for the arriving tickets. The main objective of our research approach is combining process mining and queueing theory to build a data-driven model of the ticket resolution process and making use of process mining techniques and mean queueing performance measures to investigate the efficiency of the ticket resolution process. Besides, in producing resolution time predictions we aim to utilize mean queueing performance measures. For an arriving ticket, we aim to produce predictions based on the ticket classification information of priority and category.

4.2 Research Methodology

From the starting point of our research, the event log containing ticket resolution activities, to making predictions for the ticket resolution times, we describe our research methodology in figure 4.1.

- **Log Preparation:** ICT department does not have an event log for the ticket resolution process. To perform process mining for the ticket resolution process, we define activities and compose an event log of ticket resolution activities.

- **Process Mining:** We aim to apply process mining to gain insights of the ticket resolution process from the data. We make use of process mining techniques for several purposes. Firstly, we use process mining to analyze the current performance of the ticket resolution process and to discover social relations among operators. Secondly, we use process mining to give us an idea of the actual process so that we can build a fit data-driven model of the ticket resolution process. Lastly, we use process mining tools in obtaining input parameters for our model of ticket resolution process.

- **Modelling:** We model the ticket resolution process as a queueing network of operators following the study [3] which models the process of resolution of emails to a customer contact center as a queueing network of agents.

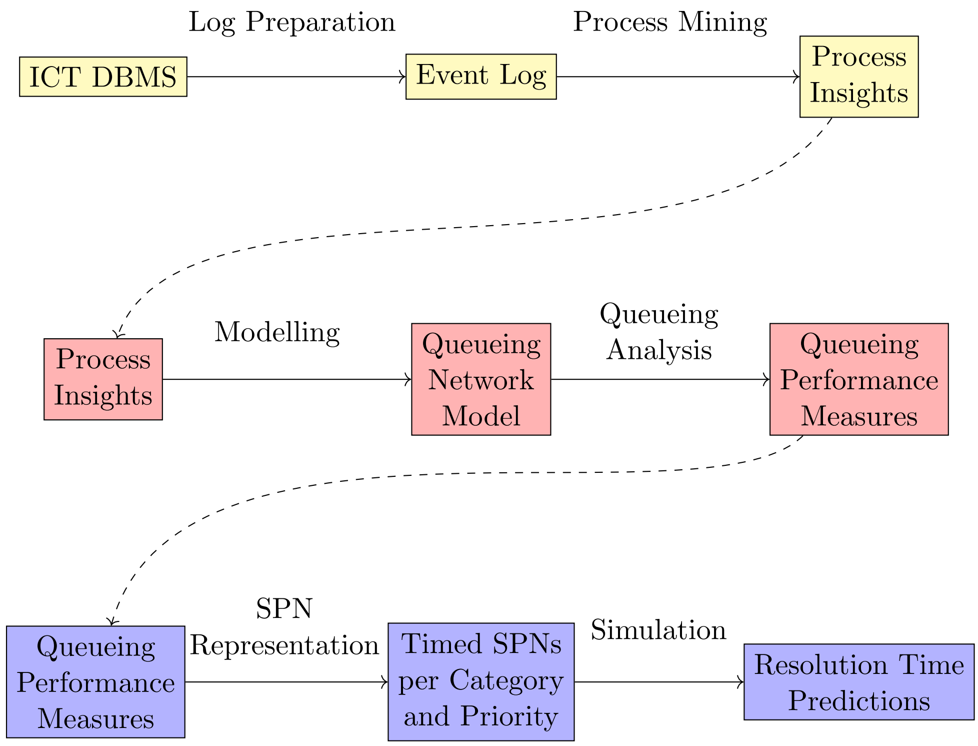


Figure 4.1: Overview of Research Approach

- **Queueing Analysis:** We obtain performance measures about the ticket resolution process by performing queueing analysis and obtaining mean queueing performance measures.
- **SPN Representation:** We make use of stochastic Petri nets (SPNs) in our queueing perspective resolution time prediction approach which takes ticket priority and category information as inputs. We use the mean queueing performance measures of the queueing network model in order to take a queueing perspective resolution time prediction approach. We represent per ticket category and priority a SPN of the ticket resolution process to obtain resolution time predictions per ticket category and priority.
- **Simulation:** We simulate the stochastic Petri nets given an initial marking, ticket category and priority to produce resolution time predictions. Our approach of using SPNs to obtain resolution time predictions is based on the study in [20].

4.3 How Are Process Mining and Queueing Theory Combined?

In our research approach, process mining and queueing theory are combined in the modeling phase. We model the ticket resolution process as a queueing network model of operators and make use of process mining to structure the queueing network similarly to the study [25]. We use process mining tools to understand the structure of the servers, to obtain queueing input parameters such as transition probabilities and service times in the queueing network model. Process mining techniques are utilized in our research to build a data-driven queueing model of the ticket resolution process.

5 Process Mining Ticket Resolution Process

In this chapter, we present the steps of performing process mining the ticket resolution process at the ICT department in LUMC and the analyses of the process by process mining tools in ProM.

5.1 Event Log Preparation

Process mining takes an event log as input and discovers information about the process based on the process instances in the event log. In order to apply process mining for a process, an event log which contains the instances of the process is required. So, the first step of process mining is preparing an event log.

In order to process mine the ticket resolution process, we need an event log which captures the activities performed for resolving the tickets. We obtain our event log by relating the tables which contain the activities that are performed on the tickets and the operators who performed the activities on the tickets in the database at the ICT department.

Our event log consists of activities that are performed on the tickets by operators and operator groups. Operators respond to the tickets in a platform called 'TopDesk'. The activities on a ticket consist of emails and resolution activities, the text entries that are sent by operators to resolve the ticket. Email activities on a ticket are performed when communicating with the ticket notifier, the person who creates the ticket, to learn more about the problem, communicating with other operators to get their help on the resolution of the ticket and assigning the ticket to another operator or operator group.

Both the content of the emails and the resolution activities on the tickets are long texts which are related to the problems that are described in the tickets. The tickets which are similar in content are related to similar problems and probably they have similar resolution activities. Therefore, the content information can be used to understand the routings of the tickets and make predictions. For instance, the studies [16] and [29] use the content of the tickets to model ticket resolution processes. However, in our research we do not make use of the text content of the resolution activities and the text content of the problem descriptions, similarly to the work in [28], we only make use of the ticket resolution sequences, operators and operator groups who performed activities on the tickets to model the ICT ticket resolution process. Therefore, in our event log we use operators and operator groups who acted on the tickets as activity names. Thus, the traces of the tickets in our log describe the flow of the tickets among operators/operator

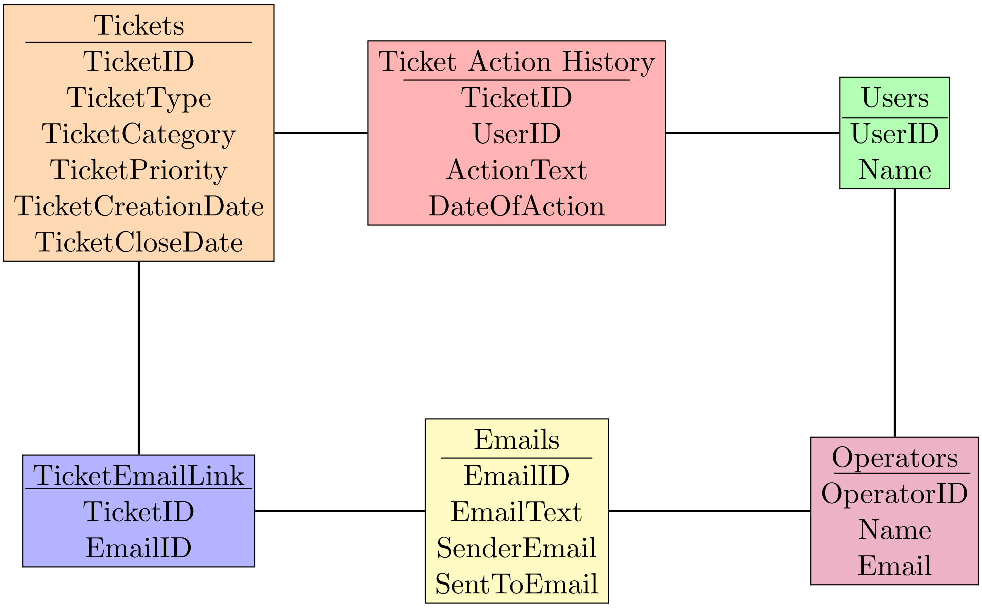
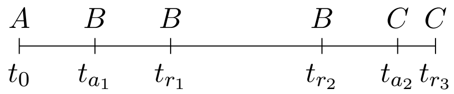


Figure 5.1: Database Diagram of Topdesk Tables

groups.

Each day lots of tickets arrive at the ticket handling system with different categories, priorities and types. Therefore, we include the ticket classification data, the type of the tickets, the priority level of the tickets, and the category of the tickets, in the log to perform filtering. Namely, our event log includes the tickets as cases, the operators/-operator groups who executed the tickets as activities, the timestamps of the activities and the emails that are executed on the tickets, and the ticket classification information such as the problem category, problem subcategory, ticket type and ticket priority.

Among the email activities on the tickets, we only include the ones which are sent for the purpose of assigning the tickets to operators or operator groups in our log. These assignment emails and together with resolution activities, our log has the timestamps of when an operator or operator group is assigned for a ticket, when an operator or operator group is performed the first resolution activity, and when an operator or operator group either resolved a ticket by performing the last resolution activity or assigning a ticket to another operator or operator group. In this way, we are able to infer the time when a ticket arrives at an operator, the time that an operator responses to a ticket (the first resolution activity by the operator), and the time that the ticket exits an operator (the last activity by an operator before a ticket exits the system or is assigned to another operator). An example of the timeline of the resolution and the assignment email activities that are executed on a ticket T is illustrated in figure 5.2.



- t_0: The ticket is registered by operator A.
- t_{a_1}: The ticket is firstly assigned to operator B.
- t_{r_1}: The first resolution activity is executed by operator B.
- t_{r_2}: The second resolution activity is executed by operator B.
- t_{a_2}: The ticket is assigned to operator C.
- t_{r_3}: The third resolution activity is executed by operator C and the ticket is resolved.

Figure 5.2: Resolution and Assignment Emails on a Ticket

The number of events that are executed per ticket and the amount of tickets that arrive over time leads to lots of records. In order to overcome this, we only take the data of the tickets that are created in a month, specifically the tickets that are created in January in the year 2015, to process mine and produce predictions.

Our log contains both operators and operator groups who perform activities on the tickets. In our research, we refer both operators and operator groups as operators for convenience.

In this research, we assume that the time that a ticket is assigned to an operator A is the time that the service duration of the ticket at A starts. The service duration of the ticket at A ends with the last subsequent activity by A. Also, there exist resolution activities by operators which are not initiated by an assignment event. For those, we assume that the service time begins with the first resolution activity and ends with the last resolution activity before the ticket leaves the process or an another operator executes an activity on the ticket.

The service times of the operators are shown in figure 5.3 for the example ticket that is described in figure 5.2.

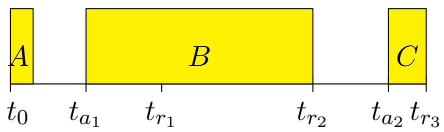


Figure 5.3: Service Times of Tickets

In order to obtain the service times of operators on each ticket in the event log based on our assumptions, we apply the ProM plug-in 'Merge Subsequent Events' to prepare our log for process mining. This plug-in merges a trace 'ABBBCC' into 'ABC' and composes service start time and service end time from the timestamp of the first event and the timestamp of the last event, respectively for each activity in the trace.

5.2 Analyzing Efficiency with Process Mining

Before moving to discovering process models and social relations with process mining, we analyze the log to investigate the current efficiency of the ticket resolution process with visualization plug-ins in ProM.

5.2.1 Basic Statistics

The log visualizer gives basic information about the log such as the number of cases, events and activities in the log, the minimum, maximum and average number of events and activities per case. We reach the following information from our log which contains operators who handle tickets as activities.

Number of Tickets	Number of Events	Number of Operators
5937	20518	255

Log Visualizer	Min	Mean	Max
Number of events per ticket	1	3	63
Number of operators per ticket	1	2	18

Table 5.1: Basic Statistical Information of the Log

5.2.2 How Many Operators Are Involved?

The number of operators involved in the resolution of the tickets is an indicator of the efficient routing of the tickets among operators and operator groups. For instance, in [29], the mean number of steps to resolution is used as a measure to define the efficiency of the ticket routing. Considering that the routing of a ticket to an operator who is not able to resolve the ticket is going to delay the resolution of the ticket, the number of operators involved per ticket is also an indicator of the efficiency of the ticket resolution process in terms of the resolution time.

The log visualization provides us the basic statistical information on the number of operators and operator groups who are involved in the resolution of the tickets. We observe that on average the tickets are terminated with resolution with the involvement of two operators, but on the other hand there are some tickets which are routed among significantly many operators. The existence of such tickets indicates an inefficiency in the routing of some tickets to the right operators and operator groups who are able to close the tickets.

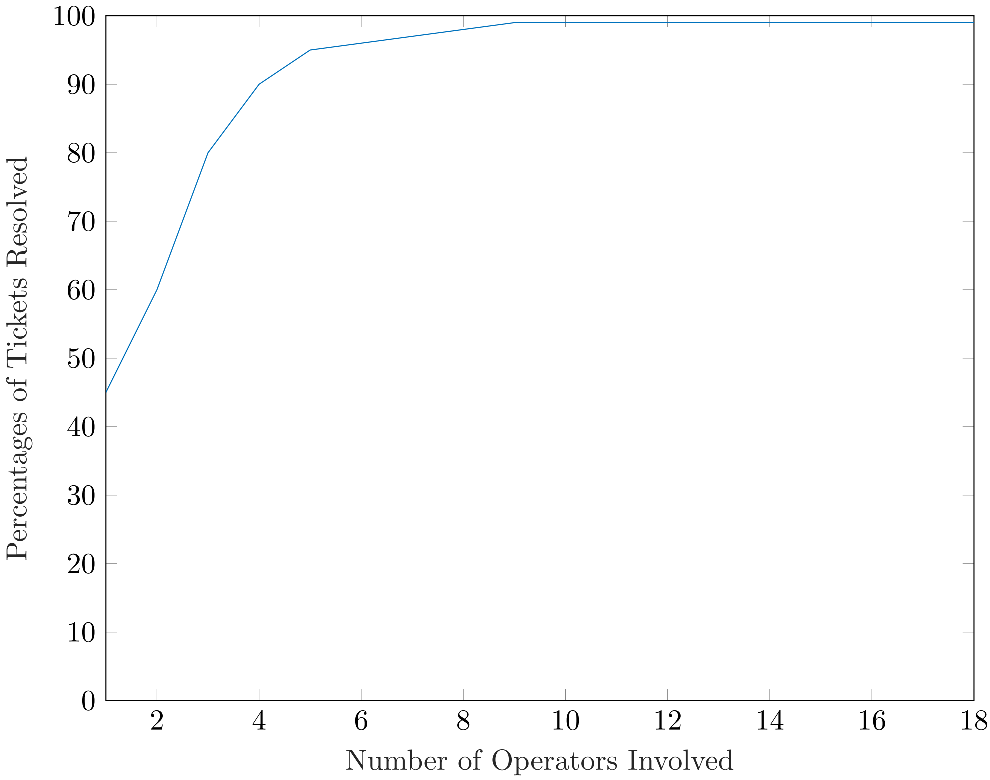


Figure 5.4: Number of Operators Involved In Resolution

60% of the tickets in our log are resolved with the involvement of maximum two operators and with the involvement of maximum three and four operators 80% and 90% of the tickets are resolved, respectively. For the 10% of the tickets minimum five operators are involved to resolve the tickets.

5.2.3 Who Closes a Category with Log Summary Tool

We filter our log by category and obtain which operators are involved in resolving a specific ticket category with the help of log summary tool. Log summary also provides the frequencies of the activities, so we learn how frequently an operator is involved in resolving the tickets which belong to a specific category. Apart from the information of who is involved in resolving the tickets of a specific category, the information of who resolved the tickets of a specific category, namely who closes the tickets of that category is important. The frequencies of the end events in log summary tool provides information of which operators are involved in closing the tickets of a specific category and how likely that the tickets of that category are going to be closed by a specific operator.

We filter our log by category and obtain a log which contains only the activities of the tickets which belong to 'Laptop' category and obtain the information of which operators have closed the tickets of 'Laptop' category and how frequently. We observe that the likelihood of an operator who is involved in the resolution of 'Laptop' tickets to resolve 'Laptop' tickets differs a lot per operator. Also, we observe that for a significant portion of the operators to whom 'Laptop' tickets arrive, their likelihoods of resolving the tickets of 'Laptop' are very low. The routing of the tickets to operators who are not likely to resolve them indicates inefficiencies in the ticket routing.

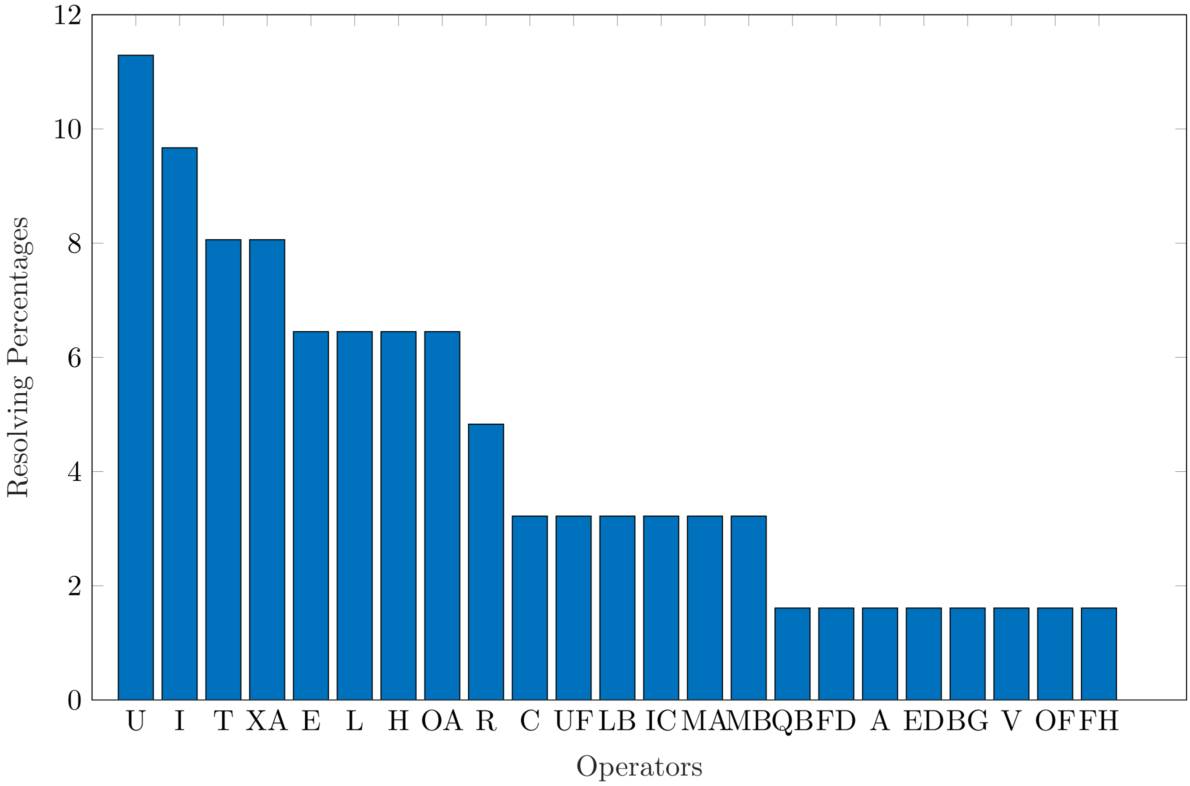
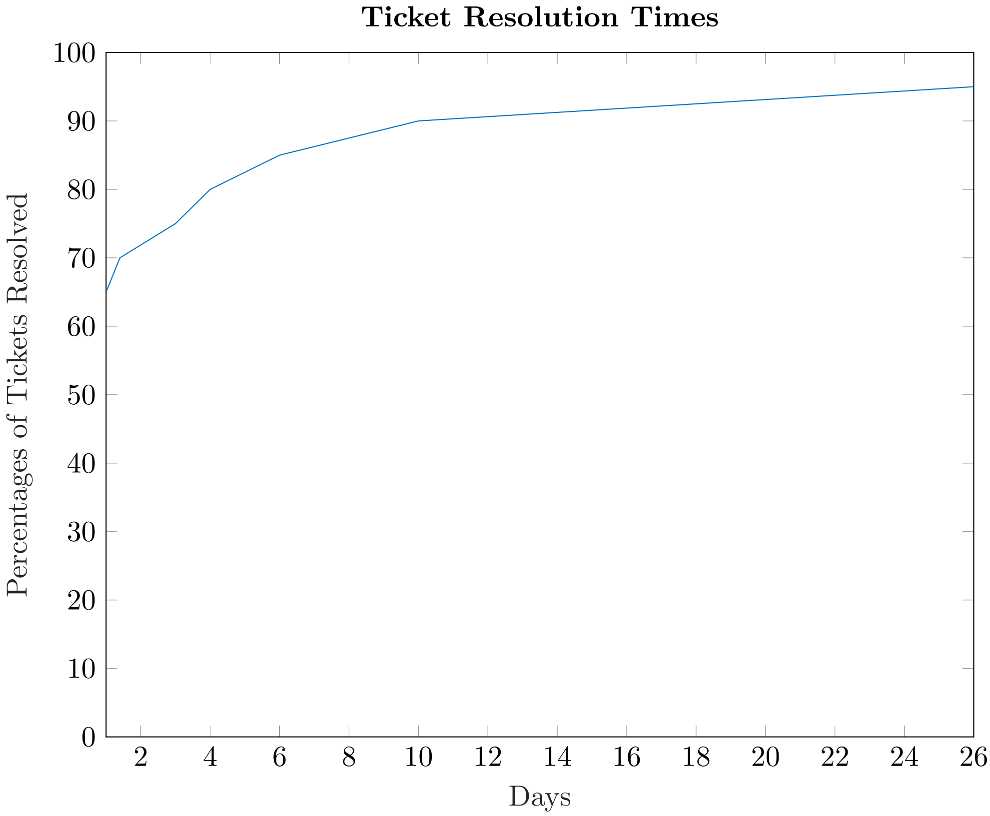


Figure 5.5: Resolution Percentages by Operators of 'Laptop' Tickets

5.2.4 How Long Do Tickets Stay Open?

With the help of time based log filtering we filter the traces by their durations and find out that 65% of the tickets are resolved within 1 day, 80% of the tickets are resolved within 4 days and 95% of the tickets are resolved within 26 days. Also, we observe that there are some exceptional tickets which have resolution time over a year. Recall that the ICT department considers the percentages of the tickets that are not resolved within one month as a KPI, KPI 3, to identify bottlenecks of the ticket resolution process. We observe that approximately 5% of the tickets in our log are not resolved within one month.



5.2.5 Are the Targets Met?

We analyze by filtering the log per priority and escalation information and using time based log filtering to see in how much of the tickets in our log the target resolution times given in table 1.4 are met. Note that KPI 2 which measures the extent to which the target resolution times are met is the main performance measure that we consider in our research.

escalated	priority	target resolution time	percentage meeting target
no	5	1 week	90
yes	5	2 weeks	88
no	4	1 day	81
yes	4	1 week and a day	90
no	3	1 hour	28
yes	3	3 days and an hour	85
no	2	30 mins	18
yes	2	1 day and a half an hour	75

Table 5.2: Percentages of Tickets Meeting Targets

We observe that short target resolution times such as 30 minutes and 1 hour are unrealistic targets to meet even though the tickets have high priorities. For the rest, we observe that the target resolution times are met for approximately 85% of the tickets.

5.2.6 Arrival Patterns with Dotted Chart Tool

We make use of the dotted chart analysis plug-in in ProM to investigate the arrival patterns of the tickets. The dotted chart shows the distribution of events in the log over time in a glance. This tool can be used to understand the arrival patterns of the activities in a log over time. We sort the tickets based on the timestamps of their first events and obtain figure 5.6 with Dotted Chart plug-in. We observe a weekly arrival pattern for the tickets in the log, namely there is a steady arrival of the tickets during the weekdays and the arrivals of the tickets stop in the weekends. It can be seen from the dotted chart that the first set of activities on the tickets are performed in succession and after some point in time the activities are distributed sparsely. This can be interpreted as that in the ticket resolution process if a ticket could not resolved within a certain period in which the first set of successive activities are performed, after that period the waiting time between the resolution activities increases leading to a long resolution time.

The axes of the dotted chart can be changed to any field that describes the events or traces. We obtain the dotted chart given in figure 5.7 with priority as an axis to learn the arrival distributions of the priority levels. We observe that the log mostly consists of low priority tickets: normal and important. There is a continuous arrival of the tickets with normal priority and the arrivals get sparser as the priority level increases.

In order to understand the arrival patterns of the tickets which belong to a specific category, it is possible to filter the events by category and then visualize the filtered log with the dotted chart tool in ProM. We use dotted chart tool to analyze the arrival rates of 'Laptop' category tickets that are created in the year 2015 (see figure 5.8). Throughout the year, in the time periods of two months we observe similar arrival rates of tickets except in the period from July 2015 to September 2015 in which lesser arrivals are observed.

5.3 Process Model Discovery with Inductive Visual Miner Tool

In order to gain insights of the process of ticket handling we make use of the Inductive Visual Miner, a process exploration tool that is developed in [11], to discover a process model. Apart from discovering a process model from a given event log, Inductive Visual Miner is capable of animating the log on the discovered model and visualizing deviations. Filtering is another feature of the Inductive Visual Miner plug-in, it is possible to filter events and traces to simplify the model. Inductive Visual Miner also shows performance information of the process model, such as frequencies, average service times, average waiting times and sojourn times of the activities. We use the performance information that Inductive Visual Miner tool in ProM provides to obtain the queue input information to build the queueing network model of the ticket resolution.

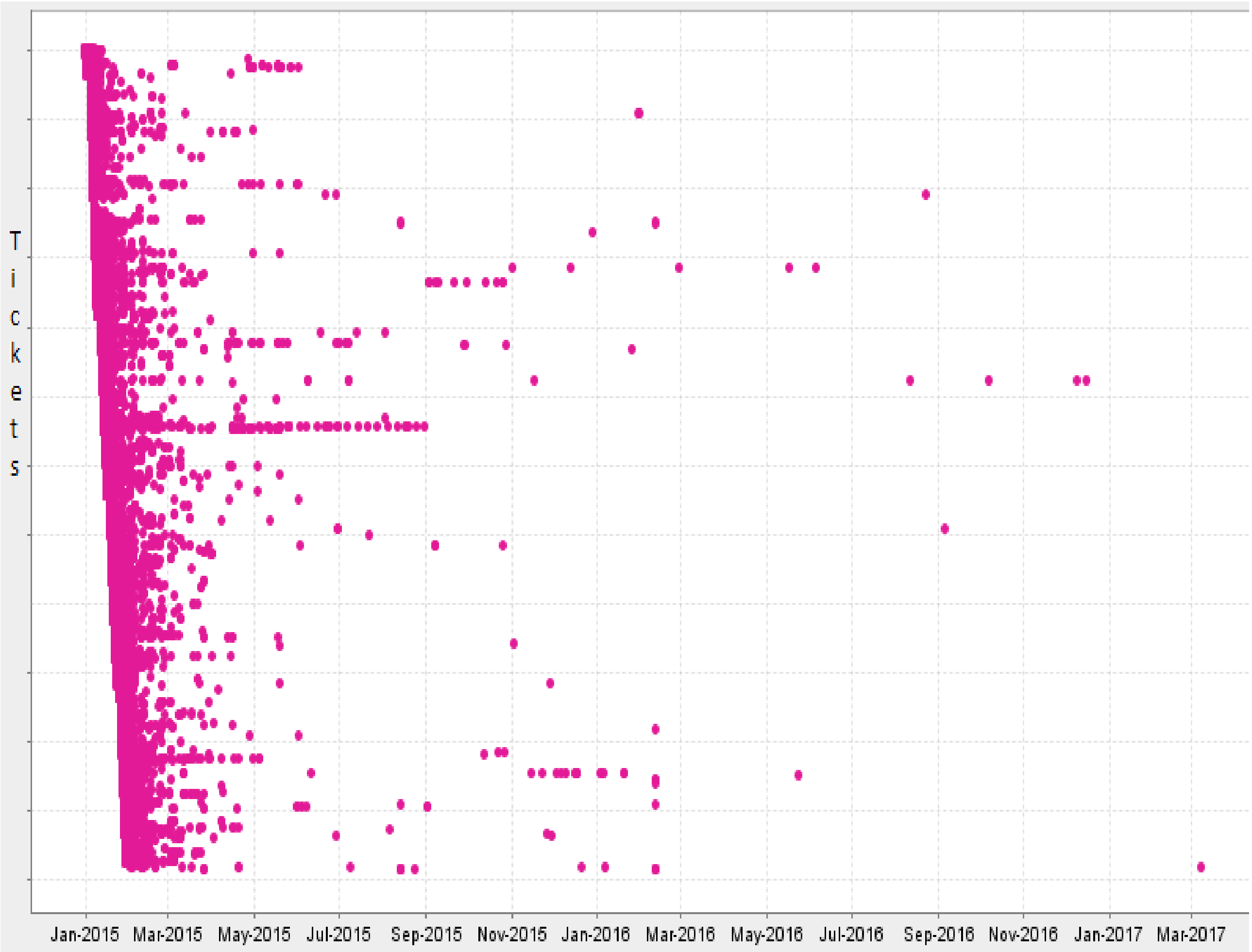


Figure 5.6: Dotted Chart of Tickets Over Time

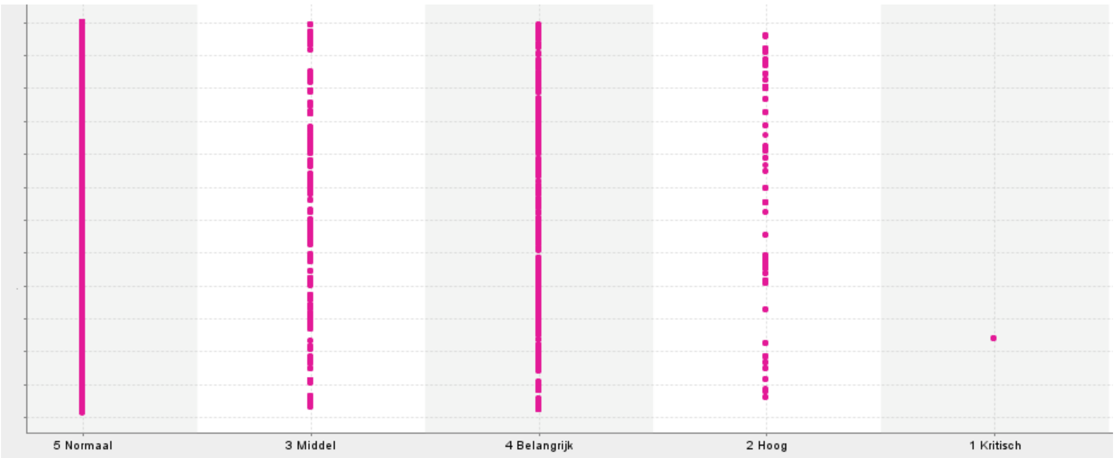


Figure 5.7: Dotted Chart of Priorities Over Time

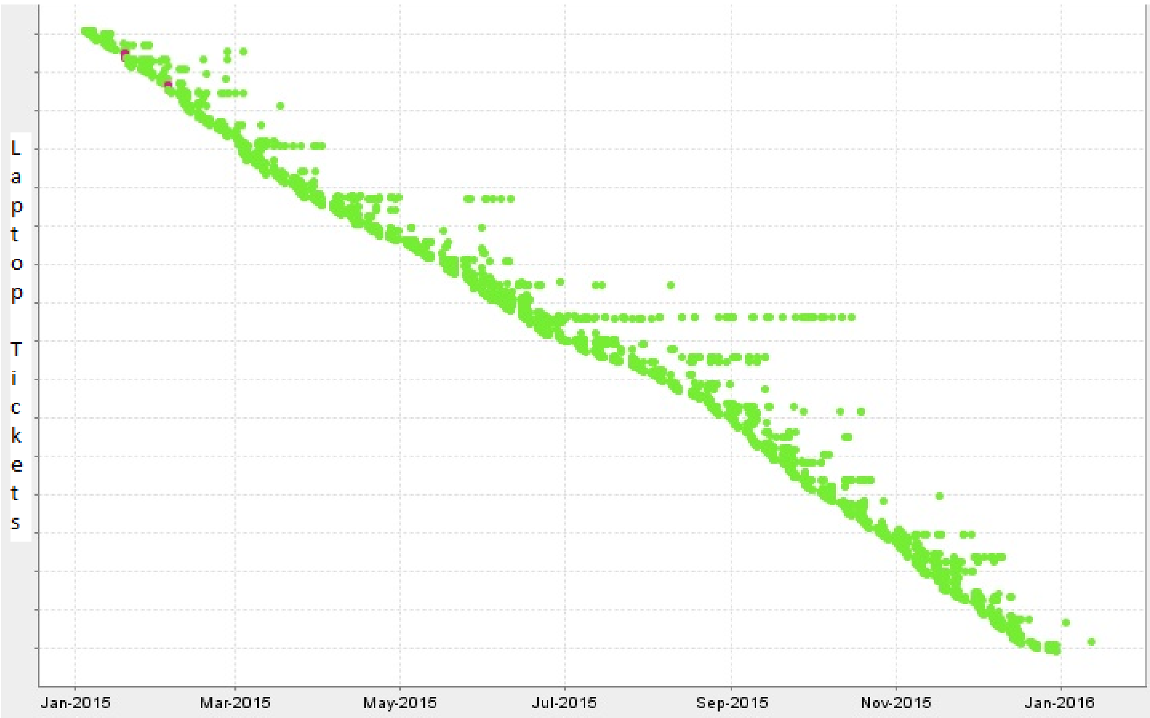


Figure 5.8: Dotted Chart of Laptop Category Tickets Over time

First, we start to explore the log with Inductive Visual Miner plug-in. Without applying any pre-mining filters, the process model that is discovered in figure 5.9 by the Inductive Visual Miner is unstructured. The discovered process model allows a vast number of paths for tickets. After the execution of an activity there is no insight into which activity is going to be followed. The discovered process model from our log indicates that the flow of the tickets among operators is not structured.

An operator serves the tickets which are assigned to him/her and the assignment of the tickets to an operator is done if the operator is specialized in the category of the ticket. For the unassigned tickets, every operator may choose to take a ticket voluntarily. So, if an operator serves a ticket which belongs to a category, then we may assume that there is a relation between the operator and the ticket categories that the operator deals with. Therefore, we filter the tickets by their categories and perform mining per category to learn the routing patterns between operators, service times of operators and waiting times between activities.

We use Inductive Visual Miner tool in ProM for the tickets that belong to the category 'Laptop'. The process model given in figure 5.10 that is obtained by Inductive Visual Miner indicates the variety of the possible paths that 'Laptop' tickets may follow among operators in the resolution process. The discovered process model of ticket category 'Laptop' involves significantly less activities, operators, than the discovered process model which is given in figure 5.9. This model is relatively has more structure than the process model which is given in figure 5.9. However, the discovered process model is still unstructured. There are no insights into which activities precede which activities.

We use the performance information that Inductive Visual Miner provides to assess the time perspective efficiency of the handling of 'Laptop' tickets and we obtain mean performance information which is given in table 5.3. The performance information of average waiting times indicates inefficiencies in responding to the tickets quickly as we observe long average waiting times for the tickets to start being served by some operators. We find that on average the tickets wait longer in transfer than in being served by the operators.

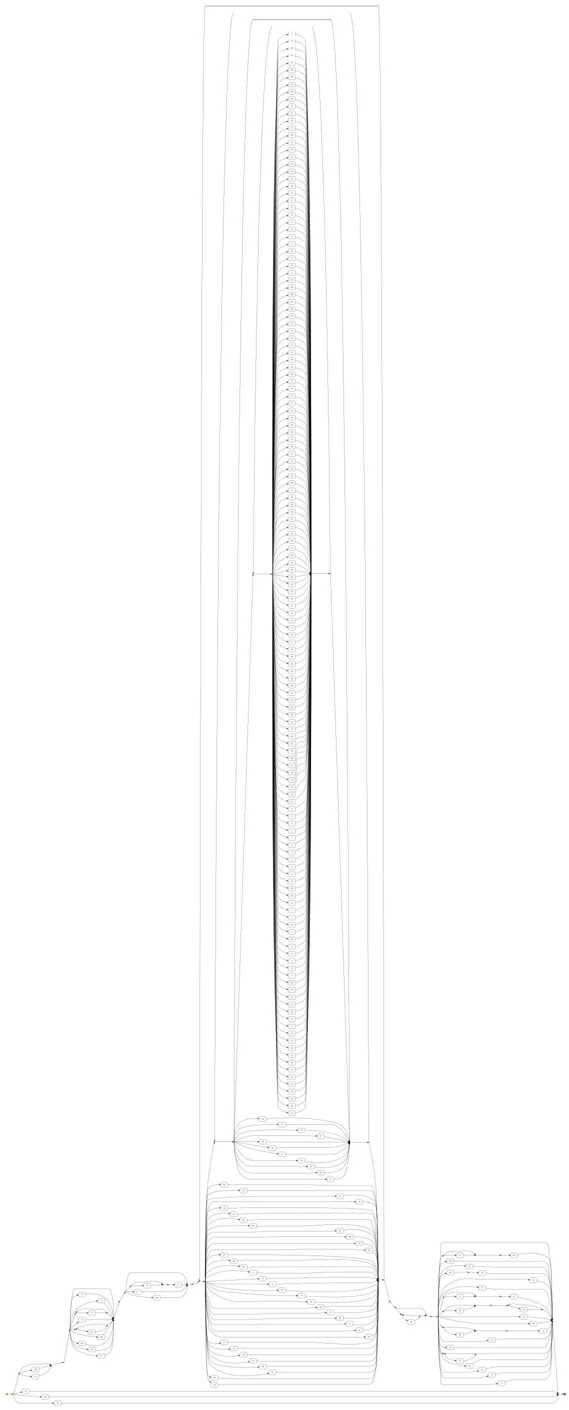


Figure 5.9: Process Model with Inductive Visual Miner

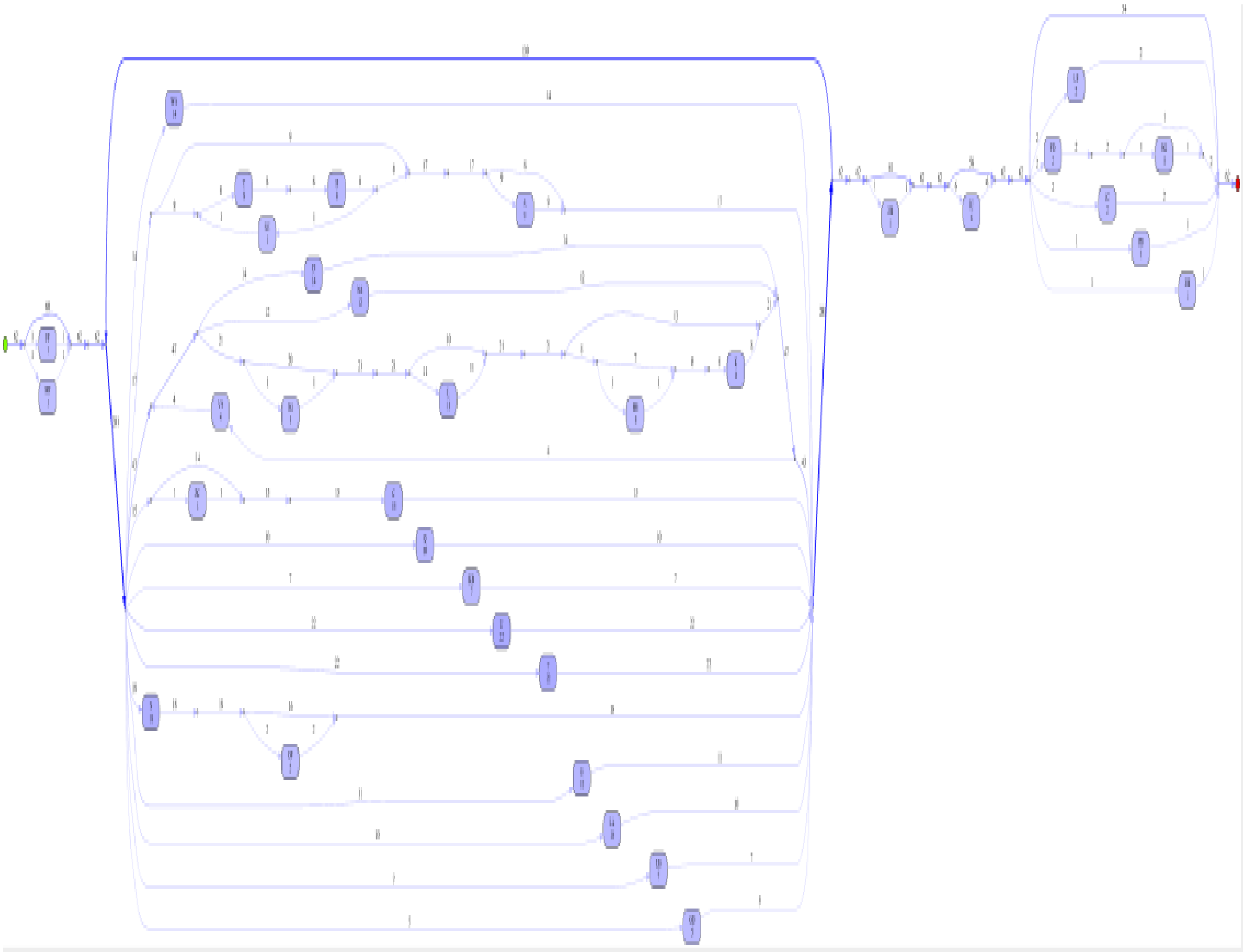


Figure 5.10: Process Model of Laptop Tickets

Operators	Num. of Occurrences	Mean Service Time (hour)	Mean Waiting Time (hour)
FF	1	0	0
MF	1	0	0
S	18	14.88	24.37
WA	14	0.32	29.25
ZC	1	0	169.73
VA	4	0	13.18
Y	8	0	0
XE	1	16.73	0
OF	2	250.47	10.98
SG	1	0	4.15
D	14	4.13	64.62
H	8	0	0.93
NA	12	1.03	12.58
C	15	33.78	22.1
Q	10	7.3	46.53
L	11	0	13.72
KB	7	0.78	16.93
U	22	0.97	11.07
A	9	0	9.12
T	22	34.82	7.73
G	11	0	26.25
LA	10	0.58	20.42
BB	1	0	765.93
LB	7	1.12	13.92
GD	5	0	206.6
K	8	0.62	25.57
ZB	1	0	71.8
IC	6	0	0.45
FD	2	0	0.55
UF	2	0.73	10.67
JC	2	0	48.32
PB	1	44.52	1.77
BG	1	3.17	15.32
EH	1	0	387.43
Mean	-	12.23	60.35

Table 5.3: Performance Measures of Laptop Tickets

5.4 Discovering Social Relations Between Operators with Causal Activity Matrix Tool

Process models are discovered by finding causal relations between activities. We take the operators who perform resolution activities on the tickets as the activities in our log. So, the discovered process models from our log describe the relations among operators. That's why we do not perform social network miner tool to discover social relations between operators.

Here we apply Casual Activity Matrix process mining tool to investigate the causal dependencies among operators in handling a certain type of category to find out more about the social structure among operators. This tool represents the causal relations between activities. Since in our log we have operator names as activities, this tool gives us the information of causal relations between operators. In this tool, the magnitude of the causal relation between any two activities are measured with a value which ranges from -1 to 1. A value of -1 in a cell of the causal activity matrix with row A and column B indicates that there is no causal relation from A to B, a value of 0 indicates that there is no evidence to confirm any relation from A to B, and a value of 1 indicates that there is a causal relation from A to B.

The metric which is used to measure the strength of the causality dependency between two activities that is described in [30] is as follows.

Let a and b be two activities in a log L, let $|a >_L b|$ be the number of times that activity a is followed by activity b in L, and let $a \Rightarrow_L b$ denote the causal dependency between a and b, then $a \Rightarrow_L b$ is defined by

$$a \Rightarrow_L b = \frac{|a >_L b| - |b >_L a|}{|a >_L b| - |b >_L a| + 1}. \tag{5.1}$$

In the colored representation of Causal Activity Matrix tool, if there is no relation found from the row activity to the column activity of a cell in the matrix, then the cell is colored with red, if there is no information about causal relation, then the cell is colored white and if there is found a casual relation, then the cell is colored blue. The closer a cell value gets to 1, it indicates stronger casual relation from the row activity to the column activity.

We discover causal relations among operators with causal activity matrix tool for 'Laptop' category tickets (see figure 5.11). This matrix shows causal relations between operators who are involved in resolving tickets of 'Laptop' category. In some cells of this matrix there are values between 0.5 and 0.75. For those cells, the value of the cell implies the strength of the causal relation from the operator in the row of the matrix to the operator in the column of the matrix. In the cases that there is a value which is close to 1 in a cell with row operator O_i and column operator O_j, causal relation measure implies that tickets after being processed by O_i are likely to be routed to operator O_j.

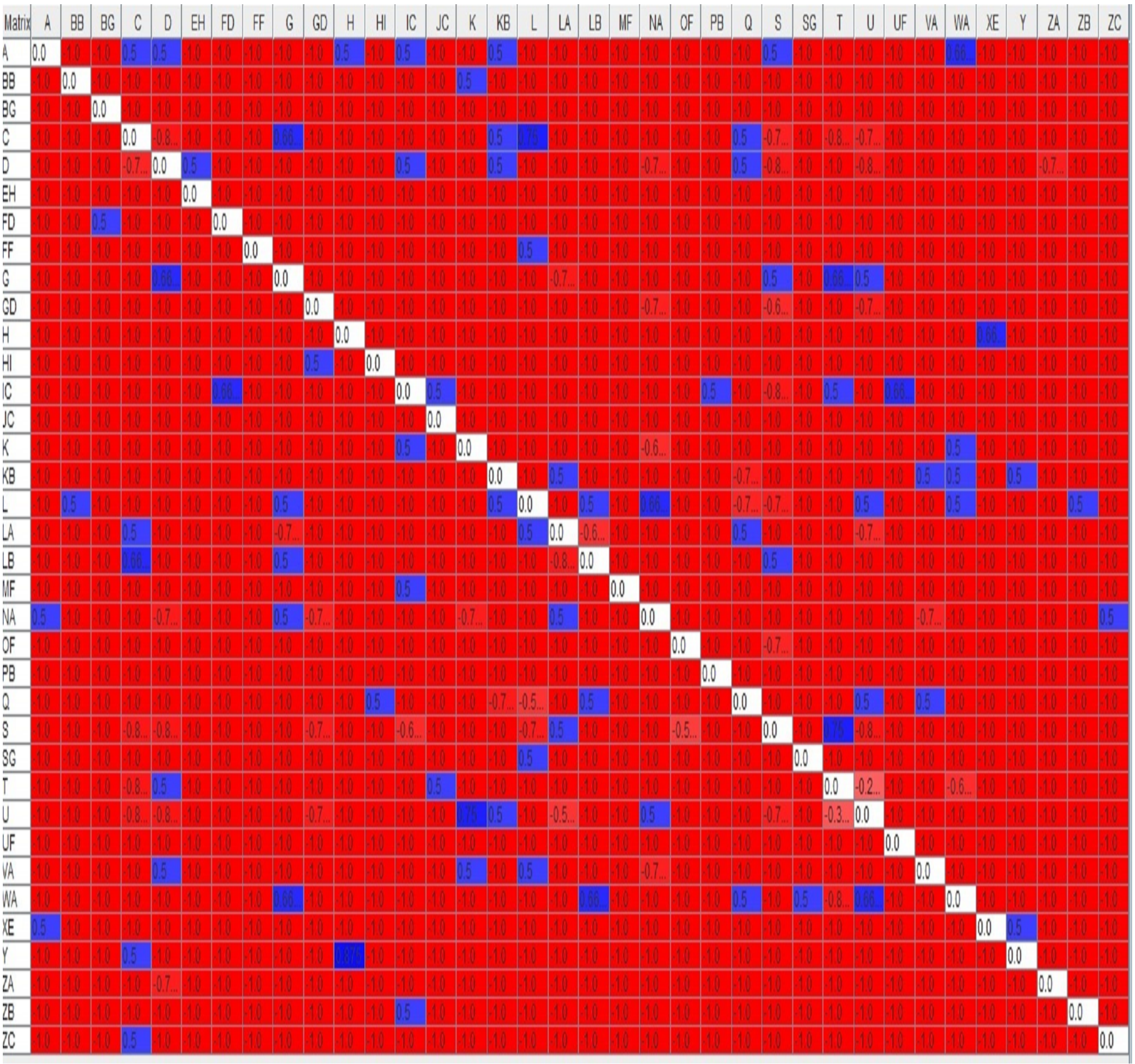


Figure 5.11: Causal Activity Matrix: Operators as Activities

6 Modelling Ticket Resolution Process

In this chapter, we present the queueing network model of the ticket resolution process, how we analyze the server structures with the help of process mining and data-drived model input parameters.

6.1 Process Insights

Our research aims to make use of process mining to gain insights of the ticket resolution process and to make use of the gained process insights to build the queueing network model of the ticket resolution process. From our log which contains operators as activities, we define the life-cycle of a ticket as a path that is followed among operators. When a ticket is transferred from an operator to another operator, the ticket is delayed and we treat the delay as an activity that a ticket goes through when the ticket is transferred. Thus, the life-cycle of a ticket consists of the resolution services by operators on the ticket and the delays between the resolution services. Namely, a ticket is either being served by an operator or is being delayed from the time that the ticket is first created to the time that it is finally resolved. Therefore, we take operators and delay as servers in our queueing network model of the ticket resolution process. In figure 6.1, the life-cycle of a ticket which has three resolution services by operators O_1, O_2 and O_3 is illustrated.

The discovered process model of the ticket resolution process shows that the tickets do not follow structured paths among operators and there are no insights into which activities precede which activities. With this insight, in our queueing network model of the ticket resolution process we allow every possible routing between any two operators via delay server.

By filtering the log by ticket categories, we observe less number of operators who are involved in the resolution and more structured processes. We consider the ticket categories as customer classes and define routing probabilities by categories in our queueing network model.

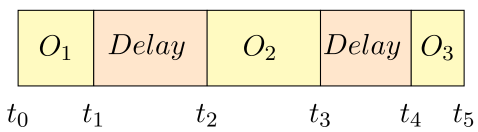


Figure 6.1: Life-Cycle of a Ticket

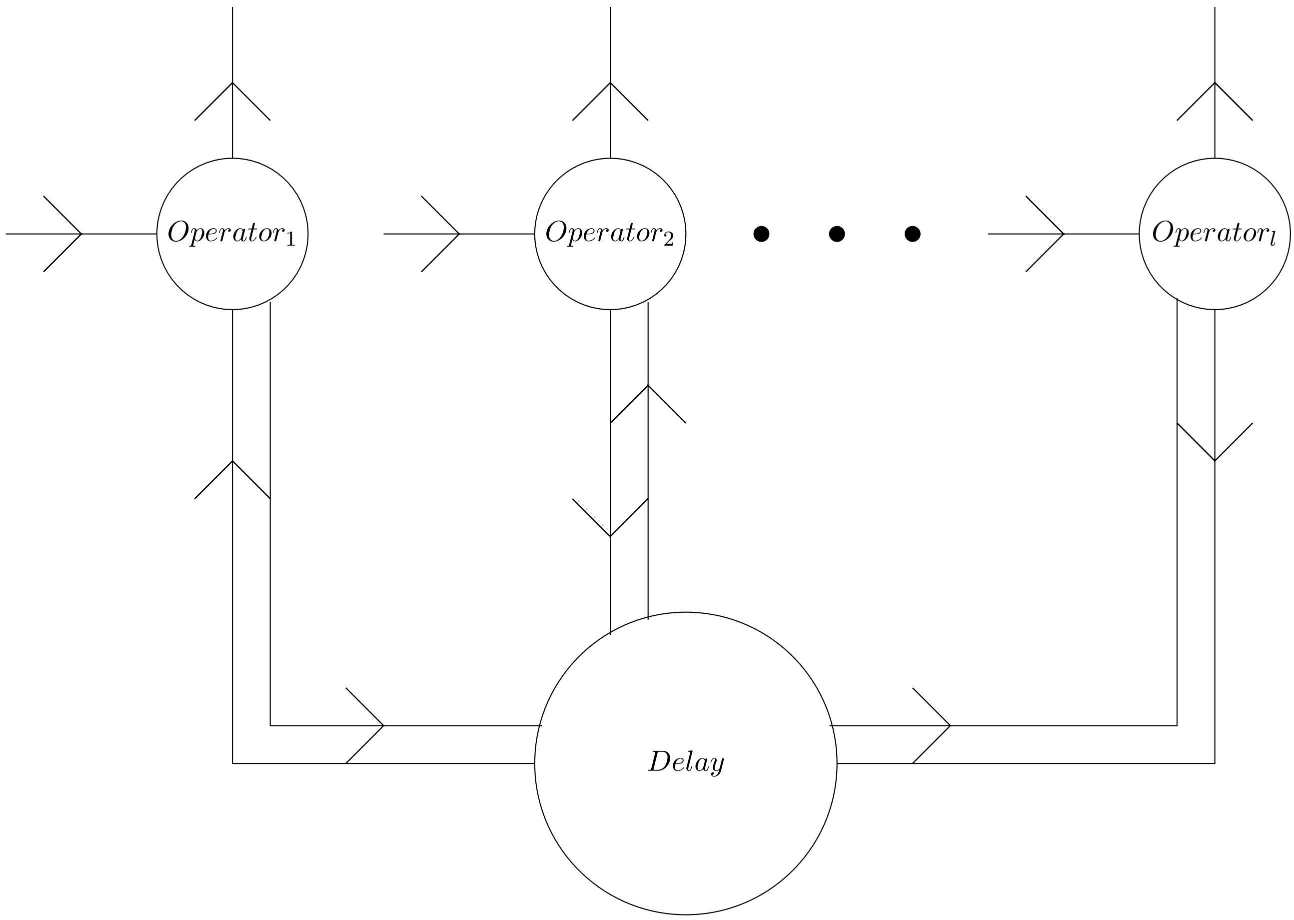


Figure 6.2: Queueing Network Model

6.2 Queueing Network Model

In our research, we consider the ticket categories as customer classes and model the ticket resolution process as a multi-class queueing network of operators and delay with ticket categories as classes. The categories that an operator is associated to, the categories that an operator is involved in the resolution, may arrive at the operator externally or from the delay node, transferred from an another operator. Upon the completion of a resolution service at an operator, a ticket may be resolved and exit the system or may be transferred to another operator via delay. A ticket is transferred among operators by visiting delay in every transfer until it is resolved.

In our research, we make use of process mining to investigate the server structures of operators and delay and also to obtain queue input information such as service times, transition probabilities, resolution probabilities, etc.

Notation
O: set of operators, with index i and j
T: set of tickets, with index t
C: set of ticket categories, with index k
P: set of ticket priorities, with index p
l: number of operators
m: number of categories
s: number of priorities
$C_{i,0}$: set of ticket categories that externally arrive at operator i
C_i: set of ticket categories that operator i is associated to
O_k: set of operators who are associated to ticket category k
$N_i = (n_1^i, n_2^i, ..., n_s^i)$: state of operator i where n_p^i, $p \in P$, is the number of p priority tickets at operator i
N: state of the system
$N_0 = (n_1^0, n_2^0, ..., n_s^0)$: state of delay where n_0^p, $p \in P$, is the number of p priority tickets in transfer
λ_k: arrival rate of ticket category k
$\lambda_{0_{i,k}}$: external arrival rate of ticket category k at operator i
$p_{i,k}$: probability that operator i resolves ticket category k
$p_{i,k,0}$: probability that operator i can not resolve ticket category k and transfers to delay
$p_{i,0,j,k}$: probability that ticket category k which is transferred from operator i is transferred to operator j
D_{p,N_0}: random variable representing the delay service time for a p priority ticket given the state of the delay
R_{i,p,k,N_i}: random variable representing the resolution service time of operator i for a ticket with priority p and category k given the state of operator i
$S_{i,k}$: random variable representing the resolution service requirement of a ticket with category k at operator i
$\mathbb{E}(S_{i,k})$: mean service requirement of a ticket with category k at operator i
$\mathbb{E}(D_{p,N_0})$: mean delay time of a ticket with priority p given the state of delay
$\mathbb{E}(R_{i,p,k,N_i})$: mean resolution service time of a ticket with priority p and category k at operator i given the state of operator i

6.3 Delay Server

In this section, we closely examine delays that occur when transferring tickets to make assumptions for the structure of the delay server.

We try to understand the factors affecting the delay that occurs when a ticket is transferred from operator O_i to operator O_j. The transfer of the ticket can be via an assignment activity or without an assignment activity.

Firstly, consider the case that the ticket is transferred from operator O_i to operator O_j via assignment. In this case, the delay that the ticket experiences is the duration between the last activity by O_i and the time that O_i decides that he/she can not resolve the ticket and assigns the ticket to O_j. ICT department handles tickets based on their priority levels and targets resolving tickets which have high priorities quicker than the tickets which have low priorities. We assume that when an operator O_i could not resolve a ticket and has to forward the ticket to another operator, O_i performs the assignment activity of the ticket, takes the decision that he can not resolve and forwards the ticket to another operator, quickly for the tickets which have high priorities.

Secondly, consider the case that the ticket is transferred from operator O_i to operator O_j without assignment and thus operator O_j takes the ticket voluntarily. In this case, the delay that the ticket experiences not only depends on the priority of the ticket also depends on the system load.

In our queueing network model of the ticket resolution process, we model the delay node as a server which has infinite capacity and thus as a server which serves all the tickets that are in transfer. Under the considerations of the factors affecting delays in transfers, we assume that the delay service time depends on the priority level and state of the delay node, $N_0=(n_0^1, n_0^2, ..., n_0^s)$ where n_0^p, $p=1, ..., s$, is the number of priority p tickets present at the delay node.

6.4 Operators as Servers

In this section, we investigate the server structure of operators. Operators who perform activities to resolve tickets are regarded as servers in our model. Service policy of operators in handling tickets is unknown. In order to better understand how an operator serves the tickets that arrive at him/her we analyze the operator logs, the logs of the events that a particular operator have performed with the help of log visualization and filtering plug-ins in ProM. See the dotted chart of the activities by the operator A (The colors represent the category of the tickets) in figure 6.3.

Our first observation is that for most of the tickets operator A has served the tickets by performing only single activities on the tickets that lead to zero service times, as we have defined the service times by the duration between the first activity and the last activity, whereas for some tickets operator has served the tickets for longer times by performing multiple activities.

In order to distinguish between the tickets which require a single activity and the tickets which require multiple activities from an operator, we determine service requirements of the tickets at operators. The service requirement of a ticket at an operator depends on what is requested in the ticket to resolve the ticket. The question of what is requested in the ticket can only be answered with the knowledge of the problem description of the ticket, the content of the ticket. However, in our research we only make use of the ticket classification information such as the type, the category, and the priority level of the tickets and we determine the routes of the tickets among the operators by their categories. In the absence of the problem description, we make use of the category information to determine the service requirements of the tickets at the operators.

Operators at the ICT department serve the tickets based on the priority levels of

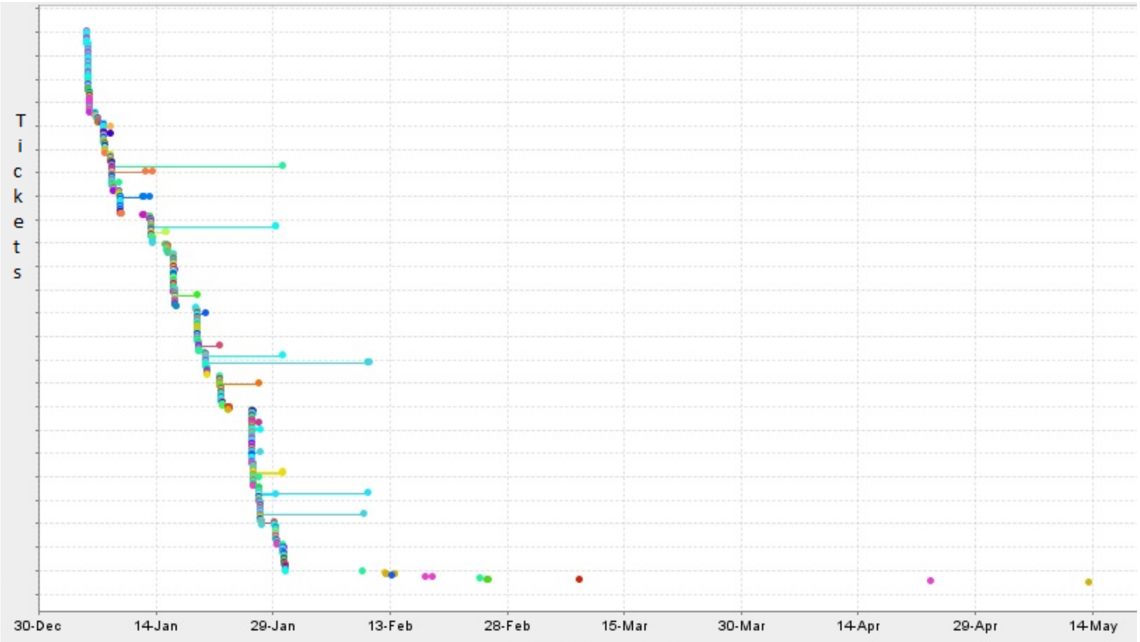


Figure 6.3: Dotted Chart of Activities by Operator A

the tickets. A ticket with a high priority needs to be resolved quickly according to the department's resolution time targets. Among the tickets that arrived at an operator, operator gives priority to the tickets which have high priorities in order to meet the target resolution times. That's why we represent the state of the operators by the number of tickets per priority that are at the operators.

We try to understand the policy of the operators in serving the tickets. We find that operators serve multiple tickets at a time. So, if an operator has a ticket which is not completed its service time at that operator, operator does not wait for the completion of the ticket to start to serve another ticket that arrived at him/her. We assume that an operator starts to serve all the tickets that arrive at him/her.

Based on our investigations on the server structure of the operators, we model the operators as infinite server queues and we assume that the service times of a ticket at an operator depends on the priority level of the ticket and the service requirement of the ticket at the operator.

6.5 Model Parameters

In this section, we describe how we derive the model parameters based on our event log of historical ticket resolution activities. These parameters are the arrival rates of the tickets at operators, the probabilities of the operators to resolve the tickets, the routing probabilities between the operators, the service requirements of the tickets at the operators, the resolution service times and the delay service times. In our research, we utilize process mining tools in ProM in obtaining model input parameters.

6.5.1 Arrival Rates

We define the set of categories that externally arrive at an operator O_i, $C_{i,0}$, as the ticket categories which O_i ever performed the first activity on a ticket which belongs to that categories.

$$If\ \exists\ t \in T\ s.t.\ t\ belongs\ to\ category\ k\ and\ O_i\ performed\ the\ first\ activity\ then\ k \in C_{i,0} \tag{6.1}$$

For $k \in C_{i,0}$, there is an external arrival of category k with rate $\lambda_{0_{i,k}}$ at operator O_i such that $\lambda_{0_{i,k}} > 0$.

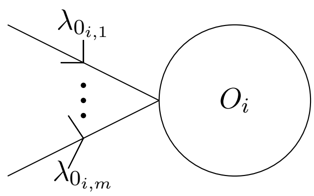


Figure 6.4: External Arrivals at Operators

We calculate $\lambda_{0_{i,k}}$ for each category $k \in C_{i,k}$ and for each operator O_i based on the observed arrivals in our log which contains the ticket resolution activities of the tickets that are created in January 2015 with the help of process mining tools.

Apart from the external arrivals, the tickets arrive at an operator O_i from the delay node. These tickets are previously processed by other operators and could not resolved, and transferred to O_i. Let $\lambda_{i,k}$ denote the arrival rate of category k tickets at operator O_i and let $\lambda_{i,D,k}$ denote the arrival rate of category k tickets which are transferred from operator O_i at delay, then we have

$$(1 - p_{i,k})\lambda_{i,k} = \lambda_{i,D,k}. \tag{6.2}$$

Let $\lambda_{D,i,k}$ be the arrival rate of category k tickets which are transferred to operator O_i from delay, then $\lambda_{D,i,k}$ is

$$\lambda_{D,i,k} = \sum_{j=1}^{j=l} p_{j,0,i,k}\lambda_{j,D,k}. \tag{6.3}$$

Total arrival rate of category k tickets at an operator O_i is the sum of the external arrival rate of category k at operator O_i and the category k arrivals that come from delay to operator O_i, thus $\lambda_{i,k}$ is

$$\lambda_{i,k} = \lambda_{0_{i,k}} + \lambda_{D,i,k}. \tag{6.4}$$

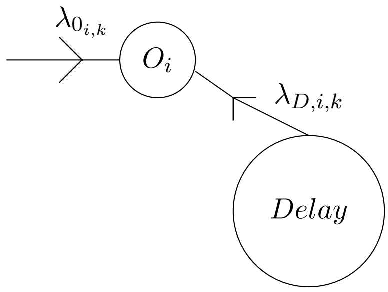


Figure 6.5: Arrivals at Operators

Note that we define the state of an operator O_i as $N_i = (n_1^i, \ldots, n_s^i)$. Let $\lambda_{i,p}$ be the arrival rate of the tickets with priority p at an operator O_i, then $\lambda_{i,p}$ is

$$\lambda_{i,p} = \sum_{k=1}^{k=m} \lambda_{i,k}\tilde{p_p} \tag{6.5}$$

where $\tilde{p_p}$ is the probability that a ticket has a priority level p.

Note that we define the state of the delay node as $N_0 = (n_1^0, n_2^0, \ldots, n_s^0)$. Let $\lambda_{i,p,D,k}$ be the arrival rate of k category and p priority tickets that are transferred from the operator O_i at delay, then $\lambda_{i,p,D,k}$ is

$$\lambda_{i,p,D,k} = \lambda_{i,D,k}\tilde{p_p}. \tag{6.6}$$

Let $\lambda_{p,D}$ be the arrival rate of p priority tickets at delay, then $\lambda_{p,D}$ is

$$\lambda_{p,D} = \sum_{i=1}^{i=l} \sum_{k=1}^{k=m} \lambda_{i,p,D,k}. \tag{6.7}$$

6.5.2 Resolution Probabilities

When a ticket is being processed by an operator, operator may resolve the ticket by the end of the resolution service duration or could not resolve the ticket and the operator may route the ticket to another operator. In our model which we have ticket categories as classes, we define the probability of an operator to resolve a class.

Let $p_{i,k}$ be the probability that operator O_i can resolve category k tickets and let $p_{i,k,0}$ be the probability that operator O_i could not resolve category k tickets and transfers to other operators via delay node. We calculate this probability from the observed instances in our log. We define $p_{i,k}$ as the following:

$$p_{i,k} = \frac{number\ of\ tickets\ of\ category\ k\ that\ is\ resolved\ by\ O_i}{number\ of\ tickets\ of\ category\ k\ that\ is\ processed\ by\ O_i}. \tag{6.8}$$

For each ticket of category k that arrives at operator O_i, operator either resolves the ticket or transfers the ticket to be processed by other operators via delay node.

$$p_{i,k} + p_{i,k,0} = 1 \tag{6.9}$$

6.5.3 Routing Probabilities Between Operators

Routing of tickets between operators occurs via delay. When an operator O_i can not resolve a ticket which belongs to category k, the ticket is transferred to another operator and with probability $p_{i,0,j,k}$ the ticket is transferred to operator O_j.

$$\sum_{j=1}^{j=m} p_{i,0,j,k} = 1 \tag{6.10}$$

We calculate $p_{i,0,j,k}$

$$p_{i,0,j,k} = \frac{number\ of\ category\ k\ tickets\ that\ are\ transferred\ from\ O_i\ to\ O_j}{number\ of\ category\ k\ tickets\ that\ are\ transferred\ from\ O_i}. \tag{6.11}$$

6.5.4 Delay Service Times

In our research, we assume that the delay is an infinite server queue with service times D_{p,N_0} that are exponential with parameters which depend on the priority level and the state of the delay, N_0. We derive these parameters from the delay activity durations that are observed in our log.

In our research, we incorporate the state of the delay in determining the delay service rates by clustering the state of the delay into two states: heavy load and normal load. In order to distinguish the delay service time when the delay is under heavy load and normal load, we derive two different parameters, one for when delay is under heavy load and one for when delay is under normal load, for the tickets per priority.

Assuming that the service time of the delay with state $N_0 = (n_1^0, n_2^0, \ldots, n_s^0)$ depends on n_p^0, $p = 1, \ldots, s$, we consider a function of the state as the following:

$$f(N_0) = \sum_{p=1}^{p=s} n_p^0. \tag{6.12}$$

We use the average of $f(N_0)$ of the delay states N_0 of the delay events in our log to cluster the state of the delay into two. Let denote the average of $f(N_0)$ of the delay states of the delay events in our log by $\overline{f_L(N_0)}$. Then, we consider the delay events which enter the delay with state N_0 which has $f(N_0) > \overline{f_L(N_0)}$ as the delay events which enter a heavy load delay and the delay events which enter the delay with state N_0 which has $f(N_0) \leq \overline{f_L(N_0)}$ as the delay events which enter a normal load delay.

Let $\mu^D_{p,heavy}$ and $\mu^D_{p,normal}$ be the service rates at delay of the tickets with priority p and when delay is under heavy load and when delay is under normal load respectively.

We derive the delay service time parameter for the tickets with priority p under a heavy load delay, $\mu^D_{p,heavy}$, and the delay service time parameter for the tickets with priority p under a normal load delay, $\mu^D_{p,normal}$, from the average delay durations of the delay events of the tickets with priority p which enter a heavy load delay and from the average delay durations of the delay events of the tickets with priority p which enter a normal load delay, respectively, in the log.

For a ticket with priority p and is being served at delay with state N_0, the delay service time of the ticket, D_{p,N_0} is exponential with a parameter which depends on N_0.

$$D_{p,N_0} = \begin{cases} Exp(\mu^D_{p,heavy}), & if \quad f(N_0) > \overline{f_L(N_0)} \\ Exp(\mu^D_{p,normal}), & if \quad f(N_0) \leq \overline{f_L(N_0)} \end{cases} \tag{6.13}$$

$$\mathbb{E}(D_{p,N_0}) = \begin{cases} \frac{1}{\mu^D_{p,heavy}}, & if \quad f(N_0) > \overline{f_L(N_0)} \\ \frac{1}{\mu^D_{p,normal}}, & if \quad f(N_0) \leq \overline{f_L(N_0)} \end{cases} \tag{6.14}$$

6.5.5 Resolution Service Times

In our research, we assume that the service time of a ticket with priority p and category k at an operator O_i given the state of the operator N_i, R_{i,p,k,N_i}, is exponential with a parameter which depends on the category, the operator and the state of the operator. We use the mean service times at operators of the instances that are in our log in order to derive the parameter of R_{i,p,k,N_i}.

In our research, we assume that the service requirement of the tickets with category k at an operator O_i is a random variable, $S_{i,k}$, which has exponential distribution with a parameter that depends on the ticket category k and the operator O_i. In our research, we distinguish the service requirement of the tickets of categories which have no requirements and the service requirements of the tickets of categories which have positive service requirements at an operator O_i. Let C_i^0 be the ticket categories which have zero mean service duration of the resolution service events by an operator O_i and C_i^+ be the ticket categories which have positive mean service duration of the resolution service events by an operator O_i in the log.

The service by operator O_i on the tickets of categories $k \in C_i^0$ only involves a single activity, a text reply for the resolution of the tickets. Recall that we derive the service times by an operator on a ticket by the time between the last activity and the first activity by the operator on the ticket. So, in the case that the operator only performs a single activity on the ticket, we do not observe any service duration. However, in reality the activity of writing a text reply and sending it takes time. Based on the discussions with the operators at ICT department about the time required to write a reply, in our research we assume that the service requirement of the ticket categories $k \in C_i^0$ at operator O_i is one minute.

For the ticket categories $k \in C_i^+$, we define the service requirement of a ticket of category k at operator O_i as an exponential with mean $\mu^S_{i,+}$ as the mean service duration of the resolution events of tickets of category k at operator O_i. We define service requirements of the ticket categories at the operators as the following:

$$S_{i,k} = \begin{cases} 1\ min, & if\ k \in C_i^0 \\ Exp(\mu^S_{i,+}), & if\ k \in C_i^+ \end{cases} . \tag{6.15}$$

We derive load dependent resolution service times by considering the fact that operators discriminate tickets with high priorities over tickets with low priorities and interpreting this fact in determining the load of an operator as high priority tickets impose heavier loads on operators than low priority tickets. In our research, we interpret the fact that operators discriminate high priority tickets into a load dependent service by operators by assuming that high priority tickets create load on operators twice of the load by low priority tickets.

The tickets are prioritized by giving them priority levels between 1 and 5. The tickets having priority levels between 1 and 4 are considered as high priority tickets and the tickets with priority level 5 are considered as low priority tickets by operators at ICT department. Thus, we assume that tickets of priorities 1, 2, 3 and 4 create twice of the load that is created by priority 5 tickets.

Let the state of an operator O_i be $N_i=(n_1^i, \ldots, n_s^i)$ and consider a function of the state $f(N_i)$ of an operator O_i which is

$$f(N_i) = \sum_{p=1}^{p=4} 2n_p^i + n_5^i. \tag{6.16}$$

Similarly to our approach in determining parameters for the delay services, we determine two resolution service time parameters: $\mu^R_{i,+,heavy}$ and $\mu^R_{i,+,normal}$ for the operator O_i when he is under heavy load and when he is under normal load, respectively. We determine the load of an operator O_i as heavy or normal based on the average of $f(N_i)$ values of the resolution events by O_i in our log. Let $\overline{f_L(N_i)}$ be the average of the $f(N_i)$ values of the resolution events by O_i in our log L. Then, in order to derive the parameters $\mu^R_{i,+,heavy}$ and $\mu^R_{i,+,normal}$, we use the average service times of the resolution events of the tickets with categories $k \in C_i^+$ at operator O_i with state N_i such that $f(N_i) > \overline{f_L(N_i)}$ and the average service times of the resolution events of the tickets with categories $k \in C_i^+$ at operator O_i with state N_i such that $f(N_i) \leq \overline{f_L(N_i)}$, respectively.

We assume that an operator O_i is heavy when the state of O_i is N_i and $f(N_i) > \overline{f_L(N_i)}$ and serves the tickets with categories $k \in C_i^+$ with rate $\mu^R_{i,+,heavy}$ and, he is under normal load when the state of O_i is N_i and $f(N_i) \leq \overline{f_L(N_i)}$ and serves the tickets with categories $k \in C_i^+$ with rate $\mu^R_{i,+,normal}$.

$$R_{i,p,k,N_i} = \begin{cases} 1\ min, & if \quad k \in C_i^0 \\ Exp(\mu^R_{i,+,heavy}), & if \quad f(N_i) > \overline{f_L(N_i)}\ and\ k \in C_i^+ \\ Exp(\mu^R_{i,+,normal}), & if \quad f(N_i) \leq \overline{f_L(N_i)}\ and\ k \in C_i^+ \end{cases} \tag{6.17}$$

$$\mathbb{E}(R_{i,p,k,N_i}) = \begin{cases} 1\ min, & if \quad k \in C_i^0 \\ \frac{1}{\mu^R_{i,+,heavy}}, & if \quad f(N_i) > \overline{f_L(N_i)}\ and\ k \in C_i^+ \\ \frac{1}{\mu^R_{i,+,normal}}, & if \quad f(N_i) \leq \overline{f_L(N_i)}\ and\ k \in C_i^+ \end{cases} \tag{6.18}$$

7 Model Analysis

In this chapter, we analyze the queueing network model that we present in chapter 6. Each component of the queueing network, operators and the delay node has infinite capacity and serves the tickets with state dependent rates. Here, we decompose the queueing network model into its components and provide queueing analysis per component.

Our queueing network which consists of components which have infinite capacity and serve the customer with state dependent rates of exponential service times is a product form queueing network by BCMP theorem [2]. Thus, each component of our queueing network can be analyzed in isolation and a joint steady state distribution of the network can be found from the steady state distributions of the components.

Let n be the state of the system and n_i be the state of the operators for $i = 1, \dots, l$ and n_0 be the state of the delay node such that $n = (n_1, \dots, n_l, n_0)$, then the equilibrium distribution of the network is

$$\pi(n) = \frac{1}{G} V(n) \prod_{i=0}^{i=l} \pi_i(n_i) \tag{7.1}$$

where $\pi(n)$ is the steady state distribution of the queueing network, $\pi_i(i)$ is the steady state distribution of the ith component in isolation, $V(n)$ is a function of the state and G is the normalizing constant.

7.1 Analysis of Operators

We describe each operator as a server which has infinite capacity and serves the tickets with rates which depend on the ticket category and the state of the server $N_i = (n_1^i, \dots, n_s^i)$. We describe the traffic rates that enter the operators in section 6.5.1. We assume that service times and inter-arrival times of the tickets are exponential.

We define the state of an operator O_i as heavy and normal depending on $\overline{f_L(N_i)}$, a value that we obtain from the log.

$$N_i^{state} = \begin{cases} heavy, & if \quad f(N_i) > \overline{f_L(N_i)} \\ normal, & if \quad f(N_i) \leq \overline{f_L(N_i)} \end{cases} \tag{7.2}$$

The service rate of the operator O_i becomes $\mu^R_{i,+,heavy}$ when the state of the operator is heavy and $\mu^R_{i,+,normal}$ when the state of the operator is normal for the tickets with categories $k \in C_i^+$. For the tickets with categories $k \in C_i^0$, we acknowledge a service duration of 1 minute and we do not include them in defining the state of the operators, for them we acknowledge that mean sojourn time of at O_i is 1 minute independent of the state.

In order to perform queueing analysis of the operators, first we need to obtain the equilibrium probabilities. Let p_n^i be the probability that there are n tickets at operator O_i. In order to obtain the limiting probabilities of the operators, we represent the state of the operators as a continuous time Markov chain. Let the state of the chain of an operator O_i be the number of tickets present at operator O_i and denote it by n_i, then the state of the Markov chain is defined by

$$n_i = \sum_{p=1}^{p=s} n_p^i, \quad \textit{with state space } \{0, 1, \ldots\}. \tag{7.3}$$

This Markov chain is a birth and death process and let's denote the birth rate by λ_i and the death rate by μ_{i,n_i}. The birth rate λ_i for an operator O_i is the rate that the number of tickets of categories of C_i^+ at operator O_i increases by one and it can be calculated from the traffic of ticket categories of C_i^+ that visits operator O_i as the following:

$$\lambda_i = \sum_{k \in C_i^+} \lambda_{i,k}. \tag{7.4}$$

The death rate μ_{i,n_i} is the service rate of operator O_i when there are n_i tickets at operator O_i. Note that we describe load dependent service rates in which a ticket is served with different rates when the operator is under heavy load and when the operator is under normal load, thus the death rate of the Markov chain is defined by

$$\mu_{i,n_i} = \begin{cases} n_i \mu^R_{i,+,heavy}, & if\ N_i^{state} = heavy \\ n_i \mu^R_{i,+,normal}, & if\ N_i^{state} = normal. \end{cases} \tag{7.5}$$

In chapter 6, we describe a state function to determine when an operator is under heavy load or normal load. This function is not same as the state of our chain, n_i, so, we should derive when an operator is under a heavy or normal load given the number of tickets present n_i at operator O_i in order to determine the service rates.

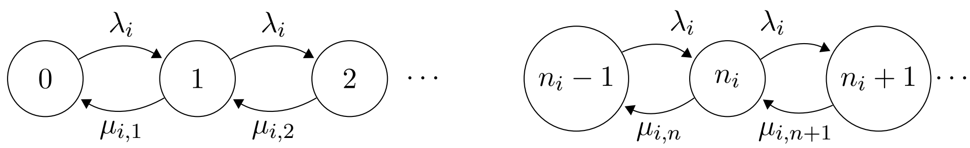


Figure 7.1: Markov Chain of Operators

In chapter 6, we define the probability of a ticket to have a priority level p as $\tilde{p_p}$. Then, given there are n_i tickets at operator O_i, $\tilde{n_p^i}$=$n_i\tilde{p_p}$ of them have priority level p. Then, we calculate the $f(N_i)$ of operator O_i given the number of tickets at the operator, $f(n_i)$ as the following,

$$f(n_i) = \sum_{p=1}^{p=4} 2\tilde{n_p^i} + \tilde{n_5^i}. \tag{7.6}$$

Let $\tilde{k_i}$ be the minimum number of tickets at operator O_i that puts O_i under heavy load. Namely $\tilde{k_i}$ is

$$\tilde{k_i} = \min_{n_i} \ s.t \ f(n_i) > \overline{f_L(N_i)}. \tag{7.7}$$

Then, we calculate μ_{i,n_i}

$$\mu_{i,n_i} = \begin{cases} n_i \mu^R_{i,+,heavy}, & if \ n_i \geq \tilde{k_i} \\ n_i \mu^R_{i,+,normal}, & otherwise. \end{cases} \tag{7.8}$$

In order to obtain the equilibrium probabilities we solve the cut equations of the chain.

$$\begin{aligned}
p_0^i \lambda_i &= p_1^i \mu^R_{i,+,normal} \\
p_1^i \lambda_i &= 2 p_2^i \mu^R_{i,+,normal} \\
\cdots & \quad \cdots \\
p^i_{\tilde{k_i}-2} \lambda_i &= (\tilde{k_i} - 1) p^i_{\tilde{k_i}-1} \mu^R_{i,+,normal} \\
p^i_{\tilde{k_i}-1} \lambda_i &= (\tilde{k_i}) p^i_{\tilde{k_i}} \mu^R_{i,+,heavy} \\
p^i_{\tilde{k_i}} \lambda_i &= (\tilde{k_i} + 1) p^i_{\tilde{k_i}+1} \mu^R_{i,+,heavy} \\
\cdots & \quad \cdots
\end{aligned} \tag{7.9}$$

Then, we obtain the stationary probability that there are n tickets at operator O_i as the following:

$$p_n^i = \begin{cases} p_0^i \frac{(\lambda_i)^n}{(\mu^R_{i,+,normal})^n n!}, & n = 1, 2, \ldots, \tilde{k_i} - 1 \\ p_0^i \frac{(\lambda_i)^n}{(\mu^R_{i,+,normal})^{\tilde{k}-1} (\mu^R_{i,+,heavy})^{n-(\tilde{k_i}-1)} n!}, & n = \tilde{k_i}, \ldots, \end{cases} \tag{7.10}$$

Together with normalization equations given as

$$\sum_{n=0}^{n=\infty} p_n^i = 1 \tag{7.11}$$

we calculate p_0^i

$$p_0^i = (\sum_{n=0}^{\tilde{k_i}-1} \frac{(\lambda_i)^n}{n!(\mu^R_{i,+,normal})^n} + \frac{\lambda_i^{\tilde{k_i}-1}}{(\mu^R_{i,+,normal})^{\tilde{k_i}-1}} (\sum_{n=1}^{n=\infty} \frac{(\lambda_i)^n}{(\mu^R_{i,+,heavy})^n (\tilde{k_i} + n - 1)!}))^{-1}. \tag{7.12}$$

From the equilibrium probabilities of an operator O_i, we calculate the mean number of tickets that are at the operator, $E(L_i)$, as the following:

$$E(L_i) = \sum_{n=0}^{n=\infty} n p_n^i = \sum_{n=0}^{n=\tilde{k_i}-1} n p_0^i \frac{(\lambda_i)^n}{(\mu^R_{i,+,normal})^n n!} + \sum_{n=\tilde{k}}^{n=\infty} \frac{n p_0^i (\lambda_i)^n}{n! (\mu_{i,+,normal})^{\tilde{k}-1} (\mu_{i,+,heavy})^{(n-(\tilde{k}-1))}}. \tag{7.13}$$

By using Little's law [12] we obtain the mean sojourn time of the tickets at an operator O_i, $E(S_i)$.

$$E(L_i) = \lambda_i E(S_i) \tag{7.14}$$

From equilibrium probabilities we calculate the probability that an operator O_i works under heavy load. Let π^i_{heavy} be the probability that O_i is under heavy load. Then, π^i_{heavy} is

$$\pi^i_{heavy} = \sum_{n=\tilde{k_i}}^{n=\infty} p^i_n \tag{7.15}$$

as if there are $\tilde{k_i}$ or more tickets at an operator O_i then he/she is heavy. Similarly, we find the probability that an operator O_i is under normal load as follows

$$\pi^i_{normal} = \sum_{n=0}^{n=\tilde{k_i}-1} p^i_n. \tag{7.16}$$

We derive the sojourn time distribution of operator O_i. In an infinite server queue, the sojourn time is the service time of a customer. Let S_i be the sojourn time distribution at operator O_i and B be the service time of a ticket. The sojourn time of an arriving ticket at an operator is its service time at the operator, S_i=B. With probability π^i_{heavy} operator O_i works under heavy load and with probability π^i_{normal} he works under normal load. When he is under heavy load he serves the tickets with rate $\mu^R_{i,+,heavy}$ and serves with the rate of $\mu^R_{i,+,normal}$ when working under normal load. Let B_h and B_n be the service time that is received under heavy load and normal load, respectively, then, $B = B_h + B_n$. Let L^a_i be the number of tickets present at the operator O_i on the arrival of a new ticket at O_i, we condition on L^a_i to calculate sojourn time distribution at operator O_i.

$$\begin{aligned}
&P(S_i > t) = P(B > t) = P(B_h + B_n > t) \\
&= \textstyle\sum_{n=0}^{n=\infty} P(B_h + B_n > t)P(L^a = n) \\
&= \textstyle\sum_{n=0}^{n=\tilde{k_i}-1} P(B_h + B_n > t)P(L^a = n) + \sum_{n=\tilde{k_i}}^{n=\infty} P(B_h + B_n > t)P(L^a = n) \\
&= \textstyle\sum_{n=0}^{n=\tilde{k_i}-1} \big[\int_{x=0}^{\infty} P(B_h > t - x)\mu_{i,+,normal} \exp^{-\mu_{i,+,normal}x} dx\big] P(L^a = n)+ \\
&\textstyle\sum_{n=\tilde{k_i}}^{n=\infty} \big[\int_{x=0}^{\infty} P(B_n > t - x)\mu_{i,+,heavy} \exp^{-\mu_{i,+,heavy}x} dx\big] P(L^a = n) \\
&= \textstyle\sum_{n=0}^{n=\tilde{k_i}-1} \big[\int_{x=0}^{t} \pi^i_{heavy} \exp^{-\mu_{i,+,heavy}(t-x)} \mu_{i,+,normal} \exp^{-\mu_{i,+,normal}x} dx+ \\
&\textstyle\int_{x=t}^{\infty} \mu_{i,+,normal} \exp^{-\mu_{i,+,normal}x} dx\big] P(L^a = n)+ \\
&\textstyle\sum_{n=\tilde{k_i}}^{n=\infty} \big[\int_{x=0}^{t} \pi^i_{normal} \exp^{-\mu_{i,+,normal}(t-x)} \mu_{i,+,heavy} \exp^{-\mu_{i,+,heavy}x} dx+ \\
&\textstyle\int_{x=t}^{\infty} \mu_{i,+,heavy} \exp^{-\mu_{i,+,heavy}x} dx\big] P(L^a = n) \\
&= \textstyle\sum_{n=0}^{n=\tilde{k_i}-1} \big[\pi^i_{heavy} \exp^{-\mu_{i,+,heavy}} \mu_{i,+,normal} \frac{(1-\exp^{-t(\mu_{i,+,normal}-\mu_{i,+,heavy})})}{(\mu_{i,+,normal}-\mu_{i,+,heavy})} \\
&+ \exp^{-\mu_{i,+,normal}t} \big] P(L^a = n) \\
&+ \textstyle\sum_{n=\tilde{k_i}}^{n=\infty} \big[\pi^i_{normal} \exp^{-\mu_{i,+,normal}} \mu_{i,+,heavy} \frac{(1-\exp^{-t(\mu_{i,+,heavy}-\mu_{i,+,normal})})}{(\mu_{i,+,heavy}-\mu_{i,+,normal})} \\
&+ \exp^{-\mu_{i,+,heavy}t} \big] P(L^a = n)
\end{aligned} \tag{7.17}$$

7.2 Analysis of Delay

When a ticket could not be resolved upon its resolution service completion at an operator, the ticket is transferred to another operator and in our model, we permit the transfer event via delay node and the service time at delay represents the random time that is spent in transfer. In our model, delay node has infinite capacity and serves the tickets with exponential service times with rates that depend on priority and the state of the delay $N_0 = (n_1^0, \ldots, n_s^0)$.

We define the state of the delay node as heavy or normal depending on the value of the function $\overline{f_L(N_0)}$.

$$N_0^{state} = \begin{cases} heavy, & if \quad f(N_0) > \overline{f_L(N_0)} \\ normal, & if \quad f(N_0) \leq \overline{f_L(N_0)} \end{cases} \tag{7.18}$$

The service rates of the delay are $\mu_{p,heavy}^D$ and $\mu_{p,normal}^D$ for the tickets with priority level p when the delay node is under heavy load and normal load, respectively.

We represent the state of the delay as a continuous time Markov chain to obtain the equilibrium probabilities. Since the service times are dependent on the priority, we describe a Markov chain per priority tickets at delay. Consider a priority level p, we represent the number of p priority tickets at delay as a continuous time Markov chain. Let n_p^0 be the number of p priority tickets at delay. We describe the arrival rates of p priority tickets at delay, $\lambda_{p,D}$, in equation 6.7. Let μ_{p,n_p^0} be the service rate of delay for p priority tickets when there are n_p^0 priority p tickets at delay.

$$\mu_{p,n_p^0} = \begin{cases} n_p^0 \mu_{p,heavy}^D, & if\ N_0^{state} = heavy \\ n_p^0 \mu_{p,normal}^D, & if\ N_0^{state} = normal \end{cases} \tag{7.19}$$

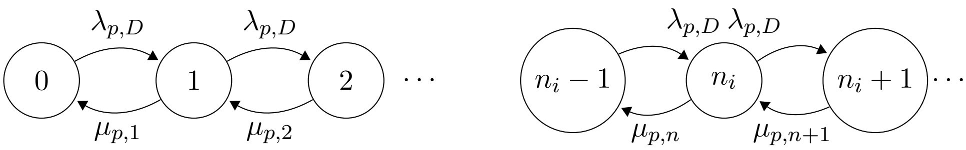


Figure 7.2: Markov Chain of Delay

In order to determine the service rate μ_{p,n_p^0} when the state of the p priority chain is n_p^0, we need to know when the state of the delay gets heavy given the state n_p^0. We defined the probability that a ticket has a priority level p as $\tilde{p_p}$. So, if there are n^0 number of tickets at delay, $n^0 \tilde{p_p}$ of them have priority level p. Then, we define

$$f(n_p^0) = \sum_{p=1}^{p=s} n_p^0 = \frac{n_p^0}{\tilde{p_p}}. \tag{7.20}$$

Let $\tilde{k_p^0}$ be the minimum number of p priority tickets at delay, that corresponds to a heavy load delay node. Namely, $\tilde{k_p^0}$ is

$$\tilde{k_p^0} = \min_{n_p^0} \ s.t \ f(n_p^0) > \overline{f_L(N_0)}. \tag{7.21}$$

Then, we calculate μ_{p,n_p^0}

$$\mu_{p,n_p^0} = \begin{cases} n_p^0 \mu_{p,heavy}^D, & if \ n_p^0 \geq \tilde{k_p^0} \\ n_p^0 \mu_{p,normal}^D, & otherwise \end{cases}. \tag{7.22}$$

Let $p_n^{0,p}$ be the probability that there are n number of p priority tickets at delay in equilibrium. In order to obtain the equilibrium probabilities we solve the cut equations of the chain.

$$\begin{aligned}
p_0^{0,p}\lambda_{p,D} &= p_1^{0,p}\mu_{p,normal}^D \\
p_1^{0,p}\lambda_{p,D} &= 2p_2^{0,p}\mu_{p,normal}^D \\
\cdots \quad & \quad \cdots \\
p_{\tilde{k_p^0}-2}^{0,p}\lambda_{p,D} &= (\tilde{k_p^0}-1)p_{\tilde{k_p^0}-1}^{0,p}\mu_{p,normal}^D \\
p_{\tilde{k_p^0}-1}^{0,p}\lambda_{p,D} &= (\tilde{k_p^0})p_{\tilde{k_p^0}}^{0,p}\mu_{p,heavy}^D \\
p_{\tilde{k_p^0}}^{0,p}\lambda_{p,D} &= (\tilde{k_p^0}+1)p_{\tilde{k_p^0}+1}^{0,p}\mu_{p,heavy}^D \\
\cdots \quad & \quad \cdots
\end{aligned} \tag{7.23}$$

Together with normalization equations

$$\sum_{n=0}^{n=\infty} p_n^{0,p} = 1 \tag{7.24}$$

we calculate $p_0^{0,p}$ as the following:

$$p_0^{0,p} = \left(\sum_{n=0}^{\tilde{k_p^0}-1} \frac{(\lambda_{p,D})^n}{n!(\mu_{p,normal}^D)^n} + \frac{\lambda_{p,D}^{\tilde{k_p^0}-1}}{(\mu_{p,normal}^D)^{\tilde{k_p^0}-1}} \left(\sum_{n=1}^{n=\infty} \frac{(\lambda_{p,D})^n}{(\mu_{p,heavy}^D)^n(\tilde{k_p^0}+n-1)!}\right)\right)^{-1}. \tag{7.25}$$

From the equilibrium probabilities $p_n^{0,p}$, we calculate the mean number of p priority tickets at delay, $E(L_0^p)$.

$$E(L_0^p) = \sum_{n=0}^{n=\infty} np_n^{0,p} = \sum_{n=0}^{n=\tilde{k_p^0}-1} np_0^{0,p}\frac{(\lambda_{p,D})^n}{(\mu_{p,normal}^D)^n n!} + \sum_{n=\tilde{k_p^0}}^{n=\infty} \frac{np_0^{0,p}(\lambda_{p,D})^n}{n!(\mu_{p,normal})^{\tilde{k_p^0}-1}(\mu_{p,heavy})^{(n-(\tilde{k_p^0}-1))}} \tag{7.26}$$

By using Little's law [12] we calculate the mean sojourn time of the p priority tickets at delay, $E(S_0^p)$.

$$E(L_0^p) = \lambda_{p,D}E(S_0^p) \tag{7.27}$$

We find the mean number of tickets at delay as

$$E(L_0) = \sum_{n=1}^{n=s} E(L_0^p). \tag{7.28}$$

We find mean sojourn time of tickets at delay $E(S_0)$ with Little's law [12]

$$E(L_0) = \lambda_D E(S_0) \tag{7.29}$$

where λ_D is the total traffic entering delay node, namely

$$\lambda_D = \sum_{p=1}^{p=s} \lambda_{p,D}. \tag{7.30}$$

7.3 Mean Network Performance

By analyzing each queue of the queueing network in isolation, we obtain the marginal equilibrium distributions and mean queueing performance measures such as mean number of tickets and mean sojourn times at individual queues. Here, we obtain mean queueing performance measures of the network from the mean queueing performance measures of its sub components. The mean number of tickets in the network is the sum of mean number of tickets at the independent components of the network.

$$E(L) = \sum_{i=1}^{i=l} E(L_i) + E(L_0) \tag{7.31}$$

The throughput of the system is the external arrivals of ticket categories to operators.

$$\lambda = \sum_{i=1}^{i=l} \sum_{k=1}^{k=m} \lambda_{0_{i,k}} \tag{7.32}$$

We find the mean sojourn time of a ticket in the system with Little's formula.

$$E(S) = \frac{\lambda}{E(L)} \tag{7.33}$$

7.4 Queueing Analysis Implementation

In this section, we implement performing queueing analysis to the operators and delay queues. We obtain mean queue performance measures for the operators, delay and the network.

7.4.1 Queue Input Parameters

We find two arrival rates of the tickets of categories of C_i^+ and C_i^0 at each operator O_i: λ_i^+ and λ_i^0, respectively. For the tickets that belong to the categories in C_i^0, operators behave as separate infinite server queues which serve each ticket one minute. In our log which contains 255 operators, we find that for 85 of the operators only the tickets of the categories in C_i^0 arrive, λ_i^+=0. We define these operators as type 1 operators (see A.2). For the operators with $\lambda_i^+ > 0$, we find the service rates when the operators are under heavy load and normal load, $\mu^R_{i,+,heavy}$ and $\mu^R_{i,+,normal}$ (see 6.18). We define the operators who only work under the normal loads as type 2 operators (see A.3) and we call the operators who switch between heavy and normal loads as type 3 operators (see A.4).

With our method to determine the loads of the operators (see 6.16), our intuition is that the operators do not serve faster when they switch to heavy loads, $\mu^R_{i,+,heavy} \leq \mu^R_{i,+,normal}$. However, we observe that only 52% of the operators with $\lambda_i^+ > 0$ serve the tickets faster when under normal loads (see A.3 and A.4). Our reasoning suggests that the reason why an operator serves faster under heavy load can be related to the ICT department's policy of letting the operators to take the tickets voluntarily. For instance, an operator who does not have many tickets, more specifically when he/she is under normal load, he/she may not choose to take any other tickets if the loads of the tickets that he/she already has are heavy and an operator who has many tickets, when he/she is under heavy load, might have chosen to take that many tickets because the loads of the tickets he/she had were not heavy. We conclude that our approach of considering the priorities and the number of the tickets that operators handle in determining the load of the operators is not sufficient and the loads of the tickets that are at the operators should be taken into consideration to determine loads of the operators under the policy of voluntarily ticket taking.

The derived queueing input parameters for the delay node are given in A.1. The tickets of priority 5 are served the slowest and the tickets of priority 1 are served the fastest at delay. For the tickets which have priorities of 2, 3 and 4, the service rates at delay are similar and these tickets are served slowly than the tickets of priority 1 and served faster than the tickets of priority 5. However, we do not observe the relation that the tickets of priority 2 are served faster than the tickets of priority 3 and the tickets of priority 3 are served faster than the tickets of priority 4.

As in the relation of the service rate of the operators when the operators are under heavy load and normal load, we expect to have that the tickets are not served faster when the delay is under heavy load. From the derived service rates of the delay from our event log, we observe that the relation of $\mu^D_{p,heavy} \leq \mu^D_{p,normal}$ holds for the tickets of priorities 1, 2, 3 and 5.

7.4.2 Queue Analysis of Operators, Delay and Network

We perform queueing analysis of each operator queue in isolation. We find the mean queueing performances of each operator for the tickets of the categories in C_i^0 and C_i^+.

For the tickets that belong to the categories in C_i^0, the service of the tickets takes one minute at the operator O_i. The mean number of tickets which belong to the categories in C_i^0 at operators, $E(L_i^0)$, is considerably small, see below the basic statistics.

Min $E\left(L_i^0\right)$	Max $E\left(L_i^0\right)$	$\overline{E\left(L_i^0\right)}$
0	0.013	0.0003

Table 7.1: Basic Statistics of Queue Lengths of C_i^0 Category Tickets

The mean number of tickets of categories C_i^+ at operators differs a lot per operator. We conclude that the operators at the ICT department do not work under similar loads. Recall that λ_i^+=0 for type 1 operators. See below the basic statistics of the mean number of tickets of categories C_i^+ at all types of operators, and type 2 and type 3 operators.

Operators	Min $E\left(L_i^+\right)$	Max $E\left(L_i^+\right)$	$\overline{E\left(L_i^+\right)}$
All Types	0	21.1127	0.9191
Type 2 + Type 3	0	21.1127	1.373

Table 7.2: Basic Statistics of Queue Lengths of C_i^+ Category Tickets

The mean number of tickets at each operator is given in A.2, A.3 and A.4. The important queueing performance measure here is the mean sojourn time of tickets of categories in C_i^+ at operators. We use this measure in our approach to produce resolution time predictions. From our analysis, we find that the mean sojourn times of tickets of categories in C_i^+ at operators differ a lot, see the statistics for type 2 and type 3 operators below.

Type 2 + Type 3	Min $E\left(S_i^+\right)$	Max $E\left(S_i^+\right)$	$\overline{E\left(S_i^+\right)}$	Median $E\left(S_i^+\right)$
hours	0.0024	5257.7894	112.2462	14.4719

Table 7.3: Basic Statistics of Mean Sojourn Times of C_i^+ Category Tickets

For the type 3 operators to whom the tickets of the categories in C_i^+ arrive and who switch to heavy load service rates, we find π^i_{normal} and π^i_{heavy} probabilities (see A.4). We find the probabilities that the sojourn time of tickets of categories in C_i^+ exceeds 1 hour for type 2 and type 3 operators (see A.3 and A.4). We find that the tickets of categories in C_i^+ are 90% likely to have a sojourn time of more than one hour at the 65% of the operators of type 2 and type 3 (see A.3 and A.4).

We find the mean number of tickets and mean sojourn times of tickets at delay per priority (see A.1). We observe that the mean number of tickets of priority 5 at delay is considerably bigger than any other priority tickets that are at delay. The mean sojourn time of the tickets of priority 1 is the smallest and the mean sojourn time of the tickets of priority 5 is the biggest. The mean sojourn time of the tickets of priorities of 2, 3 and 4 are bigger than the mean sojourn time of the tickets of priority 1 and smaller than the mean sojourn time of the tickets of priority 5, however we do not observe the relation of $E(S_0^2) \leq E(S_0^3) \leq E(S_0^4)$.

Lastly, we perform queueing analysis for the network. We find the mean number of tickets and the mean sojourn time in the network. We find that on average a ticket which arrives at the ticket resolution system at the ICT department is resolved in 4-4.5 days.

Throughput (hour)	$E(L)$	$E(S)$
7.8401	817.8288	104.3136

Table 7.4: Mean Performance Analysis of Network

8 Resolution Time Prediction

In this chapter, we propose our methodology to predict resolution times of the tickets given their category and priority information and the results obtained from implementing the prediction approach for two chosen ticket categories. We explain our approach to produce predictions and we present stochastic Petri net representation as a technique to produce resolution time predictions.

8.1 Prediction Approach

In our research, we aim to produce predictions for the resolution times of the tickets by using the ticket priority and category information. The other aspect of our research in producing resolution time predictions for the tickets is queueing theory perspective that we take to produce the predictors. In our research, we take a queueing theory perspective in predicting the resolution times of the tickets by making use of the mean queueing performance measures of the operators and delay. Other than ticket priority and category information, we use the mean sojourn times of the tickets at operators and delay as an input for our method to produce predictions for the resolution times of the tickets.

Definition 8.1.1. A stochastic Petri net is a six tuple, SPN=(P, T, F, M_0, Λ) where

- P is the set of places,
- T is the set of transitions,
- $F \subseteq (P \times T) \cup (T \times P)$ is the set of arcs,
- M_0 is an initial marking,
- Λ is the set of firing rates associated with transitions.

8.2 SPN Representation

We represent per ticket category and ticket priority a timed stochastic Petri net of the queueing network model of the ticket resolution process. Suppose that we produce a SPN for tickets with category k and priority p, then this SPN is a net among operators who are associated to category k, O_k. For each operator $i \in O_k$, we define two places in SPN: one for the state that the operator serves a ticket and one for the state that the

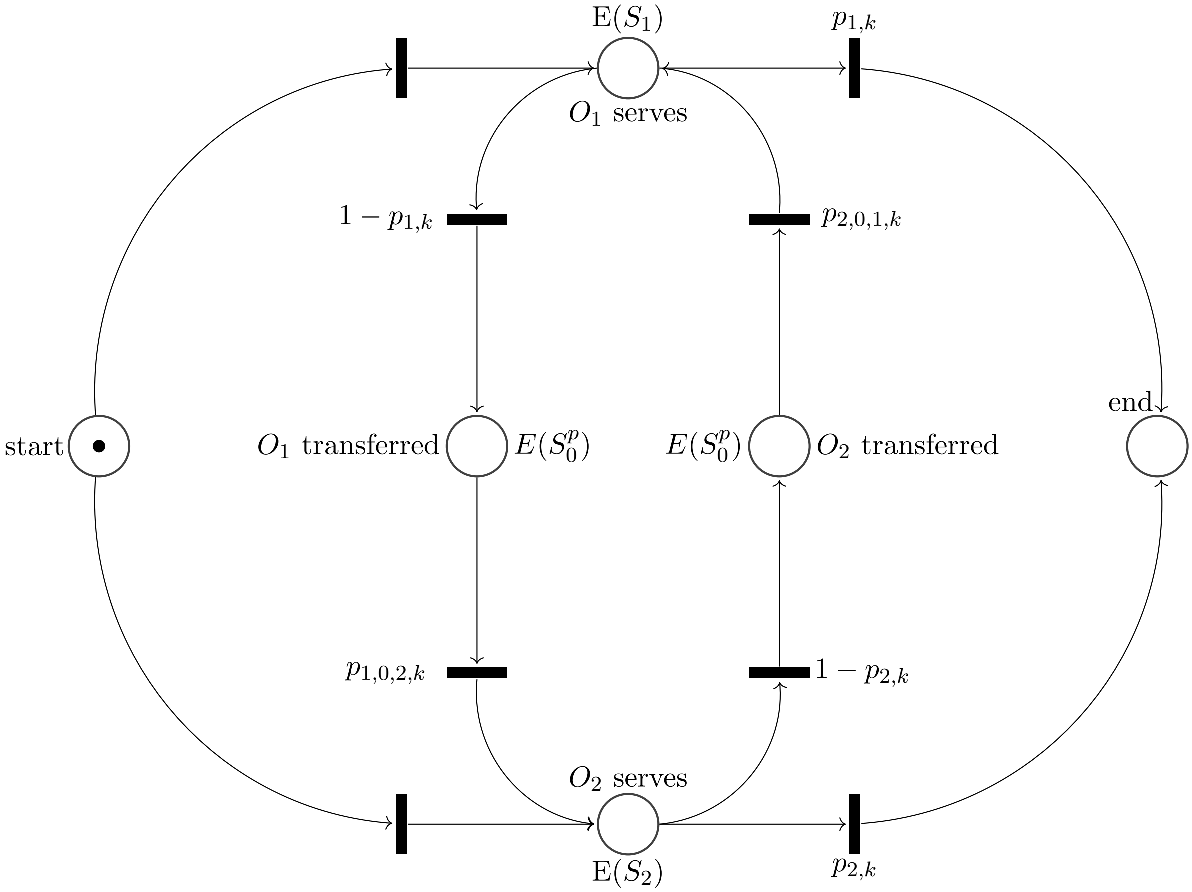


Figure 8.1: SPN Representation

operator can not resolve a ticket and transfers the ticket. The state that an operator i serves a ticket is a timed activity with deterministic duration of mean sojourn time at operator i. From the state that an operator i serves a ticket, there is an immediate transition to the state that operator i transfers the ticket and an immediate transition to the exit with transition rates $1-p_{i,k}$ and $p_{i,k}$, respectively. The state that an operator i transfers a ticket is a timed activity with deterministic duration of mean sojourn delay time for tickets with priority p. From the state that an operator i transfers a ticket, there are immediate transitions to the states that other associated operators $j \in O_k$ serve. From the state that an operator i transfers a ticket, the rate of the transition to the state that an operator $j \in O_k$ serves is $p_{i,0,j,k}$. Arrivals from start to operators occur via immediate transitions with normalized rates of external arrivals $\lambda_{0_{i,k}}$ of category k tickets at operators $i \in O_k$.

An illustration of the SPN for a ticket category k and priority p where there are two operators, O_1 and O_2, associated to ticket category k is given is figure 8.1.

8.3 Prediction Algorithm

Our prediction algorithm predicts the resolution time of an arriving ticket given its category k and priority p by simulating the stochastic Petri net of category k and priority p. Our approach to make use of stochastic Petri nets in predicting ticket resolution times are similar to the approach in [20] in predicting remaining execution time with stochastic Petri nets.

Note that the initial marking of an arriving ticket is a single token which is placed in the start place. In our approach to produce predictions for the resolution times of the tickets, it is also possible to give predictions for the running tickets, the tickets that are processed in the ticket resolution system by operators but have not exited the system yet. In order to make a prediction for a running ticket t with past, $O_1^t, O_2^t, \ldots, O_r^t$ where O_i^t for $i = 1, \ldots, r$ is the ith operator who processed the ticket t and r is the number of operators who processed t, we take the initial marking in the SPN as a single token which is placed in the place that O_r^t serves the ticket.

With an initial marking placed in the SPN, we make resolution time predictions for arriving and running tickets by simulating the SPN for a given number of iterations. Each iteration simulates a possible routing among the operators who are associated to the category of the ticket that we predict, and in each iteration, we collect the information of time to exit. We propose the mean of the collected resolution times that is found in iterations as a predictor for the resolution time of a ticket given its category, priority and initial marking.

Result: Resolution time prediction for a ticket with category k, priority p and initial marking m
Mark m in the SPN model of category k and priority p;
ExitTime ← new List;
while *NumberOfIterationsNotReached* **do**
| add.ExitTime(Exit Time of Run);
end
return mean.ExitTime;

Algorithm 1: Prediction Algorithm

8.4 Prediction Implementation

In this section, we implement producing resolution time predictions for the chosen ticket categories 'EZIS' and 'Radiologie (CS)'. Later, we assess the quality of the predictions on a set of tickets which are created after January in 2015. We implement prediction algorithm in C++ platform.

In our research, we investigate the percentages of the tickets that are resolved within targets. ICT department has target resolution times at line 1 and line 2 per ticket priority. In our research, we do not use the escalation information of the tickets. Therefore, we define upper target resolution times per priority as the sum of the target resolution times at line 1 and line 2 per priority and we investigate the percentages of tickets

Priority	SPN Prediction (hours)
1	13.26
2	29.06
3	51.4
4	28.32
5	71.82

Table 8.1: SPN Predictions for ‘EZIS’ Category

Priority	Upper Targets (hours)	Per. Resolved Within Upper Targets
1	2.25	48%
2	24.5	58%
3	73	68%
4	192	99%
5	336	98%

Table 8.2: Percentages of ‘EZIS’ Tickets Meeting Upper Targets

that are resolved within upper target resolution times with produced resolution time predictions per priority.

In our resolution time prediction approach (see Algorithm 1), we give the averages of the resolution times of the generated runs of the SPN representations as predictions for the ticket resolution times. In our research, we choose the number of runs to perform as 5000 to obtain resolution time predictions which are accurate within 5 units 95% of the time for the ticket category ‘EZIS’ by following the methodology in [37] to find the required number of trials in simulations. The obtained resolution time predictions with the SPN representations method for the tickets that of category ‘EZIS’ are given in table 8.1.

We investigate the percentages of the tickets that are resolved within the upper target resolution times by ticket priorities for the tickets that belong to ‘EZIS’ category. The obtained results of the percentages of the tickets that are resolved within the upper target resolution times by ticket priorities from the simulations of the SPN representation of ‘EZIS’ category are given in table 8.2. We observe that the target resolution times that are set by the ICT department for the tickets of high priorities, 1, 2 and 3, are not met in significantly many cases. We conclude that the short target resolution times are not attainable by the ticket resolution process at ICT department for the tickets of ‘EZIS’ category.

We investigate the minimum resolution times (in hours) per ticket priority that the tickets of category ‘EZIS’ are resolved within. SPN method which presents the average of the collected resolution times of the runs gives predictions for the resolution times which approximately 50% of the tickets are resolved within. We look for the minimum resolution times that the 50%, 70%, 90% and 100% of the tickets are resolved within. The results obtained by the simulations of SPN representation of ‘EZIS’ category per

Priority	50%	70%	90%	100%
1	9	15	32	275
2	16	38	84	333
3	34	76	149	614
4	13	36	77	421
5	51	109	191	1076

Table 8.3: Minimum Resolution Times (hours) That Tickets of 'EZIS' Category Are Resolved Within

ticket priority are given in table 8.3. We observe that minimum resolution times that the 100% of the tickets are resolved within are far away from the target times for all ticket priorities. We observe that the difference between the minimum resolution times that the 90% of the tickets are resolved within and the minimum resolution times that the 100% of the tickets are resolved within is significantly large for all ticket priorities. For a small proportion of the tickets we observe much longer resolution times than the rest of the tickets. Involvement of large number of operators in the resolution of 'EZIS' category can lead to situations that the tickets are bounced among many operators and therefore have long resolution times.

Along with the resolution time predictions that are found with the method described in Algorithm 1 which are based on the average performances of the ticket categories, ICT department may consider the minimum resolution times that the large portion of the tickets are resolved within as predictions for the resolution times. We present the minimum resolution times that the 90% of the tickets of 'EZIS' category are resolved within as the upper resolution predictions.

To compare the quality of the resolution time predictions with SPN representation method, we implement an another prediction method for the tickets that belong to 'EZIS' category. Similarly to the approach of using recent delay history to predict the delay in [7], we use the recent history of resolution times to predict the resolution times of new tickets. In our approach, we take the resolution time of the ticket of 'EZIS' category and of p priority that is most recently resolved in the system as the resolution time prediction for a ticket of 'EZIS' category and of p priority that arrives at the system. We assess the quality of the predictions which use the SPN representation and the recent history of the resolution times with mean absolute error (MAE), mean squared error (MSE) and root mean squared error (RMSE) accuracy measures. Additionally, we provide the prediction errors of the target resolution times by using ticket escalation information. The accuracies of the prediction methods for 'EZIS' category tickets are given in table 8.4.

In terms of MSE and RMSE accuracy measures, the resolution time predictions which are based on the average performances of the SPN representations give more accurate results, while in terms of MAE, the resolution time predictions which are based on the recent resolution history perform better. Our prediction methods which use ticket category and priority information predict the resolution times of the 'EZIS' tickets better

Prediction	MAE	MSE	RMSE
SPN Representation Mean	112.01	109632.06	331.11
SPN Representation Upper	203.34	123343.40	351.20
Recent History	95.66	117661.43	343.02
Target Resolution Times	229.58	131189.7	362.20

Table 8.4: Prediction Errors for 'EZIS' Tickets

Priority	SPN Prediction (hours)
1	65.07
2	107.37
3	162.61
4	104.69
5	213.67

Table 8.5: SPN Predictions for 'Radiologie (CS)' Category

than the target resolution times which are based on the ticket escalation information and ticket priority in all three prediction accuracy measures.

We also implement obtaining resolution time predictions for the tickets that belong to 'Radiologie (CS)' category. In order to obtain predictions which are accurate within 5 units 95% of the time, we choose the number of runs to perform as 10000 for the ticket category 'Radiologie (CS)' by following the methodology given in [37]. The predictions from the averages of the resolution times of the runs of the SPN representation of 'Radiologie (CS)' category per ticket priority are given in table 8.5. The predicted resolution times for the tickets which belong to 'Radiologie (CS)' category are much longer than the predicted resolution times for the tickets which belong to 'EZIS' categories for all ticket priorities. We conclude that the performance of the ticket resolution process in terms of resolution times differs per ticket category.

We also implement recent history prediction for the tickets of 'Radiologie (CS)'. The errors of the predictions are given with MAE, MSE and RMSE accuracy measures in table 8.6. We give the prediction errors of the target resolution times by using ticket escalation and priority information. The target resolution times predict the resolution times of 'Radiologie (CS)' tickets more accurately than our prediction methods which use ticket category and priority as input in terms of MAE measure. However, in terms of MSE and RMSE accuracy measures, SPN representation prediction gives more accurate results.

We investigate the percentages of the tickets that are resolved within the upper target times by ticket priorities for the tickets that belong to 'Radiologie (CS)' category. The obtained results from the simulations of the SPN representation of 'Radiologie (CS)' category are given in table 8.7. The ICT department has target resolution times by ticket priorities which do not distinguish the ticket categories. We observe that the performance of resolving 'Radiologie (CS)' tickets is significantly worse than the performance

Prediction	MAE	MSE	RMSE
SPN Representation Mean	492.12	1307180.11	1143.32
Recent History	520.93	1419869.21	1191.58
Target Resolution Times	487.26	1319493	1148.69

Table 8.6: Prediction Errors for 'Radiologie (CS)' Tickets

Priority	Upper Targets (hours)	Per. Resolved Within Upper Targets
1	2.25	6%
2	24.5	9%
3	73	48%
4	192	81%
5	336	78%

Table 8.7: Percentages of 'Radiologie (CS)' Tickets Meeting Upper Targets

of resolving 'EZIS' tickets in terms of meeting the target resolution times in all ticket priority levels. We conclude that targets that are based on ticket priority levels only are not reflected similarly in resolving different ticket categories. As it is in the resolution of 'EZIS' category tickets, meeting short resolution target times for priority levels of 1, 2 and 3 is not attainable for the large portion of the tickets of 'Radiologie (CS)'.

We consider the number of operators involved in the resolution as an another measure for assessing the performance of the ticket resolution process. We collect the number of visits to delay at each run of the SPNs of 'EZIS' and 'Radiologie (CS)' categories and obtain the average number of operators involved in the resolution of 'EZIS' and 'Radiologie (CS)' categories. The results by simulating SPNs of 'EZIS' and 'Radiologie (CS)' categories are given in table 8.8. We observe that the resolution of the 'Radiologie (CS)' category is considerably less efficient than the resolution of the 'EZIS' category in terms of the number of operators involved in resolution.

8.5 Numerical Experiments

In the tables 8.18 and 8.19, we give the operators and their performances of mean sojourn times and resolution probabilities in the resolution of 'EZIS' and 'Radiologie (CS)' category tickets that we use as inputs in SPN representations for producing ticket resolution time predictions. Our finding is that the resolution performances of both of the ticket categories are inefficient in meeting the target resolution times that is set by

Category	'EZIS'	'Radiologie (CS)'
Mean Num. of Operators Involved	2.19	3.91

Table 8.8: Mean Number of Operators Involved in 'EZIS' and 'Radiologie (CS)'

Priority	Upper Targets (hours)	SPN Pred. (hours)	Per. Res. Within Upper Targets
1	2.25	12.23	48%
2	24.5	27.92	60%
3	73	49.91	71%
4	192	27.57	99%
5	336	70.27	98%

Table 8.9: Resolution Performance of 'EZIS' with Reduced Sojourn Times at Operators

Priority	Upper Targets (hours)	SPN Pred. (hours)	Per. Res. Within Upper Targets
1	2.25	59.83	6%
2	24.5	98.49	8%
3	73	153.46	49%
4	192	95.08	82%
5	336	208.43	79%

Table 8.10: Resolution Performance of 'Radiologie (CS)' with Reduced Sojourn Times at Operators

the ICT department. In this section, we conduct experiments for 'EZIS' and 'Radiologie (CS)' categories that will improve the performances of the resolutions of the tickets and we assess the improvements in terms of the resolution times, the likelihoods of the resolutions that are within the targets and the number of operators involved per ticket.

8.5.1 Slightly Serving Faster

Firstly, we consider what would happen if the operators serve slightly faster. Since in our SPN representation approach of predicting the resolution times of the tickets we use mean sojourn times of the operators as the times that are spent at the operators, we consider reducing mean sojourn times of the operators. We obtain resolution time predictions for 'EZIS' and 'Radiologie (CS)' categories by reducing the mean sojourn time of each operator who are involved in the resolution by 10% and we obtain the percentages of the tickets that are resolved with a resolution times which are within the upper target resolution times with reduced mean sojourn times at operators. The obtained results with reduced mean sojoun times at operators for 'EZIS' and 'Radiologie (CS)' categories are given in tables 8.9 and 8.10, respectively. We find that the impact of reducing the mean sojourn times by 10% at the operators who are involved in the resolution is not significant in improving the resolution performances in both of the ticket categories.

8.5.2 Slightly Responding Faster

Other than the resolution times spent at operators, the times that are spent at delay adds to the resolution times of the tickets. We consider how will performance improve if tickets are responded slightly quicker. In SPN representation prediction approach, we use mean sojourn time at delay per ticket priority as the duration that is spent at delay.

Priority	Upper Targets (hours)	SPN Pred. (hours)	Per. Res. Within Upper Targets
1	2.25	13.27	49%
2	24.5	27.69	60%
3	73	48.86	71%
4	192	26.64	99%
5	336	65.27	98%

Table 8.11: Resolution Performance of 'EZIS' with Reduced Sojourn Times at Delay

Priority	Upper Targets (hours)	SPN Pred. (hours)	Per. Res. Within Upper Targets
1	2.25	67	6%
2	24.5	103.81	8%
3	73	150.56	49%
4	192	97.53	82%
5	336	194	79%

Table 8.12: Resolution Performance of 'Radiologie (CS)' with Reduced Sojourn Times at Delay

We consider reducing the mean sojourn time at delay by 10% for all priorities. We obtain resolution time predictions and their performances of meeting the upper targets with the reduced delay mean sojourn times. The obtained results with reduced sojourn time at delay are given in tables 8.11 and 8.12 for ticket categories 'EZIS' and 'Radiologie (CS)', respectively. We find that reducing the mean sojourn time at delay by 10% does not significantly improve the resolution performance of the tickets of both categories. We conclude that small improvements in the resolution process like 10% in serving the tickets or responding to the tickets of 'EZIS' and 'Radiologie (CS)' categories are not sufficient to achieve upper target resolution times.

8.5.3 With a New Good Operator

We investigate the effect of increasing the resources in the resolution of 'EZIS' and 'Radiologie (CS)' categories in improving their resolution performances. We consider adding a new operator in the resolution of the tickets of 'EZIS' and 'Radiologie (CS)' categories. We consider copying a good current operator in the resolution process of both ticket categories to investigate how the performance of the ticket resolution will be improved if there was an another good operator. We consider an operator as good if he/she has a high resolution probability and/or low mean sojourn resolution time. We add a new operator who has the mean sojourn time, resolution probability, external arrival rate and routing probabilities of the operator who has the minimum mean sojourn time among the operators who have the highest resolution probability among all current operators. We copy 'EZIS OP 27' and 'RAD OP 12' in 'EZIS' and 'Radiologie (CS)' ticket categories, respectively (see tables 8.18 and 8.19). We obtain predictions with the new operators and obtain the percentages of the tickets that are resolved within the upper target times. The obtained results are given in tables 8.13 and 8.14 for 'EZIS' and

Priority	Upper Targets (hours)	SPN Pred. (hours)	Per. Res. Within Upper Targets
1	2.25	13.10	50%
2	24.5	29.00	58%
3	73	52.71	69%
4	192	27.25	99%
5	336	73.86	97%

Table 8.13: Resolution Performance of 'EZIS' with a New Good Operator

Priority	Upper Targets (hours)	SPN Pred. (hours)	Per. Res. Within Upper Targets
1	2.25	62.96	6%
2	24.5	101.78	10%
3	73	160.17	49%
4	192	101.15	82%
5	336	204.63	79%

Table 8.14: Resolution Performance of 'Radiologie (CS)' with a New Good Operator

'Radiologie (CS)' categories, respectively. For both 'EZIS' and 'Radiologie (CS)' ticket categories, copying a good operator does not improve the performance of the ticket resolution significantly.

8.5.4 With A More Efficient Routing

In both ticket categories that we produce resolution time predictions, we observe that there are relatively many operators involved who have no probabilities of resolving tickets that arrive at them (see 8.18 and 8.19). Note that tickets can only be resolved by operators who have positive resolution probabilities and routing tickets to operators who are not able to resolve causes tickets bounce among operators. Therefore, routing of the tickets to operators who have zero resolution probabilities increases the number of operators involved in the resolution causing long resolution times for tickets. We investigate how the performance of the ticket resolution improves if only the operators who have positive resolution probabilities are involved in the resolution. We remove operators who are not able to resolve tickets from the resolution process and distribute the ticket traffic that arrives at them to the operators who have positive resolution probabilities equally. We obtain resolution time predictions with our SPN approach by only considering the operators who have positive resolution probabilities for 'EZIS' and 'Radiologie (CS)' ticket categories. We investigate the average number of operators involved in the resolutions and the percentages of the tickets that are resolved within upper target resolution times.

The performance improvements in the resolution of both 'EZIS' and 'Radiologie (CS)' category tickets when only considering the operators who have positive resolution probabilities are better than the performance improvements by reducing the mean sojourn times of the operators by 10%, reducing the mean sojourn delay time by 10%, and adding a new good operator in the resolution for all ticket priorities. We observe that the performance improvements in 'Radiologie (CS)' category considerably more than

Priority	Upper Targets (hours)	SPN Pred. (hours)	Per. Res. Within Upper Targets
1	2.25	12.28	51%
2	24.5	25.92	60%
3	73	43.65	73%
4	192	24.12	100%
5	336	60.96	98%

Table 8.15: Resolution Performance of 'EZIS' with Only Operators Who Have Positive Resolution Probabilities

Priority	Upper Targets (hours)	SPN Pred. (hours)	Per. Res. Within Upper Targets
1	2.25	59.78	7%
2	24.5	80.71	15%
3	73	116.35	58%
4	192	80.33	85%
5	336	146.04	88%

Table 8.16: Resolution Performance of 'Radiologie (CS)' with Only Operators Who Have Positive Resolution Probabilities

the performance improvements in 'EZIS' category in terms of the percentages of the tickets that are resolved within the targets and in terms of the average number of operators involved in the resolution (Note that there are 10 and 15 operators who have zero resolution probabilities in 'EZIS' and 'Radiologie (CS)' categories, respectively). For 'Radiologie (CS)' category, the percentages of the tickets that are resolved within the upper target resolution times increased by 10% for the tickets of priority levels of 3 and 5. On average, one less operator is involved in the resolution of the 'Radiologie (CS)' category, when only considering the operators who have positive resolution probabilities.

By removing the operators who have zero probabilities of resolving the tickets, we obtain a ticket resolution process which is more efficient in routing the tickets. We conclude that the ticket resolution performances of 'EZIS' and 'Radiologie (CS)' categories do not meet the upper target resolution times for the tickets of priority levels of 1, 2 and 3 even when the tickets are routed more efficiently among the operators for significant portions of the tickets.

Category	'EZIS'	'Radiologie (CS)'
Ave. Num. Of Operators. Involved	1.99	2.82

Table 8.17: Average Number of Operators Involved in Resolution with Only Operators Who Have Positive Resolution Probabilities

Table 8.18: Operators 'EZIS' Category

EZIS OPS	E(S) (hours)	RES PROB
EZIS OP 1	8.4915	0.1982
EZIS OP 2	11.9471	0.6879
EZIS OP 3	1.5369	0.5455
EZIS OP 4	1.2362	0.625
EZIS OP 5	0.0053	0.5417
EZIS OP 6	0.0167	0.4737
EZIS OP 7	0.0167	0
EZIS OP 8	1.2968	0.4
EZIS OP 9	0.0167	0.7586
EZIS OP 10	1.5223	0.6207
EZIS OP 11	0.0167	0.5833
EZIS OP 12	61.2548	0.6522
EZIS OP 13	0.0167	0.3182
EZIS OP 14	0.1091	0.9
EZIS OP 15	14.6232	0.3
EZIS OP 16	0.0167	0.7857
EZIS OP 17	0.0167	0.2727
EZIS OP 18	0.0167	0.2
EZIS OP 19	0.0167	0.5556
EZIS OP 20	0.0167	0.75
EZIS OP 21	0.0167	0.7143
EZIS OP 22	11.2465	0.2
EZIS OP 23	0.0167	0
EZIS OP 24	0.0167	0.3333
EZIS OP 25	0.0167	0
EZIS OP 26	0.0167	0
EZIS OP 27	0.0167	1
EZIS OP 28	103.7741	0
EZIS OP 29	0.0167	0.5
EZIS OP 30	0.0167	0
EZIS OP 31	0.0167	0
EZIS OP 32	0.0167	1
EZIS OP 33	0.0167	1
EZIS OP 34	0.0167	0
EZIS OP 35	0.0167	0
EZIS OP 36	0.0167	0

Table 8.19: Operators 'Radiologie (CS)' Category

RAD OPS	E(S) (hours)	RES PROB
RAD OP 1	11.7111	0.0833
RAD OP 2	0.8225	0.6
RAD OP 3	8.4915	0.1111
RAD OP 4	65.9788	0.4167
RAD OP 5	61.2548	0
RAD OP 6	0.0167	0
RAD OP 7	14.6231	0
RAD OP 8	0.0167	0
RAD OP 9	198.1744	0.5
RAD OP 10	0.01667	0
RAD OP 11	16.0487	0
RAD OP 12	14.5272	1
RAD OP 13	0.0167	0
RAD OP 14	0.0167	0
RAD OP 15	0.0167	0
RAD OP 16	0.0167	0
RAD OP 17	0.0167	0
RAD OP 18	0.0167	0
RAD OP 19	0.0167	0
RAD OP 20	0.0167	0
RAD OP 21	0.0167	0

9 Conclusions and Recommendations

In this chapter, we interpret the results and findings of our research on the ticket resolution process in conclusions section. Later, we give recommendations to improve the performance of the process and give further research ideas for the ticket resolution process of ICT department in LUMC.

9.1 Conclusions

In this section, we describe our interpretations of the results of our research on the ticket resolution process. We group our drawn conclusions by the technique of analysis conducted.

9.1.1 Conclusions From Current Efficiency Analysis by Process Mining

- **The handling of the tickets is inefficient**
  We obtain by process mining the event log of ticket resolution activities several indicators of inefficient ticket handling.
  - **Tickets Are Bounced Among Operators**
    We found that for the 10% of the tickets there are more than 5 operators are involved in the resolution of the tickets.
  - **Tickets Are Routed Inefficiently**
    We found by process mining tools that tickets are routed to operators who are not able resolve them.
  - **Some Tickets Stay Too Long**
    We found that 5% of the tickets are stayed open for more than one month.
  - **Short Target Resolution Times Are Not Likely to Be Met**
    Even though the tickets have high priorities and their resolution urgency is acknowledged by the operators, set short target resolution times for tickets of priorities 1 and 2 are not met.
  - **Tickets Wait Long In Transfer**
    We found from mean performance analysis with process mining that tickets experience long mean waiting times at some operators.
- **Tickets Do Not Follow Structured Paths Among Operators**
  The discovered process model from the log which contains activity names as the operator names who perform resolution activities on tickets is unstructured. Process

models per categories are considerably more structured, but they do not describe the process well either. Discovered process models do not provide insights into which activities are followed by which activities and allow a vast number of scenarios of ticket resolution process among operators.

9.1.2 Conclusions From Queueing Analysis

We draw several conclusions on the performance of the ticket resolution process from the mean queueing performance measures of operators, delay and network.

- **Operators Do Not Distinguish Priority Levels of 2, 3 and 4 In Responding**
  We draw this conclusion based on the found mean sojourn times at delay per priority. The resolution urgency of tickets of priority level 1 and the non-urgency of tickets of priority level of 5 are acknowledged by operators in responding to tickets, but for the tickets of priority levels of 2, 3 and 4 we found that response times of the operators are similar.

- **Resolution Service by Operators Varies A Lot**
  We draw this conclusion as the mean sojourn times at operators varies a lot.

- **On Average A Ticket Stays In the System For 4-4.5 Days**
  We draw this conclusion from the mean performance of the network.

9.1.3 Conclusions From Resolution Time Prediction

We state the conclusions drawn on the current performance of the ticket resolution process in terms of meeting the target resolution times per ticket priority and on the policies to improve the performance of the ticket resolution process from the numerical experiments.

- **Performance of the Ticket Resolution Process Differs Per Category**
  We obtained resolution time predictions for two ticket categories: 'EZIS' and 'Radiologie (CS)'. We observed significant differences in the performance of the ticket resolution process of these categories in terms of the percentages of the tickets which are resolved within the target resolution times in all ticket priorities.

- **SPN Representation Method Predicts the Resolution Times Better**
  The resolution time prediction method of the research which takes ticket category and priority as input predicts the resolution times better in terms of MSE and RMSE measures than the target resolution times which use ticket escalation and priority information for the ticket categories 'EZIS' and 'Radiologie (CS)'.

- **Short Target Resolution Times Are Not Likely to Be Met**
  We draw the same conclusion that we draw by process mining the event log with the resolution time predictions. In both ticket categories that we obtain resolution

time predictions, we observe that for a significant portion of the tickets with priority levels of 1, 2, and 3 have resolution times which are longer than the targets. We conclude that the set short target resolution times are not likely to be met even though the tickets have high priorities.

- **If Operators Were to Be Slightly More Efficient In Serving or Responding to Tickets, It Would Not Improve the Performance**
  We experiment the idea of how the performance of the ticket resolution improves in terms of meeting the targets if operators serve slightly faster or respond slightly faster. From our numerical experiments of considering a 10% reduction in the mean sojourn time at operators or delay, we do not observe significant improvements in the percentages of the tickets that are resolved within targets for all ticket priorities in both ticket categories of 'EZIS' and 'Radiologie (CS)'.

- **Hiring An Operator Does Not Improve the Performance**
  We do not observe any significant improvements in the performance of ticket resolution in terms of meeting the targets when there is a copy of a good current operator involved in the resolution process for both of the ticket categories that we obtain resolution time predictions.

- **Efficient Routing Improves Performance Better**
  We consider the routing of tickets to operators who have no chance of resolving inefficient. In an efficient ticket routing, tickets should be routed to operators who are able to resolve. From our experiments of considering only the operators who are able to resolve tickets in the resolution process, we obtain significant improvements in the performance in terms of meeting the targets in both ticket categories. The performance of the ticket resolution improves better by making routing efficient than making operators efficient or increasing the resources.

9.2 Recommendations

In this section, we give recommendations for the ICT department that will improve their service to resolve the ICT related questions, service requests and malfunctions of hospital employees of LUMC.

ICT has pre-determined target resolution times for the tickets at line 1 and line 2 per ticket priority. ICT department prioritizes the resolution of tickets having higher priorities over tickets having lower priorities. ICT department is inefficient in meeting the targets. We recommend ICT department to update the target resolution times in a way which reflects the performance of the ticket resolution so to prevent customer disappointment.

The target resolution times only consider the priority of the tickets. In our research, we produced predictions for tickets by using priority and category information as input. We implemented obtaining predictions for the ticket categories of 'EZIS' and 'Radiologie (CS)'. Our prediction method predicted the resolution times of the tickets of 'EZIS' and

'Radiologie (CS)' categories accurately than the target resolution times by means of MSE and RMSE accuracy measures. With our prediction approaches, we reached results that indicate significant performance deviations in the resolution of 'EZIS' and 'Radiologie (CS)'. Therefore, our first recommendation for the ICT department is to incorporate the ticket category information in determining target resolution times as the performance differ per category.

In our investigations into the performances of the ticket resolution, we found that although the handling of the tickets having high priorities are quicker, the short target resolution times that are set for tickets of priorities of 1, 2 and 3 are not likely to be met. Our second recommendation for the ICT department is to consider not having short targets such as 1 hour as it is not an attainable target with the current performance of the ticket resolution process.

Tickets are assigned to operators/operator groups that are associated to the category of the tickets randomly. The routing of the tickets among the operators/operator groups do not follow a procedure or policy. We found that the routing of the tickets is inefficient. Tickets are bounced among considerably many operators and are routed to operators who have no chance of resolving.

We investigated a routing policy which considers resolution processes of categories which only involve the operators who are able to resolve and found out that this policy improves the performance of the ticket resolution significantly. Therefore, our third recommendation is to route tickets to operators by considering the ability of the operators to resolve categories instead of routing tickets to associated operators randomly.

9.3 Further Research

In our research, we used routing sequence and classification information of the tickets in performing process mining to learn insights of the ticket resolution process from the historical instances of the process and we did not use the content of the tickets and the activities. We used category information to define service requirements of the tickets at operators.

We argue that solely ticket classification information is not sufficient to define the extent of the resolution service required to resolve the tickets as tickets are created for various of reasons. We suggest that clustering of the tickets by contents with techniques such as text mining the resolution service requirement can be better described. We also suggest clustering of the operators as a further research idea.

We found that mostly the literature on ticket resolution focuses on efficient ticket routing. In our research, we focused on measuring and predicting the performance of the ticket resolution process. Finding optimal routing policies was out of the scope of our research. We suggest conducting research on achieving an optimal automated ticket routing mechanism that will route the tickets to meet targets more.

Bibliography

[1] C.W. Gunther A.J.M.M. Weijters A. Rozinat, A.K. Alves de Medeiros and W.M.P. van der Aalst. Towards an evaluation framework for process mining algorithms. Working paper, 2007.

[2] Forest Baskett, K. Mani Chandy, Richard R. Muntz, and Fernando G. Palacios. Open, closed, and mixed networks of queues with different classes of customers. *Journal of the ACM*, 22(2):248–260, 1975.

[3] Mohan Raj Chinnaswamy, Manjunath Kamath, and Robert A. Greve. Queueing models for analyzing customer contact center operations.

[4] Wil M.P.van der Aalst. *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Springer, 2011.

[5] Wil M.P.van der Aalst. *Process Mining: Data Science in Action*. Springer, 2016.

[6] Francesco Folino, Massimo Guarascio, and Luigi Pontieri. *Discovering High-Level Performance Models for Ticket Resolution Processes*, pages 275–282. Springer Berlin Heidelberg, 2013.

[7] Rouba Ibrahim and Ward Whitt. Real-time delay estimation based on delay history. *Manufacturing & Service Operations Management*, 11(3):397–415, 2009.

[8] Gert Janssenswillen, Niels Donders, Toon Jouck, and Benoit Depaire. A comparative study of existing quality measures for process discovery. *Information Systems*, 71:1 – 15, 2017.

[9] K. Krinkin, E. Kalishenko, and S.P.S. Prakash. Process mining approach for traffic analysis in wireless mesh networks. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7469 LNCS:260–269, 2012.

[10] Geetika T. Lakshmanan, Davood Shamsi, Yurdaer N. Doganata, Merve Unuvar, and Rania Khalaf. A markov prediction model for data-driven semi-structured business processes. *Knowledge and Information Systems*, 42(1):97–126, 2015.

[11] Sander J.J. Leemans, Dirk Fahland, and Wil M.P. van der Aalst. Process and deviation exploration with inductive visual miner. `http://www.processmining.org/_media/blogs/pub2014/bpmdemoleemans.pdf`, 2014.

[12] John D. C. Little. A proof for the queuing formula: L = λw. *Operations Research*, 9(3):383–387, 1961.

[13] L.T. Ly, S. Rinderle, P. Dadam, and M. Reichert. Mining staff assignment rules from event-based data. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 3812 LNCS:177–190, 2005.

[14] R. Mans, H. Schonenberg, G. Leonardi, S. Panzarasa, A. Cavallini, S. Quaglini, and W. Van Der Aalst. Process mining techniques: An application to stroke care. volume 136, pages 573–578, 2008.

[15] R. S. Mans, M. H. Schonenberg, M. Song, W. M. P. van der Aalst, and P. J. M. Bakker. *Application of Process Mining in Healthcare – A Case Study in a Dutch Hospital*, pages 425–438. Springer Berlin Heidelberg, 2009.

[16] Gengxin Miao, Louise E. Moser, Xifeng Yan, Shu Tao, Yi Chen, and Nikos Anerousis. Generative models for ticket resolution in expert networks. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 733–742. ACM, 2010.

[17] M. polato, A. Sperduti, A. Burattin, and M. de Leoni. Time and activity sequence prediction of business process instances. *pre-print*, 2016. arXiv:1602.07566.

[18] S. Pravilovic, A. Appice, and D. Malerba. Process mining to forecast the future of running cases. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8399 LNAI:67–81, 2014.

[19] 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.

[20] Andreas Rogge-Solti and Mathias Weske. *Prediction of Remaining Service Execution Time Using Stochastic Petri Nets with Arbitrary Firing Delays*, pages 389–403. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[21] Rob J.B. Vanwersch Ronny S. Mans, Wil M.P.van der Aalst. *Process Mining in Healthcare: Evaluating and Exploiting Operational Healthcare Processes*. Springer, 2015.

[22] Marcella Rovani, Fabrizio M. Maggi, Massimiliano de Leoni, and Wil M.P. van der Aalst. Declarative process mining in healthcare. *Expert Systems with Applications*, 42(23):9236–9251, 2015.

[23] A. Rozinat, I. S. M. De Jong, C. W. Günther, and W. M. P. Van Der Aalst. Process mining applied to the test process of wafer scanners in asml. *Transactions on Systems, Man, and Cybernetics, Part C*, 39(4):474–479, 2009.

[24] V. Rubin, C.W. Günther, W.M.P. Van Der Aalst, E. Kindler, B.F. Van Dongen, and W. Schäfer. Process mining framework for software processes. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4470 LNCS:169–181, 2007.

[25] A. Senderovich, M. Weidlich, L. Yedidsion, A. Gal, A. Mandelbaum, S. Kadish, and C.A. Bunnell. Conformance checking and performance improvement in scheduled processes: A queueing-network perspective. *Information Systems*, 62:185–206, 2016.

[26] Arik Senderovich, Sander J. J. Leemans, Shahar Harel, Avigdor Gal, Avishai Mandelbaum, and Wil M. P. van der Aalst. *Discovering Queues from Event Logs with Varying Levels of Information*, pages 154–166. Springer International Publishing, 2016.

[27] Arik Senderovich, Matthias Weidlich, Avigdor Gal, and Avishai Mandelbaum. *Queue Mining – Predicting Delays in Service Processes*, pages 42–57. Springer International Publishing, 2014.

[28] Qihong Shao, Yi Chen, Shu Tao, Xifeng Yan, and Nikos Anerousis. Efficient ticket routing by resolution sequence mining. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 605–613. ACM, 2008.

[29] Peng Sun, Shu Tao, Xifeng Yan, Nikos Anerousis, and Yi Chen. *Content-Aware Resolution Sequence Mining for Ticket Routing*, pages 243–259. Springer Berlin Heidelberg, 2010.

[30] W. M. P. van der Aalst, H. A. Reijers, A. J. M. M. Weijters, B. F. van Dongen, A. K. Alves de Medeiros, M. Song, and H. M. W. Verbeek. Business process mining: An industrial application. *Information Systems*, 32(5):713–732, 2007.

[31] W. M. P. van der Aalst, M. H. Schonenberg, and M. Song. Time prediction based on process mining. *Information Systems*, 36(2):450–475, 2011.

[32] 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.

[33] Wil M. P. van der Aalst. *Extracting Event Data from Databases to Unleash Process Mining*, pages 105–128. Springer International Publishing, 2015.

[34] W.M.P. Van Der Aalst and A.K.A. De Medeiros. Process mining and security: Detecting anomalous process executions and checking process conformance. *Electronic Notes in Theoretical Computer Science*, 121(SPEC. ISS.):3–21, 2005.

[35] S.K.L.M. Vanden Broucke, J. De Weerdt, J. Vanthienen, and B. Baesens. A comprehensive benchmarking framework (cobefra) for conformance analysis between procedural process models and event logs in prom. pages 254–261, 2013.

[36] H.M.W. (Eric) Verbeek and R. P. Jagadeesh Chandra Bose. Prom 6 tutorial. `http://www.promtools.org/prom6/downloads/prom-6.0-tutorial.pdf`, 2010.

[37] Wayne L. Winston. *Simulation Modeling Using @Risk*. Duxbury, 2nd edition, 2000.

[38] T. Yampaka and P. Chongstitvatana. An application of process mining for queueing system in health service. In *2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pages 1–6, 2016.

Appendices

1 Keywords and Abbreviations

Keyword	Meaning
Ticket	An incident notification
Incident management	Management of life-cycles of incidents
Service desk	First place that ICT related incidents and requests are registered and handled
Service desk employee	ICT employee at first line
Operator	Specialist ICT employee at second line
Resolution Time	Time from register till resolution

Abbreviation	Meaning
ICT	Information and Communication Technologies
LUMC	Leiden University Medical Center
SPN	Stochastic Petri Nets
EZIS	Electronic Hospital Information System
MSE	Mean Squared Error
MAE	Mean Absolute Error
RMSE	Root Mean Squared Error

2 Queue Input Parameters and Queue Performance Analyses

2.1
Queue Input Parameters and Mean Performance Measures of Delay

priority	$\lambda_{p,D}$	$\mu^D_{p,normal}$	$\mu^D_{p,heavy}$	$E(L^p_0)$	$E(S^p_0)$
1	0.0040	28.2	28.2	0.0001	0.0354
2	0.2043	0.08	0.07	2.827	13.8374
3	0.6317	0.09	0.03	21.0454	33.3144
4	1.2083	0.07	0.08	15.6275	12.9331
5	11.3562	0.03	0.02	567.8	49.9992

2.2
Queue Input Parameters and Mean Performance Measures of Operators

Table A.2: Type 1 Operators

Operators	λ_i^0	Service Rate	$E(L_i^0)$
ADM	0.0054	60	0.0001
ADU	0.0013	60	0.0000
ASG	0.004	60	0.0001
AST	0.0282	60	0.0005
BEL	0.0027	60	0.0000
BIU	0.0296	60	0.0005
BTF	0.0013	60	0.0000
DBB	0.0013	60	0.0000
DCFB	0.0202	60	0.0003
DCFN	0.0027	60	0.0000
DCO	0.0013	60	0.0000
DI3	0.0081	60	0.0001
DIV4	0.0027	60	0.0000
DRU	0.0013	60	0.0000
DSL	0.0134	60	0.0002
EDK	0.0013	60	0.0000
EMH	0.0013	60	0.0000
EOFB	0.1156	60	0.0019
EXTR	0.6116	60	0.0102
FMF	0.0081	60	0.0001
FVHW	0.0134	60	0.0002
GLI	0.0175	60	0.0003
GMT	0.0323	60	0.0005
GWI	0.0027	60	0.0000
HFK	0.0013	60	0.0000
HJW	0.004	60	0.0001
HPD	0.0027	60	0.0000
HRV	0.0013	60	0.0000
HST	0.0013	60	0.0000
HZI	0.0188	60	0.0003
IBBH	0.0444	60	0.0007
IBO	0.129	60	0.0022
IBV	0.0833	60	0.0014
ICIT	0.0645	60	0.0011
ID4	0.004	60	0.0001
IDM	0.082	60	0.0014
IEO	0.0013	60	0.0000
IEX	0.1761	60	0.0029
IHO	0.0054	60	0.0001
INT	0.0874	60	0.0015

IPBP	0.039	60	0.0007
IPHR	0.0497	60	0.0008
IPP	0.0363	60	0.0006
IPRB	0.0309	60	0.0005
IPSBP	0.0376	60	0.0006
ISC	0.039	60	0.0007
ISE	0.0685	60	0.0011
ISHTE	0.0121	60	0.0002
ISTO	0.0659	60	0.0011
ITO	0.0282	60	0.0005
JCU	0.004	60	0.0001
JIE	0.0067	60	0.0001
JMO	0.0027	60	0.0000
JMVB	0.0081	60	0.0001
JNB	0.0296	60	0.0005
KFB	0.0538	60	0.0009
KIB	0.0188	60	0.0003
KTS	0.0013	60	0.0000
LOP	0.7863	60	0.0131
LWA	0.0013	60	0.0000
MDZ	0.0054	60	0.0001
MHO	0.0013	60	0.0000
MIR	0.0067	60	0.0001
MJV	0.0013	60	0.0000
MKH	0.0027	60	0.0000
MST	0.0013	60	0.0000
ONS	0.0376	60	0.0006
RJC	0.0013	60	0.0000
RJPS	0.0134	60	0.0002
RPN	0.0013	60	0.0000
RRSD	0.0134	60	0.0002
RSN	0.0013	60	0.0000
RTL	0.0013	60	0.0000
SCR	0.0013	60	0.0000
SDF	0.0013	60	0.0000
SHW	0.0094	60	0.0002
SJE	0.0013	60	0.0000
SJP	0.0054	60	0.0001
SOT	0.004	60	0.0001
SQL	0.0376	60	0.0006
TEL	0.0538	60	0.0009
WAL	0.0013	60	0.0000
WHT	0.0067	60	0.0001

WON	0.0054	60	0.0001

Table A.3: Type 2 Operators

Operators	λ_i^0	$E(L_i^0)$	λ_i^+	$\mu^R_{i,+,normal}$	$\mu^R_{i,+,heavy}$	$E(L_i^+)$	$E(S_i^+)$	$P(S_i > 1hr)$
BB	0.0000	0.0000	0.0054	0.9195	0.9195	0.0059	1.0875	0.3987
BGK	0.0121	0.0002	0.0175	0.0694	0.0694	0.2523	14.4167	0.9330
CHO	0.0013	0.0000	0.0013	5.4546	5.4546	0.0002	0.1833	0.0043
COW	0.0094	0.0002	0.0013	0.4003	0.4003	0.0032	2.4979	0.6701
CPO	0.0000	0.0000	0.0054	0.1625	0.1625	0.0332	6.1541	0.8500
DAM	0.0000	0.0000	0.0013	0.0073	0.0073	0.1779	136.8667	0.9927
DMW	0.0000	0.0000	0.0094	0.0471	0.0471	0.1996	21.2310	0.9540
ES	0.0000	0.0000	0.0040	0.0073	0.0073	0.5443	136.0833	0.9927
EST	0.0000	0.0000	0.0067	0.0051	0.0051	1.3243	197.6632	0.9950
FEL	0.0175	0.0003	0.0040	0.1090	0.1090	0.0367	9.1729	0.8967
FMB	0.0000	0.0000	0.0054	0.0043	0.0043	1.2583	233.0198	0.9957
GER	0.0175	0.0003	0.0094	7.4534	7.4534	0.0013	0.1342	0.0006
HPS	0.0000	0.0000	0.0013	0.5505	0.5505	0.0024	1.8167	0.5767
HRA	0.0054	0.0001	0.0013	0.0011	0.0011	1.1658	896.7484	0.9989
IDV	0.0000	0.0000	0.0013	0.0007	0.0007	1.8022	1386.3165	0.9993
JME	0.0148	0.0002	0.0040	420.0000	420.0000	0.0000	0.0024	0.0000
JTG	0.0067	0.0001	0.0013	2.2785	2.2785	0.0006	0.4389	0.1024
JWS	0.0000	0.0000	0.0013	0.3030	0.3030	0.0043	3.3000	0.7386
KDJ	0.0000	0.0000	0.0027	0.0775	0.0775	0.0349	12.9083	0.9255
KEW	0.0094	0.0002	0.0215	0.3045	0.3045	0.0706	3.2841	0.7375
KME	0.0040	0.0001	0.0027	0.0020	0.0020	1.3476	499.1290	0.9980
MAN	0.0027	0.0000	0.0094	0.0096	0.0096	0.9755	103.7741	0.9904
MBJ	0.0013	0.0000	0.0013	0.0119	0.0119	0.1093	84.0499	0.9882
MBU	0.0000	0.0000	0.0202	69.2307	69.2307	0.0003	0.0144	0.0000
MDR	0.0121	0.0002	0.0027	0.0027	0.0027	1.0003	370.4898	0.9973
MDI	0.0000	0.0000	0.0081	0.0090	0.0090	0.8999	111.1028	0.9910
MGR	0.0000	0.0000	0.0013	0.0014	0.0014	0.9620	739.9679	0.9986
MGS	0.0013	0.0000	0.0081	0.2490	0.2490	0.0325	4.0167	0.7796
MJS	0.0000	0.0000	0.0081	0.0177	0.0177	0.4578	56.5167	0.9825
MP	0.0000	0.0000	0.0054	0.2281	0.2281	0.0237	4.3833	0.7960
NAJ	0.0027	0.0000	0.0013	0.0324	0.0324	0.0401	30.8667	0.9681
NKR	0.0027	0.0000	0.0108	0.0127	0.0127	0.8526	78.9465	0.9874
OEK	0.0000	0.0000	0.0040	0.0178	0.0178	0.2250	56.2613	0.9824
ORT	0.0000	0.0000	0.0040	0.0143	0.0143	0.2798	69.9443	0.9858
PG	0.0013	0.0000	0.0054	0.0659	0.0659	0.0819	15.1633	0.9362
PGA	0.0000	0.0000	0.0013	0.0002	0.0002	6.8351	5257.7894	0.9998
PMM	0.0000	0.0000	0.0054	0.0049	0.0049	1.0961	202.9752	0.9951
RB	0.0000	0.0000	0.0269	0.0466	0.0466	0.5772	21.4575	0.9545

RBLE	0.0013	0.0000	0.0013	9.2308	9.2308	0.0001	0.1083	0.0001
SA	0.0000	0.0000	0.0081	120.0000	120.0000	0.0001	0.0083	0.0000
STE	0.0054	0.0001	0.0202	0.6734	0.6734	0.0300	1.4851	0.5100
VEL	0.0000	0.0000	0.0040	0.1619	0.1619	0.0247	6.1778	0.8506
VPR	0.0094	0.0002	0.0108	0.0988	0.0988	0.1093	10.1233	0.9059
VZH	0.0013	0.0000	0.0202	0.0952	0.0952	0.2122	10.5052	0.9092

Table A.4: Type 3 Operators

Operators	λ_i^+	$\mu^R_{i,+,normal}$	$\mu^R_{i,+,heavy}$	$E(L_i^+)$	$E(S_i^+)$	π_{heavy^i}	π_{normal^i}	$P(S_i > 1hr)$
ACK	0.0121	0.0803	0.0158	0.7652	63.2367	0.5347	0.4653	0.9843
AIB	0.6949	0.4364	0.6524	1.0680	1.5369	0.3262	0.6738	0.7445
AKG	0.0054	120.0000	0.0413	0.1307	24.2083	0.1225	0.8775	0.9595
AMV	0.0161	0.0023	2.5668	0.0063	0.3896	0.0063	0.9937	0.0768
AVG	0.0094	0.0056	28.8889	0.0003	0.0346	0.0003	0.9997	0.0000
BDE	0.0215	0.1907	0.0840	0.1289	5.9959	0.0139	0.9861	0.8310
BRJ	0.1976	0.2028	0.1529	1.1857	6.0006	0.3400	0.6600	0.8975
CAZ	0.0040	0.0136	0.0060	0.6705	167.6170	0.4885	0.5115	0.9941
CCE	0.0202	0.0102	0.0064	3.1377	155.3330	0.9566	0.0434	0.9936
CDB	0.0175	0.0033	0.0103	1.9444	111.1080	0.5797	0.4203	0.9960
CMI	0.0148	0.0556	0.0295	0.5015	33.8857	0.3944	0.6056	0.9709
CTH	0.0632	0.0211	0.0158	3.3984	53.7714	0.7671	0.2329	0.9896
CTK	0.1478	0.0866	0.0315	4.6847	31.6962	0.9908	0.0092	0.9689
DA	0.0215	0.0602	0.0602	0.3573	16.6188	0.3004	0.6996	0.9416
DHA	0.2043	0.0601	0.0576	2.9875	14.6232	0.7312	0.2688	0.9652
DJS	0.0430	0.0041	0.0050	8.5215	198.1740	0.9981	0.0019	0.9950
DK	0.0067	0.0243	60.0000	0.0001	0.0167	0.0001	0.9999	0.0000
DMBR	0.0161	0.0200	1.9277	0.0084	0.5188	0.0083	0.9917	0.1455
DPL	0.1358	0.5257	26.1655	0.0052	0.0382	0.0052	0.9948	0.0000
DVK	0.0067	0.4970	0.4970	0.0135	2.0122	0.0134	0.9866	0.6084
DWB	0.0161	0.0065	0.0065	2.4612	152.8680	0.9147	0.0853	0.9935
EAJ	0.0847	1.2344	0.5115	0.1656	1.9551	0.1526	0.8474	0.5996
EGL	0.0349	0.0239	0.0134	2.4542	70.3211	0.6921	0.3079	0.9913
EJG	0.0161	0.0575	0.0575	0.2801	17.3972	0.2443	0.7557	0.9441
EJP	0.0121	0.0059	0.0666	0.7586	62.6938	0.0611	0.9389	0.9946
EMP	0.2245	6.1871	0.1409	1.5935	7.0979	0.7968	0.2032	0.8686
ERZ	0.0349	0.0063	0.0031	11.2462	322.2400	1.0000	0.0000	0.9969
FHG	0.0228	0.0265	0.0265	0.8607	37.7482	0.5771	0.4229	0.9739
FL	0.0296	0.2291	0.0669	0.4421	14.9367	0.3573	0.6427	0.9352
FMH	0.0484	0.0102	0.0541	1.3386	27.6579	0.3373	0.6627	0.9894
FPE	0.0497	0.0661	0.0918	0.5414	10.8928	0.4181	0.5819	0.9123
GAW	0.0739	0.0756	0.0935	0.8655	11.7111	0.2056	0.7944	0.9492
GBJ	0.0659	0.0127	0.0091	7.0718	107.3110	0.9728	0.0272	0.9914

GHO	0.0067	0.0132	0.0103	0.5687	84.8854	0.1215	0.8785	0.9898
GJR	0.0054	0.0458	0.0032	1.6732	309.8500	0.8124	0.1876	0.9968
GM	0.5040	0.3604	0.2219	1.5287	3.0331	0.4188	0.5812	0.8468
HAB	0.0874	0.0424	0.2838	0.8216	9.4000	0.1033	0.8967	0.9630
HCK	0.3642	0.0423	0.0589	4.3512	11.9471	0.9175	0.0825	0.9514
HIF	0.1048	0.0101	0.0163	6.4195	61.2548	0.9885	0.0115	0.9842
HOL	0.1841	0.6158	0.6158	0.2989	1.6238	0.2584	0.7416	0.5402
HZE	0.0094	0.0315	0.0062	1.5096	160.5900	0.7790	0.2210	0.9938
IDS	0.0699	0.0120	0.0332	1.0029	14.3473	0.7875	0.2125	0.9791
IFO	0.1116	3.4866	3.4866	0.0320	0.2868	0.0315	0.9685	0.0306
IMA	0.0161	0.0152	0.0152	1.0623	65.9788	0.6543	0.3457	0.9850
IST	0.2567	0.2360	2.1120	0.5694	2.2180	0.0319	0.9681	0.7943
JB	0.0565	0.0075	0.0187	2.6801	47.4353	0.7358	0.2642	0.9894
JBM	0.0632	0.0149	0.0129	4.9005	77.5390	0.9552	0.0448	0.9883
JEF	0.0067	0.0061	0.0061	1.0952	163.4640	0.6655	0.3345	0.9939
JFZ	0.0349	0.0744	0.0744	0.4690	13.4387	0.3744	0.6256	0.9283
JJ	0.0228	0.0053	0.0588	0.3879	17.0150	0.3215	0.6785	0.9429
JMP	0.0081	0.0242	0.0836	0.0969	11.9667	0.0924	0.9076	0.9198
JMW	0.0188	0.3583	1.3793	0.0136	0.7250	0.0135	0.9865	0.2518
JN	0.0323	0.0285	0.0226	1.4279	44.2072	0.7602	0.2398	0.9776
JP	0.0323	0.2307	0.1029	0.3140	9.7214	0.2695	0.7305	0.9022
JPG	0.0860	0.0110	0.2341	0.3674	4.2716	0.3074	0.6926	0.7913
JPN	0.0470	0.0042	0.0044	10.7812	229.3870	1.0000	0.0000	0.9957
JZB	0.0081	0.0070	0.0199	0.4072	50.2750	0.3345	0.6655	0.9803
KF	0.0538	0.1021	0.1520	0.3540	6.5794	0.2981	0.7019	0.8590
KGTF	0.2070	0.0068	0.1329	1.9463	9.4024	0.5764	0.4236	0.9572
KK	0.0470	0.0596	0.0473	0.9935	21.1380	0.6297	0.3703	0.9538
KK	0.1546	1.3459	3.2394	0.0477	0.3087	0.0466	0.9534	0.0392
KR	0.5323	3.1993	0.6569	0.8103	1.5223	0.5553	0.4447	0.5185
LAE	0.1048	0.0382	0.0050	21.1127	201.4570	1.0000	0.0000	0.9950
LANE	0.0685	0.0175	0.1329	0.5156	7.5268	0.4028	0.5972	0.8756
LEL	0.0188	0.0135	2.5767	0.5858	31.1569	0.0021	0.9979	0.9866
LEZ	0.0578	0.0062	0.9063	0.5604	9.6959	0.7922	0.2078	0.6250
MAET	0.0054	0.0207	0.0004	14.4851	2682.4200	1.0000	0.0000	0.9996
MAR	0.1196	0.2134	0.0623	1.9194	16.0487	0.8533	0.1467	0.9396
MAS	0.0470	0.0061	0.0065	7.2603	154.4750	0.9940	0.0060	0.9936
MBS	0.0148	0.6918	0.0235	0.6286	42.4733	0.4667	0.5333	0.9767
MBZ	0.0309	1.0155	1.0155	0.0304	0.9848	0.0300	0.9700	0.3622
MDE	0.0067	0.0097	0.0097	0.6873	102.5890	0.4971	0.5029	0.9903
MDS	0.1747	0.9501	0.3621	0.4824	2.7615	0.3827	0.6173	0.6962
MDU	0.0013	0.0064	0.0064	0.2030	156.1500	0.1837	0.8163	0.9936
MEM	0.0067	0.1448	0.0124	0.5421	80.9083	0.4185	0.5815	0.9877
MJA	0.0645	0.1413	0.5882	0.1097	1.7000	0.1039	0.8961	0.5553

MKU	0.0148	0.0537	0.0429	0.3448	23.3000	0.2917	0.7083	0.9580
MMB	0.8132	0.2042	0.3535	1.0546	1.2968	0.7517	0.2483	0.8092
MPU	0.0820	0.0176	0.0509	1.8538	22.6078	0.5508	0.4492	0.9812
MSC	0.8401	0.4704	0.2302	0.0044	0.0052	0.9980	0.0020	0.7943
NDK	0.0054	0.0310	0.0175	0.3079	57.0133	0.2650	0.7350	0.9826
NFK	0.0349	0.0199	0.0055	6.3597	182.2270	0.9983	0.0017	0.9945
NGH	0.1062	0.0383	0.0619	1.5428	14.5272	0.4938	0.5062	0.9752
PC	0.0228	0.2587	16.3636	0.0014	0.0611	0.0014	0.9986	0.0000
PHH	0.1156	0.0466	0.0353	2.6125	22.5997	0.6604	0.3396	0.9793
PK	0.0067	0.1043	0.1043	0.0642	9.5872	0.0622	0.9378	0.9010
PKP	0.0081	0.3176	0.3176	0.0255	3.1487	0.0252	0.9748	0.7279
PWE	0.0403	0.1271	0.0982	0.4106	10.1883	0.3367	0.6633	0.9065
RAE	0.0753	0.0091	0.1867	1.1067	14.6975	0.1713	0.8287	0.9886
RAS	0.0632	0.0673	0.1143	0.5528	8.7464	0.4246	0.5754	0.8920
RBB	0.6734	1.8695	0.4728	0.8324	1.2362	0.2435	0.7565	0.4133
RJR	0.0148	0.0599	13.6364	0.0011	0.0733	0.0011	0.9989	0.0000
RK	0.0242	0.1110	0.0409	0.5919	24.4568	0.4467	0.5533	0.9599
RST	0.1841	0.0065	0.0862	2.0705	11.2465	0.6852	0.3148	0.9605
RTE	0.0134	0.0327	7.0588	0.0019	0.1417	0.0019	0.9981	0.0009
RTI	0.0390	0.0103	0.0276	1.6658	42.7117	0.4864	0.5136	0.9908
RZV	0.0323	0.0757	0.0965	0.3346	10.3583	0.2844	0.7156	0.9080
SD	0.0511	0.0155	1.2159	0.0420	0.8225	0.0412	0.9588	0.2965
SDI	0.0739	0.1960	0.0365	1.2807	17.3299	0.3803	0.6198	0.9249
SHE	0.0981	0.0098	0.0238	0.6264	6.3851	0.9462	0.0538	0.9789
SKB	0.0497	0.0617	0.0219	1.9134	38.4992	0.5580	0.4420	0.9812
SL	0.0296	0.0365	0.0320	0.9249	31.2478	0.6034	0.3966	0.9685
SLJ	0.0430	0.0041	0.0131	3.3680	78.3248	0.8613	0.1387	0.9903
SM	0.0551	0.5290	0.0770	0.7156	12.9881	0.5111	0.4889	0.9259
SMR	0.0349	0.0783	0.0783	0.4459	12.7755	0.3597	0.6403	0.9247
SMW	0.0108	0.0465	0.0021	5.1786	479.5000	0.9944	0.0056	0.9979
SNU	0.0161	0.0019	0.0067	2.4075	149.5350	0.9100	0.0900	0.9933
SO	0.0296	0.0283	0.5491	0.0539	1.8211	0.0525	0.9475	0.5775
SRB	0.0390	0.0549	0.0549	0.7106	18.2195	0.5086	0.4914	0.9466
SSC	0.0121	0.0199	0.1440	0.0840	6.9452	0.0806	0.9194	0.8659
SSM	0.1599	0.0888	3.0068	0.6660	4.1649	0.0171	0.9829	0.9157
STH	0.0296	0.0205	0.0062	3.8340	129.5270	0.7491	0.2509	0.9952
ST	0.5766	0.3216	0.1036	0.9285	1.6104	0.7581	0.2419	0.9220
TAW	0.0108	0.3979	0.3979	0.0271	2.5133	0.0268	0.9732	0.6717
TDO	0.0565	0.0166	0.0812	1.1522	20.3922	0.2562	0.7438	0.9855
TJ	0.0296	0.0025	0.0133	2.0957	70.8009	0.6727	0.3273	0.9937
TSP	0.0605	0.0093	0.0414	1.7819	29.4523	0.5233	0.4767	0.9867
TTN	0.0161	0.0125	1.0000	0.0161	1.0000	0.0160	0.9840	0.3679
TU	0.0108	0.0315	0.0315	0.3431	31.7695	0.2904	0.7096	0.9690

VAA	0.0215	0.1066	0.9917	0.0217	1.0083	0.0214	0.9786	0.3709
VDN	0.2742	2.6388	0.3282	0.2061	0.7516	0.0503	0.9497	0.1439
VFM	0.4651	0.0379	0.0552	3.9494	8.4915	0.9692	0.0308	0.9494
VM	0.0457	0.0115	1.2562	0.0364	0.7961	0.0357	0.9643	0.2847
WDV	0.0323	0.1019	3.7500	0.0086	0.2667	0.0086	0.9914	0.0235
WJG	0.0148	0.0982	0.0982	0.1507	10.1792	0.1399	0.8601	0.9064
WRL	0.3347	2.0022	9.1679	0.0365	0.1091	0.0358	0.9642	0.0001