MASTER

Comparing, analysing, and optimising business process variants

Samuel, J.

Award date:
2015

Technische Universiteit Eindhoven University of Technology
Department of Mathematics and Computer Science
Architecture of Information Systems Research Group

Comparing, Analysing, and Optimising Business Process Variants

Master thesis

Joelian Samuel
Student number 0872415

Supervisors:
Prof. dr. ir. H.A. Reijers
D.M.M. Schunselaar MSc.
Caspar Jans MSc. (DSM)

Final document

DSM
BRIGHT SCIENCE. BRIGHTER LIVING.

Eindhoven, 31 August 2015

Abstract

It is common when a specific business process in one organization has many similarities with a related business process in other organizations. However, even though these business processes might look the same, they are actually different because of the unique characteristics of the organization. This is what we called business process variants. This condition brings an opportunity for the organization to optimize their business process by learning from the business process variant. However, in most cases, this task is a time-consuming and labour-intensive task. PETRA overcomes this challenge by providing a toolset to analyse these business process variants in an automated way which resulted a new process model that performs better compared to the existing business process. PETRA aims to be a generic and extensible framework such that it can be applied in any domains. Nevertheless, up to now, PETRA has only been tested in the municipality domain. This motivates us to investigate the applicability of PETRA in another domain. This thesis reported another case study that using PETRA in a multinational company, i.e. DSM. In order to get the business process variants, our study took input from three different business units within DSM. Our study confirmed that PETRA might be applicable to the multinational company domain and provided several scenarios of PETRA application within DSM.

Keywords: process variants, PETRA, applicability of PETRA, case study, multinational company.

Preface

This master thesis is the final milestone in order to achieve my Master of Science degree in Business Information Systems at the Eindhoven University of Technology.

The fulfilment of this thesis was not possible without the support from of some people, and I would like to thank them all. First of all, I would like to thank prof. Hajo Reijers, my first supervisor, for the opportunity to work in this project and all the supports, motivations, and guidance throughout the whole project. Next, I would also like to thank Dennis Schunselaar, my daily supervisor and mentor, for all the precious feedbacks and discussions we had during the project. I also would like to thank all the people at PA&S department at DSM and especially Caspar Jans, my supervisor from DSM, for the support and cooperation during this project.

I also would like to express my thanks to my friends, especially to the member of 'EHV Perjuangan'; thanks for the support, sharing, and feedbacks. Thank you for many events we shared together and thank you for being my family in this country for the last two years.

Last, I would like to thank my family, for always being there when I needed them. Their motivation over the last few months gave me the extra energy to finish this thesis. Without them I would have definitely not made this far.

Joelian Samuel

Eindhoven, August 2015

Table of Contents

List of Tables

List of Figures

Chapter 1. Introduction

This master thesis is carried out as part of the Master Business Information Systems (BIS). This project is a collaboration between DSM and the Architecture of Information System (AIS) group of the Mathematics and Computer Science Department of the Eindhoven University Technology (TU/e).

1.1. Background

It is common when a specific business process in one organization has many similarities with a related business process in other organizations, e.g., a purchase order process. A purchase order process usually starts with a request from a user and most of the time it includes an approval process. However, even though all purchase orders might look the same, they end up being different due to the unique characteristics of its organization. This can be seen within the CoSeLoG project [27]. In the CoSeLoG project, there are 10 municipalities and each of them has modelled the same business process differently. The difference of one particular business process between municipalities is caused by the "couleur locale". These differences are not necessarily bad as it now also becomes possible to combine various differences into a new model which is not (yet) used by any of the municipalities. This new model can actually be an improvement to the current model of a municipality. These variants are combined into one configurable process models. Furthermore, the configurable process model is used as input by PETRA so that it can help municipalities improve the business process. PETRA is a generic tool to explore a space of possible model by repeatedly analysing the parameters of the process model and giving the comparison of analysis result [1]. In [1], PETRA has been tested using the building permit case study by taking an input from two different municipalities. Petra yields eight variants of the process model and provides one optimal process model with the shortest throughput time. Figure 1 illustrates the position of Petra in the building permit case study.

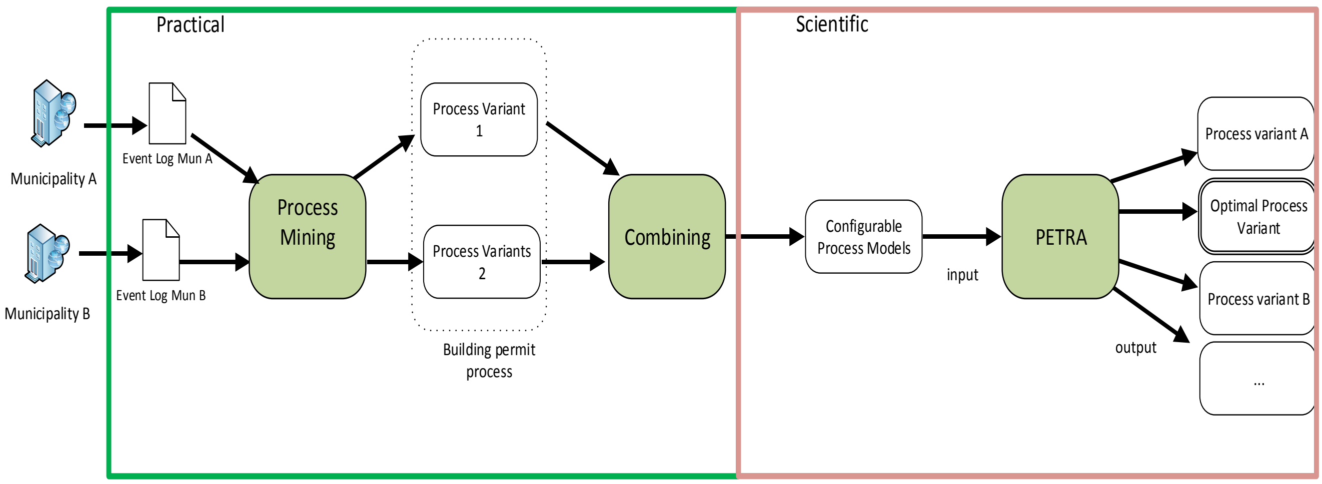


Figure 1 the building permit case study

The similar challenges within the CoSeLoG project also apply to DSM. DSM has several business units according to the product/market combination. These different business units run the same business process, but in different ways. Therefore, the aim of this project is to apply techniques developed in the CoSeLoG project within DSM. The project will have the practical and scientific part. The practical part, a prerequisite of the scientific part, will be the comparison and analysis of the process model. Meanwhile, the scientific part will try to apply techniques from the CoSeLoG project to find the process model(s) that has a better score on their performance indicator.

1.2. Problem Description

PETRA is a generic and extensible framework with a goal, it can be applied in any domains. Nevertheless, up to now, PETRA has only been tested in the CoSeLoG project. As such, it will be interesting to investigate the applicability of PETRA in another domain. Therefore, our main research question is:

How applicable is PETRA framework outside the domain of municipalities (e.g. multinational companies)?

In order to answer the main research question, we conducted a study using PETRA outside the domain of municipalities. The DSM case is a good opportunity for validating PETRA's applicability in a multinational company case. However, we cannot directly answer our main question because DSM does not have the variants of process models as the input for PETRA. Therefore, in order to tackle the main research question, we derive the main research question into three sub research questions:

a. *How can we discover the process models variants within DSM?*
b. *What are the variation point of the process model variants within DSM?*
c. *How can the generated process model variants be used as input for PETRA?*

1.3. Research scope

DSM as a multinational company has many different business processes. Therefore, in this study, we put our focus on the sales business process domain and specifically the Order to Cash (OTC) business process. In this project, we limit our scope to the OTC process because it has a sufficient amount of data to begin with. The OTC process is a family of business processes related to goods or services that DSM sells. In Figure 2, the high level abstraction of the OTC process is given. It has five sub processes from order intake to cash in. The process starts with the manual input of the process order by sales employee in the *Order Intake* sub process. After this process, the delivery plan are created either automatically or manually in the *Delivery* sub process. The delivery plan determines when the order should be delivered to the customer considering the availability of stock and the transportation. According to these plans, an order is distributed in the *Distribute* sub process. Next, the invoice is created in the *Bill* sub process. Last, DSM receives the payment from the customer in the *Cash In* sub process. In order to get a variety of business processes, we take the data from three different business units in DSM, so called BU1, BU2, and BU3. Besides that, we only consider orders which have been completely processed, i.e. for which the *Cash In* process has been executed.

Figure 2 high-level abstraction of OTC reference control flow

1.4. Methodology

Given the research question and the scope, Figure 3 shows the methodology that is used in this project. The methodology consists of the activities to answer the research question. To discover the process model, we conduct a process mining project that starts with the *log preparation* step and concludes in the *process model validation* step. In the *log preparation* step, we conduct pre-processing steps to cleanse the log from orders which have been incompletely processed such that it is ready to be used for the next step. Next, we conduct an experiment to discover the process model which uses the process mining algorithms. This step yields three variants of process model as depicted in the Figure 3 as *process model unit A, process model unit B, and process model unit C.* In order to ensure that our process model already depicts the real life situation, we involve the process expert to validate the process model. Moreover, we also use the simulation functionality in PETRA to validate the time behaviour of the process model. Next, we analyse and compare the process models to discover the variation point of the process models. The variation point of the process model is used to determine the configuration point such that we can combine them and develop the configurable process models. Finally, PETRA uses the configurable process models to find the better process model.

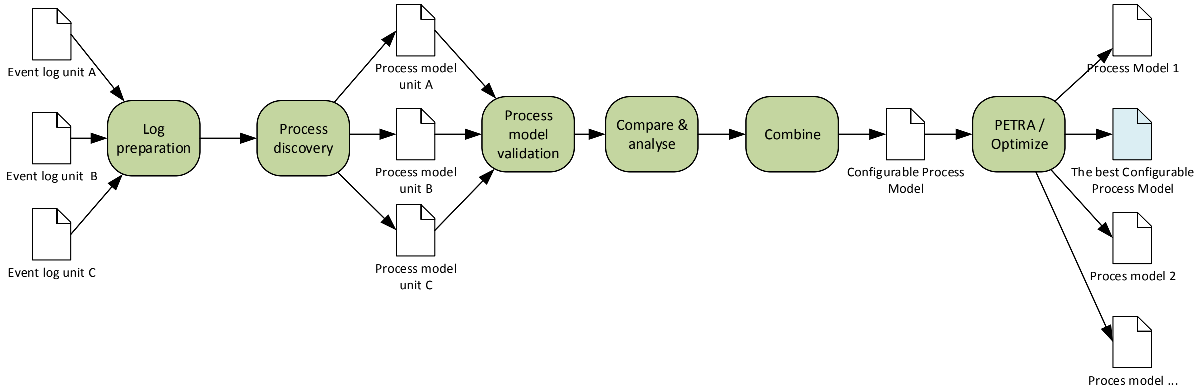


Figure 3 the methodology

1.5. Thesis Outline

The thesis is divided into seven chapters. In Chapter 2, we present a preliminary study about tools and technique that we use in this study. In Chapter 3, we present the result of the steps related to discovering and validating the process model. In Chapter 4, we present the result of the analysis and comparison of the process models. In Chapter 5, we discuss the development process of the configurable process models and the optimization results using PETRA. Chapter 6 presents the related works to this study. Finally, in Chapter 7, a summary is given. We also include a list of future works that can improve the results of this thesis.

Chapter 2. Preliminaries

In this chapter, we provide a general explanation for several concepts that we use in this study. This chapter starts with an explanation of the metrics to measure the quality of control flow and the concept of the configurable process models. Next, we present several process mining algorithms we use for the process discovery activity. Then, we explain about business process redesign and PETRA for analysing and optimizing task.

2.1. Measuring the process

Discovering the control flow of a process is a delicate task since one log can be transformed into many control flows. Therefore, we need a metric to determine the quality of the control flow. The control flow can be measured by four different quality dimensions, namely:

- Replay fitness
  Replay fitness is used to quantify how many traces can be replayed using the discovered control flow. A replay process is an aligning process between the actual activities in the log and what can be mimicked in the control flow. However, it may be the case that some of the activities in the log cannot be mimicked in the control flow, or vice versa [7]. The term 'move on log' is used when the activities in the log cannot be aligned in the control flow. Meanwhile, the 'move on model' is used when the activity in the control flow that should be executed does not have the counterpart activity in the log. In both cases, the replay fitness of the control flow has a lower value since the control flow cannot precisely mimic the behaviour of the process in the log.
- Simplicity
  In many cases, the discovered control flow is very hard to read. Such model is usually called a spaghetti-like process. Hence, the simplicity dimension is introduced to measure the complexity level of the discovered control flow.
- Precision
  Fitness and simplicity alone are not sufficient [9]. We can make a control flow that is simple and has a perfect fitness value that we called a flower model [6], as can be seen in Figure 4. Based on the first two dimensions, this model is an acceptable model. However, this model is not very useful because it cannot depict the precise behaviour of the process. Therefore, the precision dimension is introduced to measure the ratio of the behaviour allowed in the model and the actual behaviour in the log. If all the behaviours allowed by the model are actually observed in the log, then the control flow has a perfect precision value.
- Generalization
  Finally, the generalization dimension is used to measure the ability of the control flow to cope with the future behaviour of the process. A very precise model usually cannot handle the next case that is unseen the in log.

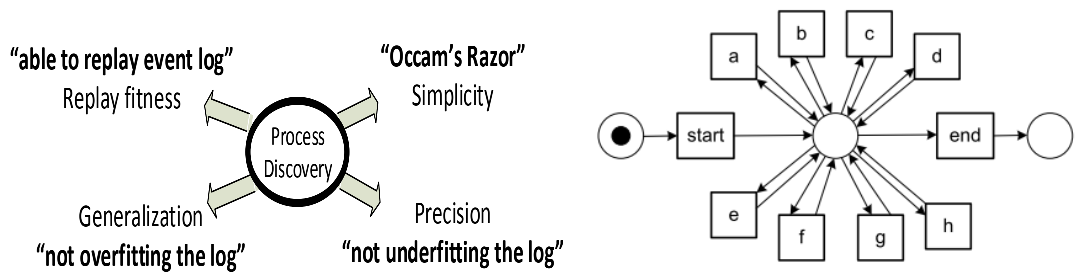


Figure 4 (left) the quality dimension of control flow, (right) the flower model

Ideally, a good process model has high scores for all of the dimensions. However, it is very difficult, or nearly impossible, to achieve in real life situations since a higher value of one dimension might lower the value of another dimension. For example, the flower model in Figure 4 that has a high value of the replay fitness and the simplicity on one hand and has a low value of the precision on the other hand. If we want to increase the precision value of the flower model, it can only be achieved by having the control flow with lower simplicity value. Nevertheless, the replay fitness and the precision are more important than the other dimension since these dimensions focus on the quality of the control flow to capture behaviour seen on the log while the other dimensions focus on the structure of the control flow (simplicity) and the future aspect of the control flow (generalization). Moreover, we can get the sense of the generalization value by looking at the precision value. Therefore, for practical reasons, we only consider the replay fitness and the precision as the quality measurement of the control flow. Furthermore, in order to measure fitness and precision, we use the work from [7].

2.2. Configurable process models and Process Trees

Configurable process models are a compact representation of a family of process models [1]. A family of process models are similar process models with a different set of process model elements. A selection of process model elements is called a configuration. Many configurable process models have been defined over the years such as configurable EPC [20,21], configurable YAWL [22], and CoSeNets [23]. PETRA picked CoSeNets as foundation and extended CoSeNets into a new model called Process Trees as its model representation [24]. Process Trees are a block-structured process model with one single entry and exit point. The block-structured model is a model that can be divided recursively into parts having single entry and exit point. Figure 5 shows an example of a block-structured model.

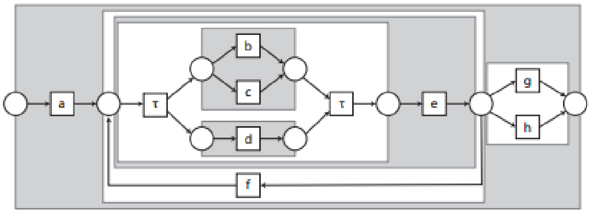


Figure 5 example of block-structured work flow net taken from [4]

In Process Trees, there are five different perspectives that are explained in the following sections.

2.2.1. Control flow perspective

The control flow depicts the path of business process and it is denoted with edges and nodes. Furthermore, a node can be a block or a task. A task is an actual activity that is executed in the system and it could be either automatic or manual task and a block represents the relation between its child nodes. There are ten types of block: AND, SEQ, OR, XOR, LOOPXOR, LOOPDEF, DEF CHOICE, TIMEOUT EVENT, MESSAGE EVENT and PLACEHOLDER. Every node can be configured with two options:

(i) Hide: replace the node with an automatic task and it is denoted with curved arrow

(ii) Block: delete entirely the sequence where the node belongs to from control flow and it is denoted with no-entry sign.

Besides the two options, we also can have another configuration called *substitution.* A substitution is the option to replace a part of control flow with a predefined sub process. The middle figure in Figure 6 shows the example of a simple process trees with configurable points. It has two configurable points in node B and node C where node B can be hidden while node C can be blocked. This Process Trees can be derived into three different control flows, as can be seen in Figure 6.

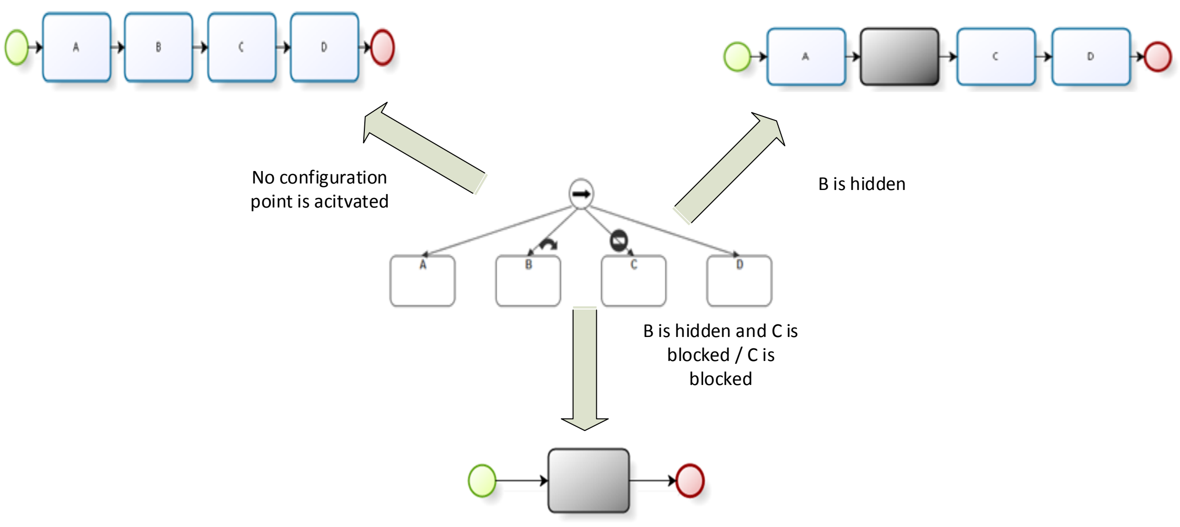


Figure 6 three control flows derived from one process tree with configurable points

The black box is the automatic task. The automatic task does not need neither processing time nor resource. The top left picture on the Figure 6 presents the control flow with no configuration points is taken. The top right picture shows the control flow where the activity B is hidden. The bottom picture depicts the control flow where the activity B is hidden and C is blocked or where the activity C is blocked. The bottom figure only shows one automatic task since the activity C is blocked. This blocking makes all activities in the sequence that contains activity C are omitted.

2.2.2. Resource perspective

The resource perspective handles the resource definition, the resource work schedule, and the resource selection for every task. The configuration point for this perspective is which resource work schedule can be kept or removed and the task responsibility of the resource. Table 1 shows the example of the configurable work schedule of resource R1. R1 has three different work schedule due to the shift regulation in his company. He can be available on Monday to Saturday from 08.00 – 16.00, on Monday – Saturday from 18.00 – 06.00 the next day, or not available at all during the week. The resource selection for every task depends on the control flow perspective. For example, if R1 has the activity B and C in Figure 6 as his task responsibility, R1 will only responsible for the activity C when the activity B is hidden.

Table 1 example of resource work schedule

Res	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
R1	06 - 18	06 - 18	06 - 18	06 - 18	06 - 18	06 - 18	
	18 - 06	18 - 06	18 - 06	18 - 06	18 - 06	18 - 06	

2.2.3. Data perspective

Data perspective deals with expressions and variables. A variable is a container to store the data that is used along the process. It can be written by a task and the value to be written is determined with certain probability. Furthermore, an expression is a guard of the outgoing edge such that the process can only go through that outgoing edge when its expression value evaluates to true. The expression consists of variables, its specific value, and operator. XOR, LOOPXOR, and OR-block reads specific variables to pick one of the possible outgoing edges. The configuration point for this perspective determines which expression can be kept/removed, which variable value can be kept/removed, which variable in one node can be read/ not read, and which variable in one task can be written / not written. Figure 7 shows an example data perspective in the Process Trees and the usage in the control flow. Since it has one configuration point, the data perspective can be configured into two possibilities. The XOR-choice reads the variable *xor1* to determine which outgoing edges the process should pick. The initial value of the variable *xor1* used in the control flow in Figure 7 is -1. At first, it has an equal probability for the XOR-choice to choose between the activities B, C, and D, i.e. 0.33. When the activity B is blocked, the activities C and D still has an equal probability but with a higher probability since no activity B anymore, as depicted in the bottom right picture in Figure 7.

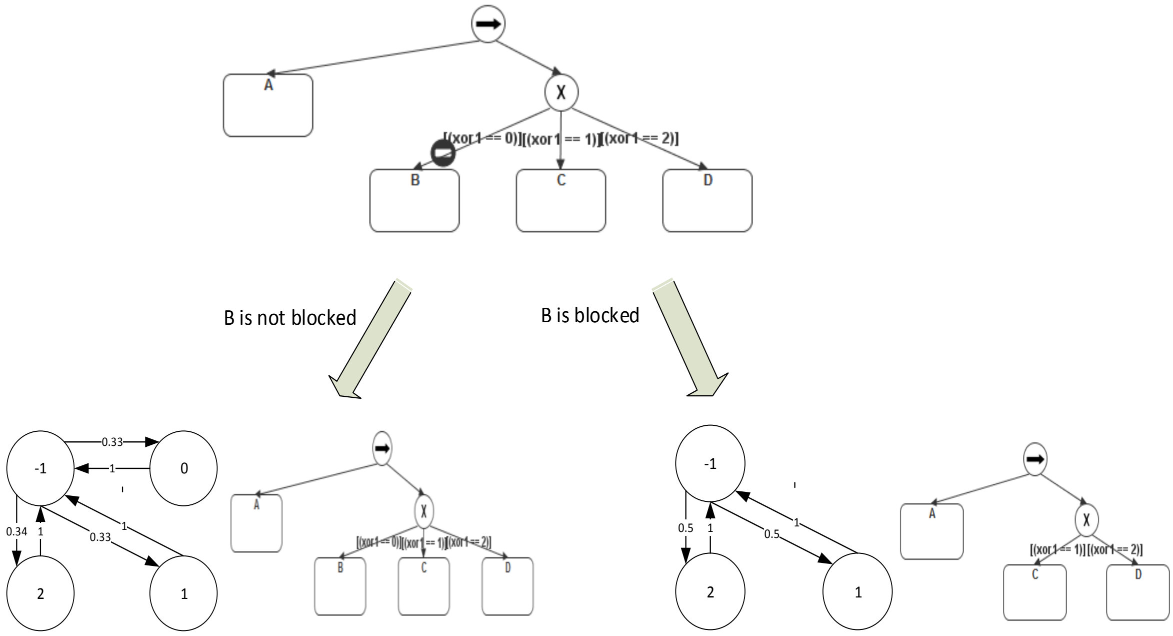


Figure 7 example of process tree with data perspective

2.2.4. Environment perspective

Currently, the environment perspective only has one property, i.e. the arrival process. The arrival process specifies the distribution of the arrival of new cases in the process. Configuring this perspective means we choose the suitable distribution describing the arrival the case.

2.2.5. Experiment perspective

The experiment perspective consists of the simulation properties, e.g. the replication length, the warm up period, and the number of replications. Since this perspective is not related to the control flow, all possible configurations of other perspectives use the same configuration of the experiment perspective.

2.3. Control Flow Discovery

This section presents the techniques or algorithms we choose to discover the control flow. The idea of control-flow discovery is to compose automatically a process model that can mimic the behaviour seen in the log. Various algorithms have been developed such as the fuzzy miner [18], the heuristics miner [17], the α algorithm [16], the inductive miner [4, 5], and the ETM miner [3]. The last two algorithms have been chosen as the algorithm to mine the process in this study because they can guarantee the soundness of the control flow while the other algorithms cannot [28, 4]. In most of the cases, the unsound control flow can be immediately discarded without considering the log. A control flow is sound if and only if all the process activities can be executed and some satisfactory end state can always be reached [28].

2.3.1. Evolutionary tree miner (ETM) [3]

ETM is a genetic algorithm that is already extensively used in other fields. In general, the approach of ETM can be seen in Figure 8. It starts with generating random control flow as initial input (1) and after that it computes the fitness to find the best control flows (2). For every generation, we only consider the best control flows to be carried out to the next generation (3a). Next, the rest of control flows will be crossed over or mutated to create a new input (3b) and the process will be iterated (4). This process can be terminated by giving a maximum duration or when it already achieved the desired fitness value that was given (5).

ETM is chosen because it can discover a control flow that optimizes all four quality dimensions [3]. Moreover, the importance level of dimensions also can be set during the mining process. However, since this algorithm is based on genetic algorithm, it has a longer run time compared to the inductive miner.

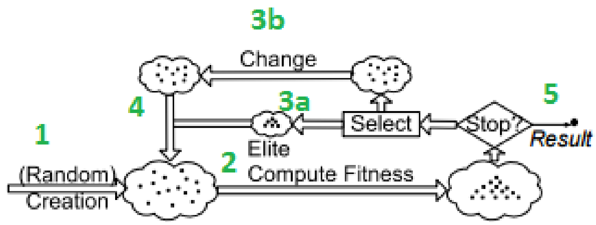


Figure 8 ETM algorithms taken from [3]

2.3.2. Inductive miner-infrequent [4, 5]

The inductive miner-infrequent (IMi) is an extension technique of the inductive miner [5]. The inductive miner uses a divide-and-conquer approach to mine the model. This technique tries to divide the log into several small logs that represent different blocks in the block-structured model. As an extension of the inductive miner techniques, IMi uses the Pareto principle to eliminate the infrequent behaviour in the log for every block. The Pareto principle states that, for many cases, only 20 % of the cause that are considered important because they produce 80 % of the effect. This extension results in a control flow with better results compared to the inductive miner [5]. However, this technique has the drawback that it usually produces a control flow with a lower precision value compared to the ETM.

2.3.3. Silent transition

The discovering process sometimes produces the silent transitions in the control flow. The silent transition is an activity that is automatically executed with no processing time and resource. The process mining algorithms use the silent transition to model the 'other' option of an XOR-choice or OR-choice. The 'other' option can cover the other activities that actually are being executed in a case but which are not depicted in the control flow. These activities are not modelled in the control flow because their occurrence is less significant compared to other activities. Another use of silent transition is when a case does not choose any available activities in XOR-choice or OR-choice such that it needs an escape way from this XOR-choice or OR-choice. Figure 9 depicts the control flow with silent transition. The black box is the silent transition. Since we have the silent transition, it is possible to have the case that directly executes activity D after activity A, escaping the activity B and C.

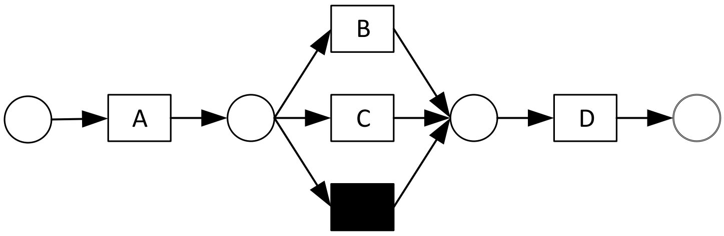


Figure 9 the control flow with silent transition

2.4. Business Process Redesign (BPR)

The need for an organization to remain competitive drives them to look at the current business process and try to optimize the current one. One way to optimize the current one is by redesigning the current business process. In [10], some heuristic rules are introduced to guide the practitioners when redesigning the current business process. These heuristic rules have been developed by evaluating many previous works and gaining experiences from large companies. In general, there are seven focuses to start the BPR: customer, business process operation, business process behaviour, organization, information, technology, and external environment. Since our study is based on business process and its properties as input, we only use two out of seven focuses, which are:

a. Business process operation
   There are five heuristic rules within this focus:
   (i) Order types: separate different types of order into different business processes
   (ii) Task elimination: omit all unnecessary tasks which do not add value to the customer
   (iii) Order-based work: consider removing batch processes
   (iv) Triage: try to divide the general task into two or more alternative tasks, or vice versa.
   (v) Task composition: try to combine several tasks into one general task, or vice versa.
b. Business process behaviour
   There are four heuristic rules within this focus:
   (i) Resequencing: relocate task to a more appropriate position
   (ii) Knock-out: *order knocks out in a decreasing order of effort and in an increasing order of termination probability [10].*
   (iii) Parallelism: try to parallel the process when possible.
   (iv) Exception: differentiate the exceptional flow into separate business process.

Some of these heuristic rules are used when we determine a configuration point in our configurable process model.

2.5. PETRA

PETRA stands for *Process models based Extensible Toolset for Redesign and Analysis [24].* It is a framework to explore a space of possible process models by repeatedly analysing elements of the space and supporting the collection and comparison of analysis result [1]. The space of possible process models is generated from the configurable process model. Afterwards, PETRA analyses the possible process models in order to get the better model. Figure 10 gives the architecture of PETRA.

First, PETRA constructs a specific Process Tree from the configurable process model. This Process Tree is passed to a tool through interface A to be analysed. Currently, PETRA provides several different tools, e.g. CPN-tools and L-SIM to analyse the Process Tree, and can be expanded for further needs. Next, the tool gives the result to PETRA through interface B. After getting the result, PETRA adds the Process Tree with the analysis result as a property of the Process Tree, such as: throughput time, resource idle time, resource working time, and task processing time. The Process Trees' properties in PETRA are inspired and developed according to the *WfMC* standard of *BPSim [25].* As the final result, PETRA yields a set of Process Trees that have been quantitatively analysed. Given the result, the user can decide which process model suits the business needs and can be applied to her business.

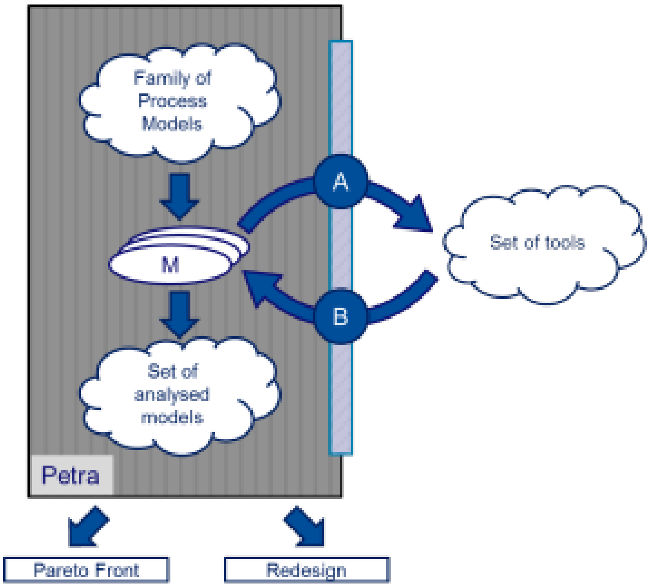


Figure 10 Architecture of PETRA taken from [24]

Chapter 3. Process model discovery

In order to find the variants of and to optimize the process model, we have to discover the process model itself first and ensure that the process model closely represents the real life situation. The process models can be seen in many perspectives as explained in Chapter 2, such as control flow perspective, resource perspective, data perspective, environment perspective, and experiment perspective. This chapter presents the result of process model discovery from every perspective and the validation process of the process model. This chapter starts with the detailed explanation of the OTC process in DSM which is the object of this study. Subsequently, the explanation of the event logs that we collect as the main input for the process discovery is given. Afterwards, we present the result of the control flow discovery process using two process mining algorithms that have been chosen in Chapter 2. Next, the discovery process of the resource perspective is explained and it is followed by the explanation of the discovery process of the data perspective, the environment perspective, and the experiment perspective. Finally, the validation process results are presented both from the process expert perspective and the time behaviour of the process perspective.

3.1. Order to Cash process

DSM as a global science-based company has a wide range of product variety from health supplement to materials. One of the strategies to handle these variances is to form a business cluster representing the coherent product/market combinations. DSM has four clusters to handle the whole range of their products: Nutrition, Material Science, Pharma, and Polymer Intermediates, and every cluster has one or more business units. Every business unit has many business processes to support their daily routines. Nevertheless, one of the business processes that is essential to all business units is Order to Cash (OTC) process. The OTC process is a family of business processes related to goods or service that DSM sells. In our study, we analyse three variants of the OTC process from three different business units, so called BU1, BU2, and BU3.

In general, the OTC process can be divided into two groups based on the relation to the order: main process and supporting processes. Moreover, the main process can be divided into five sub processes as depicted in Figure 2: order intake, delivery, distribute, bill, and cash in. The main process is similar for every business unit and it relies heavily on the people executing it.

In the *Order Intake* sub process, the customer order is manually put into the system by the sales employee. Afterwards, several checks, including the credit check, the stock availability check, and the price check are done automatically by the system. If one of these checks gives a negative result, the system will change the status of order into blocked and only the person in charge can release this blocked order. In a normal case, an order that has passed through all the checking will be directly go to the *Delivery* sub process. Afterwards, the order is packed and shipped in the *Distribute* sub process and the invoice will be sent to customer in the *Bill* sub process. The main process is finished when DSM has already received the payment from the customer in the *Cash In* sub process. Along the process from the *Order Intake* sub process to the *Distribute* sub process, the customer can make some changes such as the required delivery date change and the quantity change. These changes from the customer make the OTC process more complex and has longer throughput time. One of the characteristics of the OTC process is more than 40 % of the orders are subject to change or corrections.

The supporting processes are the processes that indirectly related to sell the goods or service, such as:

- Master Data Management: All administrative tasks of data used in OTC. It includes material master, vendor master, and customer master.
- Pricing: This process is to set a target price, sales price and margin setting.

The supporting process can reduce the manual entry and order interruptions if done properly. Nevertheless, in our study, we only consider the main process of the OTC process.

Furthermore, the main process has three types, which are:

a) Standard order

The standard order is the very generic type of OTC process. DSM directly sells the goods to the customer in this type and the goods are owned by DSM. Figure 11 depicts the flow of this process. In this process flow, the *delivery* sub process is divided into two processes: *deliver* and *transportation plan.* Furthermore, the *distribute* sub process is divided into four processes: *check in, picking, load,* and *check out.* The *bill* sub process is also divided into several processes: *ship costing* and *billing.* Last, the *cash in* sub processes is divided into two processes: *cash in* and *ship cost settlement.*

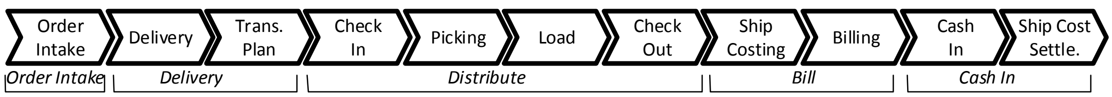


Figure 11 standard order process flow

b) Third party order

Third party order (TPO) is a business process when DSM wants to re-sell goods that DSM bought from a third party. The third party directly delivers the goods to the DSM customer. Figure 12 shows the process flow of TPO. In this variant, the *deliver* and *distribute* sub process do not exist and as substitution, this variant has several activities to buy the product from a third party in the middle of process flow (the processes with grey background), which are *purchase request, purchase order, goods receipt,* and *invoice verification.*

Figure 12 third party order process flow

c) Service and Project order

Besides goods, DSM also sells services such as consulting and maintenance. Due to the different type of product, the activities in the process flow are completely different from goods process flow. Figure 13 shows the process flow of the project order. It starts with the creation of project structure and the plan of the project costs. Next, the project order is created in the system and the project can be executed afterwards. Depends on the agreement, in the middle of the project works or at the end of the project works, the actual costs are calculated and the bill is sent to the customer. The project can be settled and closed after the result has been analysed and the report is sent to the customer. Nevertheless, the system only records the *order intake* sub process, the *billing* sub process, and the *cash in* sub process since the other activities are not related to the sales system.

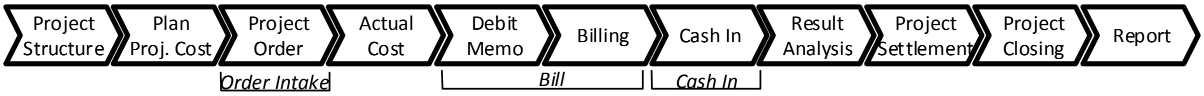


Figure 13 project order process flow

In our study, we consider all three types of the OTC process. Yet, the standard order, as the generic type of OTC process, is the only type that is shared by all business units. Therefore, in our study, we focus more on analysing the activities related to the standard order.

3.2. Event Log

The event log is the main input of process mining activities since most of the activities in process mining need the event log. The event logs for our study are generated from ARIS PPM and contains redo logs from different tables in the SAP system. In this study, we have 23 different log files for every business unit in XML format. Figure 14 shows a snapshot of one of the log files. The log files contain the incomplete log. The incomplete log is the order that is blocked or cancelled due to the credit issue or manual cancelation from the customer. Besides that, the log file contains many fields which are not too important for discovering the control flow. Therefore, the pre-processing steps are needed to combine all of these files into one file event log, eliminate unnecessary fields, and filter out the incomplete log. The filtered log is the log that contains only traces of the complete order of the OTC process. A trace in the filtered log is a representation of one order and consists of several fields: trace id, process id, process name, process datetime, username, filename, and process predecessor id. Since there is only one field about time for every activity, we assume that the time stored in the filtered log is the completion time of activity. Furthermore, the characteristics of the filtered log for every business unit can be found in Table 2.


```xml
<event>
  <attribute type="VBAK-ERZET">193620</attribute>
  <attribute type="CDHDR-TCODE">VA02</attribute>
  <attribute type="CDPOS-TABKEY">3400000109467</attribute>
  <attribute type="CDHDR-UTIME">193133</attribute>
  <attribute type="CDPOS-OBJECTCLAS">VERKBELEG</attribute>
  <attribute type="VBAK-ERDAT">20150126</attribute>
  <attribute type="CDPOS-CHNGIND">U</attribute>
  <attribute type="FRAGMENT_CLASS">Corrective Activity</attribute>
  <attribute type="CDPOS-UNIT_NEW"></attribute>
  <attribute type="CDHDR-USERNAME">USER---</attribute>
  <attribute type="CDHDR-UDATE">20150206</attribute>
  <attribute type="DD04T-DDTEXT">Overall status of credit checks</attribute>
  <attribute type="CDPOS-CUKY_OLD"></attribute>
  <attribute type="CDPOS-FNAME">CMGST</attribute>
  <attribute type="CHANGED_FIELDNAME">Overall status of credit checks</attribute>
  <attribute type="CDPOS-CHANGENR">0008053324</attribute>
  <attribute type="CDPOS-UNIT_OLD"></attribute>
  <attribute type="CHANGED_FIELDNAME_CODE">CMGST</attribute>
  <attribute type="CDPOS-OBJECTID">0000109467</attribute>
  <attribute type="VBAK-VKORG">3400</attribute>
  <attribute type="VBUK-VBELN">0000109467</attribute>
  <attribute type="CDPOS-CUKY_NEW"></attribute>
  <attribute type="PROCESS_KEY_OWN">VBUK_HDR_CHNG-0000109467#-#0008053324#-#VBUK#-#CMGST#-#fip</attribute>
  <attribute type="PROCESS_KEY_PREDECESSOR">VBAK-0000109467#-#fip</attribute>
  <attribute type="FUNCTION_NAME">Set Overall status of credit checks on customer order</attribute>
  <attribute type="CDPOS-TABNAME">VBUK</attribute>
  <attribute type="DD03L-ROLLNAME">CMGST</attribute>
  <attribute type="CDPOS-VALUE_OLD">D</attribute>
  <attribute type="VBAK-VBTYP">C</attribute>
  <attribute type="CDPOS-VALUE_NEW">B</attribute>
</event>
```


Figure 14 Snapshot of log file

Table 2 Log general characteristics

BU	Log duration	Amount of traces	Avg. number of activity / trace	Std. deviation avg. number of activity / trace	Avg. throughput time	Std deviation avg. throughput time
BU1	June – Dec 2014	25,656	12.85	6.90	63.07 days	23.96 days
BU2	May 2014 – Feb 2015	10,872	12.93	5.96	52.80 days	19.31 days
BU3	July 2014 – Jan 2015	27,466	14.03	4.07	61.77 days	28.75 days

3.3. Control flow perspective

We use the two process mining techniques to discover the control flow perspective: ETM and IMi. At the first trial, the discovering process with the filtered log did not produce a promising result. The IMi produces a model with high complexity, as can be seen in Figure 15 while the ETM cannot produce a sufficient number of generations after running it for 3 days. Figure 15 is a sample of a control flow with a good replay fitness value but has a low precision value. The low precision value makes this control flow hard to read and to analyse. The reason of this poor result is that the log has too many different trace patterns. A trace pattern is a unique sequence of activities from the *order intake* sub process to the *cash in* sub process. Therefore, we conduct a trace pattern analysis for each business unit. Table 3 shows the overview result of trace pattern analysis. The analysis shows that every business unit indeed has many trace patterns. BU1 has 8,743 trace patterns, BU2 has 2,799 trace patterns, and BU3 has 10,455 trace patterns. The most frequent trace pattern in BU1 occurs 1,786 times, BU2 occurs 840 times, and BU2 occurs 1,590 times. This total occurrence is considerably low compared to the total amount of the trace patterns of each business unit. Moreover, most of the trace patterns occur only one time in the log in each business unit. BU1 has 7,118 trace patterns with the total occurrence of 1, BU2 has 2,006 trace patterns, and BU3 has 8,218 trace patterns. In order to decrease the variability of the log, we filter the log such that it only consists of up to the 150 most frequent trace patterns. The 150 most frequent trace patterns is chosen because the top 150 trace patterns occurs at least 10 times in the log. The occurrence of 10 is chosen to filter the infrequent trace pattern from the log. Moreover, to investigate the effect of the trace pattern filtering to the precision and replay fitness value, we also use the control flow with top 50 trace patterns and top 100 trace patterns as an input. The detail trace pattern analysis can be found in Appendix A.

Table 3 Overview of trace pattern analysis

BU	Number of Trace patterns	The highest occurrence of one trace patters	Total trace patterns with the highest occurrence	The lowest occurrence of one trace patterns	The total trace patterns with the lowest occurrence
BU1	8,743	1,786	1	1	7,188
BU2	2,799	840	1	1	2,066
BU3	10,455	1,590	1	1	8,218

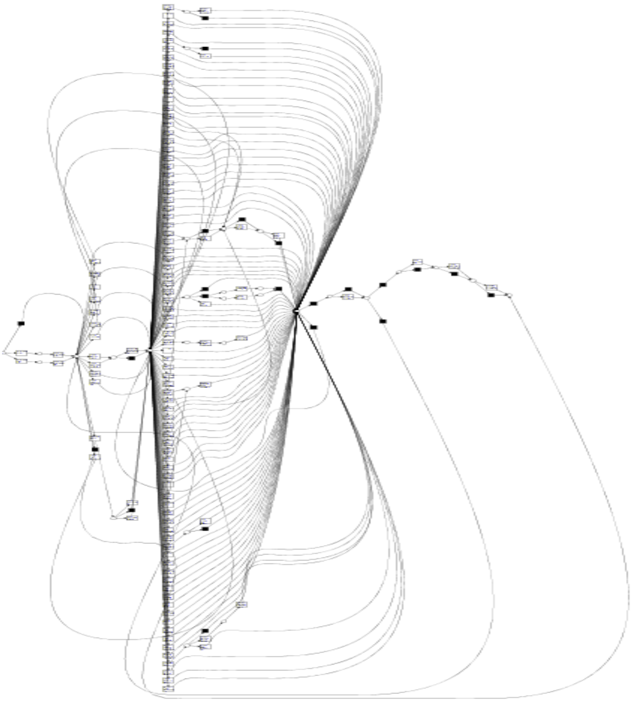


Figure 15 inductive model from complete log of BU3

3.3.1. Experiment setup

Having applied the trace pattern filtering in the filtered log, we conduct the experiment to discover the control flow. As mentioned earlier, every business unit has 3 different logs as input log:

- The log that contains traces in the top 50 trace patterns.
- The log that contains traces in the top 100 trace patterns.
- The log that contains traces in the top 150 trace patterns.

These logs are combined with 4 settings for the ETM and 1 setting for the IMi. The setting for the ETM varies the number of generations to stop the process. Meanwhile, the IMi setting applies the Pareto rules to eliminate the infrequent activities. Table 4 shows the complete settings.

Table 4 experiment setting for process discovery

Algorithm	Trace Pattern	Generations/method
ETM	100	100
ETM	100	200
ETM	100	500
ETM	100	1000
IMi	100	Infrequent (0.2)
ETM	150	100
ETM	150	200
ETM	150	500
ETM	150	1000
IMi	150	Infrequent (0.2)
ETM	50	100
ETM	50	200

Algorithm	Trace Pattern	Generations/method
ETM	50	500
ETM	50	1000
IMi	150	Infrequent (0.2)

3.3.2. General Results

After running the experiment, several conclusions can be derived:

1. In most cases, the IMi yields the control flow with more activities compared to the ETM.
2. In all business units, the IMi yields a model with higher replay fitness value compared to the model discovered by the ETM. However, the opposite situation happens in term of the precision value, where the ETM gives better result compared to the IMi.
3. For the ETM, we could not find any clear relation between the total number of trace patterns used and the precision value as some resulted control flows from the log contains top 150 trace pattern can have higher precision value compared to the control flow resulted from the log contains top 50 trace pattern.
4. For the IMi, in general, the replay fitness value increases as the number of trace patterns used goes up with one exception when the BU1's log of top 100 trace patterns yielded the control flow with higher fitness value compared to the log of top 150 trace pattern.
5. In general, we can discover the control flow similar to Figure 2 as can be seen in Figure 16:
    a. The *order intake* sub process
    From the 'create standard order' activity to the 'set status OTC allowed' activity
    b. The *delivery* sub process
    From the 'create outbound delivery' activity to one of the shipment activities such as the 'rail' activity, the 'road' activity, and the 'ex-work' activity.
    c. The *distribute* sub process
    From the 'Transfer order' activity to the 'Post good issue' activity
    d. The *bill* sub process
    From the 'create invoice' activity to the 'create accounting process' activity
    e. The *cash in* sub process
    The 'customer payment' activity.

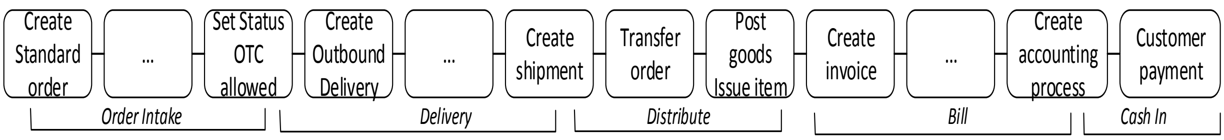


Figure 16 high level abstraction of the discovered control flow

The aim of process discovering is to get a control flow with both high precision and replay fitness value. However, since the input log does not represent all the trace patterns, we are very cautious in analysing the replay fitness and precision value. For example, in the experiment result, there is a model with the perfect score of precision that seems very unlikely to be true considering the input event log. Therefore, we choose to ignore this kind of result since it is highly possible that the model with the perfect score of precision is an over fitting model.

To improve the control flow quality, we try to compare and combine the model resulted from the IMi and the model resulted from the ETM. Since the IMi can produce the control flow with a higher replay fitness value compared to the ETM, we choose the control flow with the highest replay fitness value from the IMi as the candidate. On the other hand, we choose the control flow with the highest precision value from the ETM as the candidate. Using these two candidate control flows, we try to combine them with the aim to develop the model that has high the replay fitness value while also maintaining the good precision value. In order to do that, we propose two guidelines to combine the model:

a. From the control flow of the ETM, we try to adapt the activity sequence since a clear activity sequence increases the precision value.
b. From the control flow of the IMi, we try to adapt the XOR-choice pattern since it can increase the replay fitness value.

3.3.3. BU1 control flow combination

For BU1, we get the candidate control flow for the ETM when we use the top 50 trace patterns and 200 generations and we get the candidate control flow for the IMi when we use the top 100 traces. Table 5 shows the quality of two candidate control flows to combine and Figure 17 and 18 shows the control flow of the ETM and the IMi respectively. To simplify the combination process, we chunk the control flow according to the sub process in Figure 2 and combine the sub process.

Table 5 BU1 candidate models

PM	Algorithm	Trace pattern	Generations	Fitness	Precision
BU1	ETM	50	200	0.835	0.868
	IMi	100	Infrequent	0.898	0.617

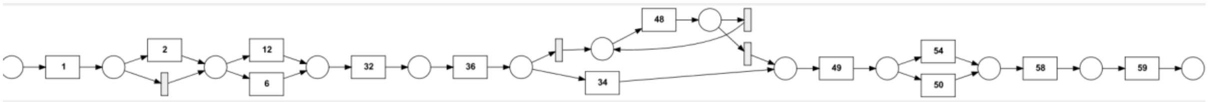


Figure 17 ETM candidate control flow for BU1

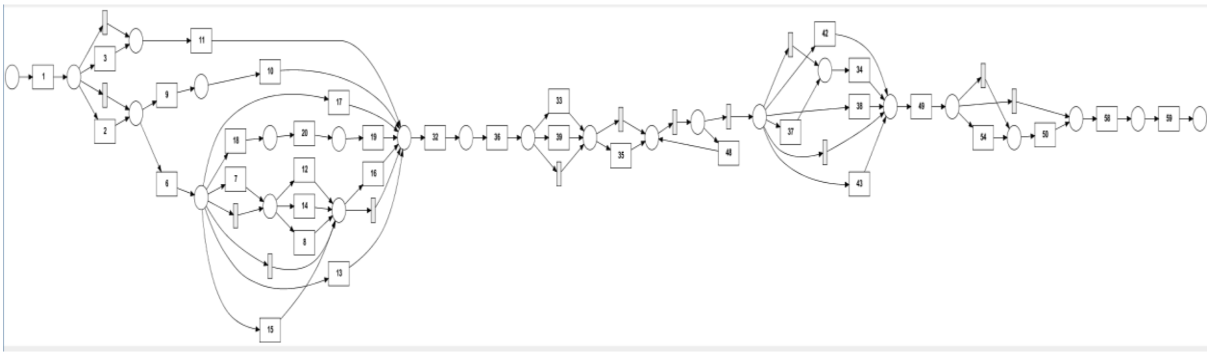


Figure 18 IMi candidate control flow for BU1

Order intake sub process.

Figure 19 shows the control flow from the IMi and Figure 20 shows the control flow from the ETM. The legend for the activity name can be found in Appendix B. From these results, it is clear that the IMi gives better control flow for this sub process because it has more detailed process behaviour compared to the ETM due to the XOR-choice pattern. After activity nr.6, the control flow from the IMi has more than one option while the control flow from the ETM has only one option. Therefore, we use the control flow from the IMi in the combined control flow.

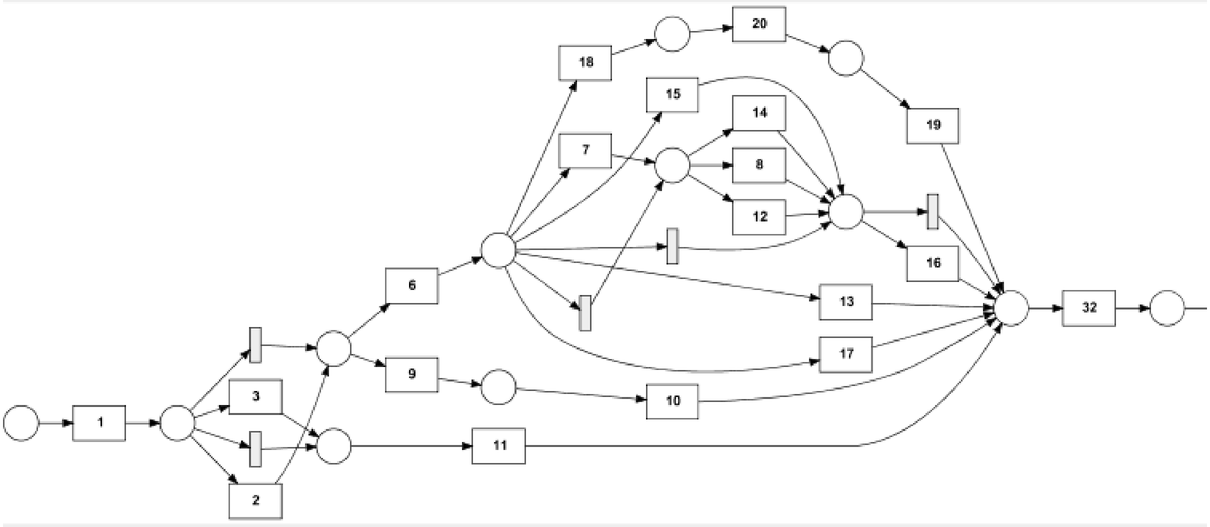


Figure 19 BU1 control flow from IMi in the order intake sub process

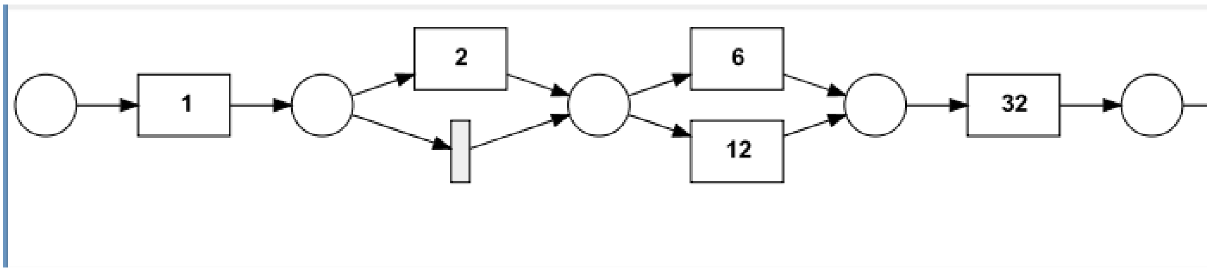


Figure 20 BU1 control flow from ETM in the order intake sub process

Delivery and Distribution sub process

Figures 21 and 22 show the control flow for the *delivery and distribution* sub process resulted from the IMi and the ETM respectively. One difference between these two results is the activities between the activity nr.34 (the activity with green box) and the activity nr.36 (the activity with a red box). In the control flow from the IMi, there are several intermediate activities between them while the control flow from the ETM does not have any intermediate activity. In the control flow from the IMi, we can discover several process orders from activity nr.36 to activity nr.34. One of the possible process orders from activity nr.36 to activity nr.34 is activity nr.36, activity nr. 39, activity nr.35, activity nr.48, activity nr.37, and activity nr.34. On the other hand, we only can discover one possible process order in the control flow from the ETM, i.e. activity nr.36, and activity nr.34. Therefore, we use the control flow from the IMi for these sub processes since it has more complete behaviour.

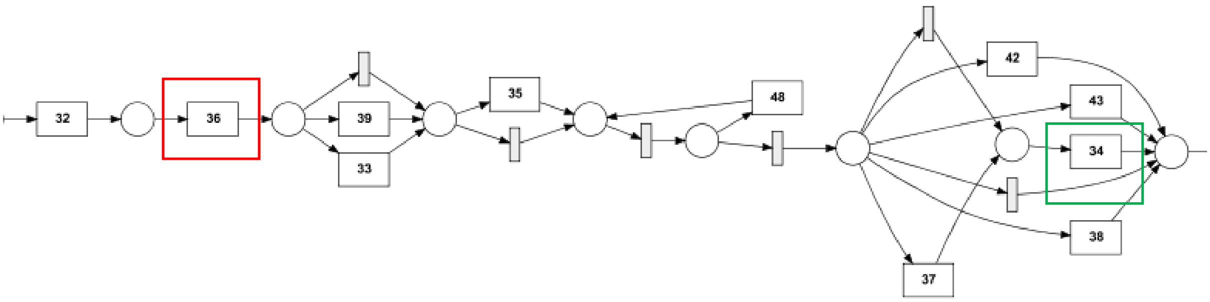


Figure 21 BU1control flow from IMi in the delivery and distribution sub process

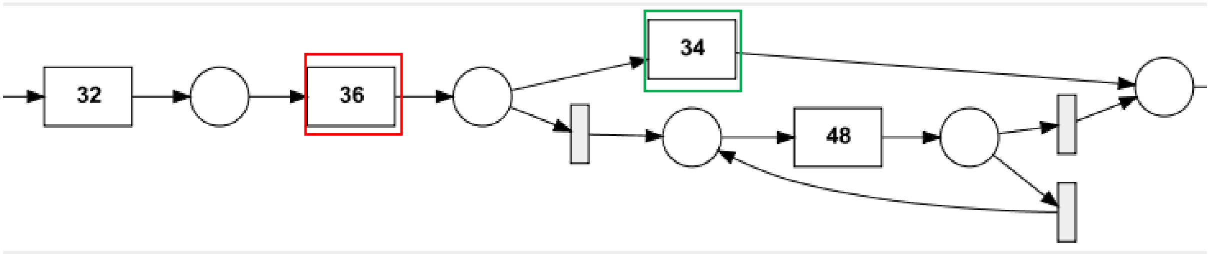


Figure 22 BU1control flow from ETM in the delivery and distribution sub process

Billing and Cash in sub process

Figures 23 and 24 show the control flow of the *billing and cash in* sub process resulted from the IMi and ETM respectively. One difference between these control flows is the activity sequence of the activity nr.54 and the activity nr.50. In the control flow from the IMi, the activity nr.54 and the activity nr.50 can be done in sequence. However, in the control flow from the ETM, the activity nr.50 and the activity nr.54 could not be executed in sequence. This behaviour makes the control flow from the ETM gets a high number of 'move on model' occurrence in the activity nr.54 such that we choose the model from the IMi in this sub process.

In conclusion, the control flow from the IMi is used completely for the BU1 control flow since it has more precise behaviour compared to the control flow from ETM. The complete result of the experiment can be seen in Appendix B.

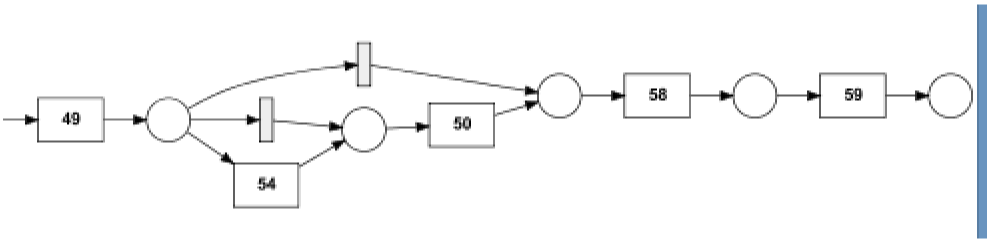


Figure 23 BU1 control flow from IMi in the billing and cash in sub process

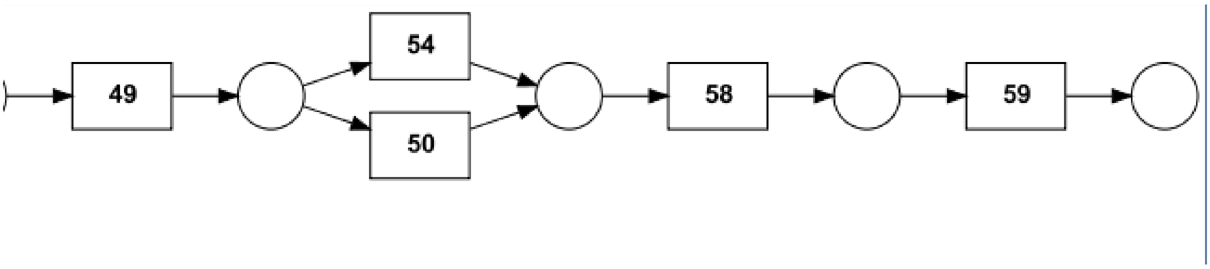


Figure 24 BU1 control flow from ETM in the billing and cash in sub process

3.3.4. BU2 control flow combination

For BU2, the candidate control flow from the ETM is acquired when using the top 150 trace patterns and 100 generations. Meanwhile, the candidate control flow from the IMi is acquired when using the top 150 trace patterns. Table 6 shows the quality of the candidates and Figure 25 and 26 show the control flow from the ETM and IMi respectively. To simplify the combination process, we chunk the control flow according to the sub process in Figure 2 and combine the sub process.

Table 6 BU2 model candidates

PM	Algo	TracePattern	Generations	Fitness	Precision
BU2	ETM	150	100	0.838	0.877
	IMi	150	infrequent	0.938	0.440

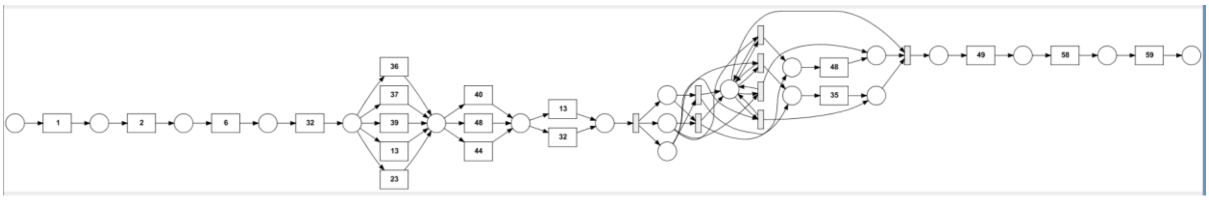


Figure 25 ETM candidate control flow for BU2

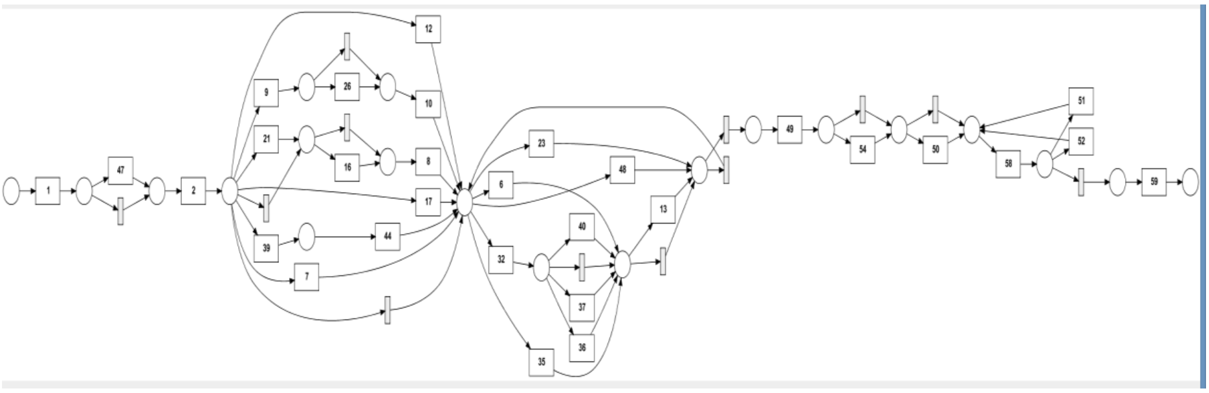


Figure 26 IMi candidate control flow for BU2

Order intake sub process

As we can see in Figure 27 and 28, the control flow from the IMi has more activities compared to the control flow from the ETM. Therefore, in our combination model we include all intermediate activities from the activity nr.2 (blue box in Figure 27) to the activity nr.6 (red box in Figure 27) from the control flow of the IMi. On the other hand, we also can see that the control flow from the ETM gives a clear order that the activity nr.6 is executed before the activity nr.32 (see red box in the Figure 28) while the control flow from the IMi has a loop that involves these two activities (see green box in Figure 27). The loop structure makes these activities can be executed in arbitrary order in the control flow of the IMi. Hence, we adapt the activity sequence from the control flow from the ETM such that the activity nr.6 is taken out from the loop (see red box in the Figure 29). Figure 29 shows the combined control flow of the *order intake* sub process.

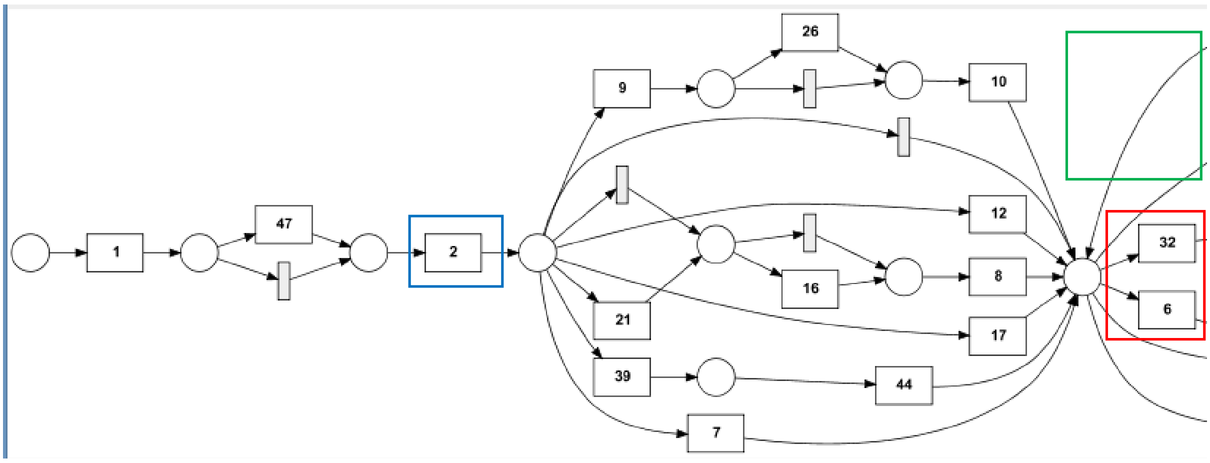


Figure 27 BU2 control flow from IMi in the order intake and delivery sub process

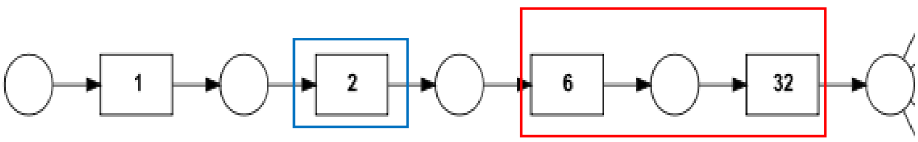


Figure 28 BU2 control flow from ETM in the order intake and delivery sub process

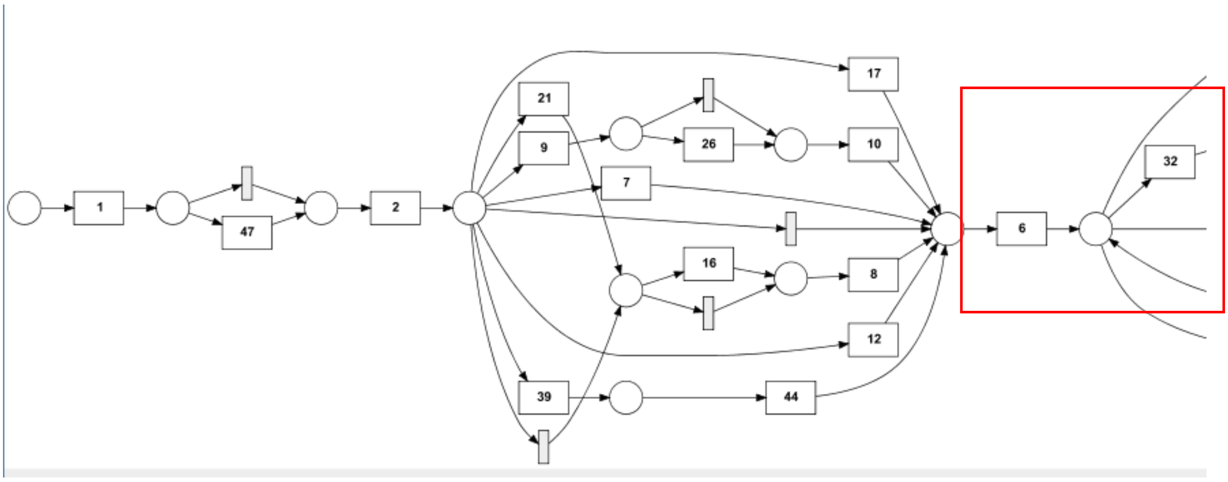


Figure 29 BU2 combined control flow in the order intake and delivery sub process

Delivery and distribution sub process

Figures 30 and 31 show the control flow of the *delivery and distribution* sub process resulted from the IMi and the ETM respectively. In the control flow from the ETM, we see the activity nr.32 appears in two different positions (see blue box in Figure 31) while the IMi models this activity with a loop. The loop structure in the control flow from the IMi can also handle the 'two different positions of one activity' situation as in the control flow from the ETM. Moreover, the loop structure can increase the replay fitness value as it can handle more behaviour of the process. Therefore, we choose the control flow from the IMi for the combined control flow.

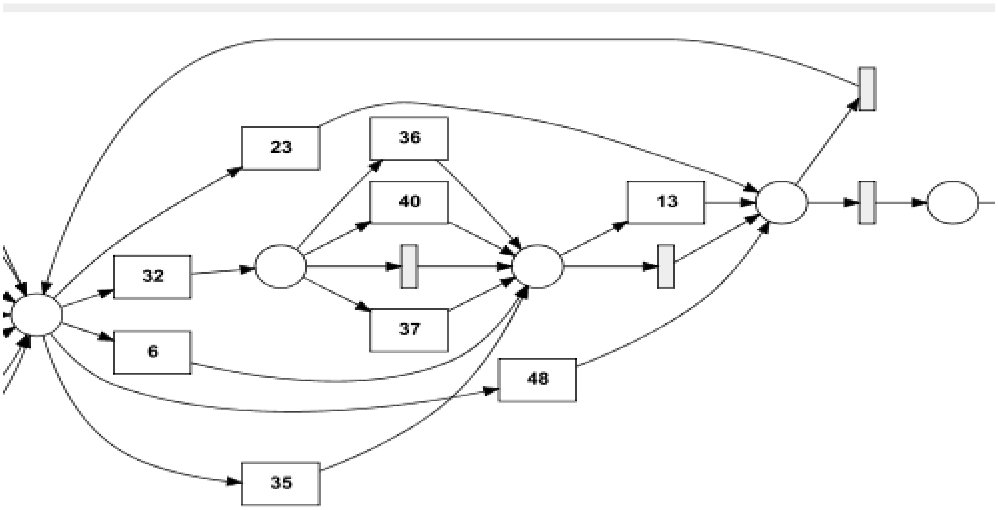


Figure 30 BU2 control flow from IMi in the delivery and distribution sub process

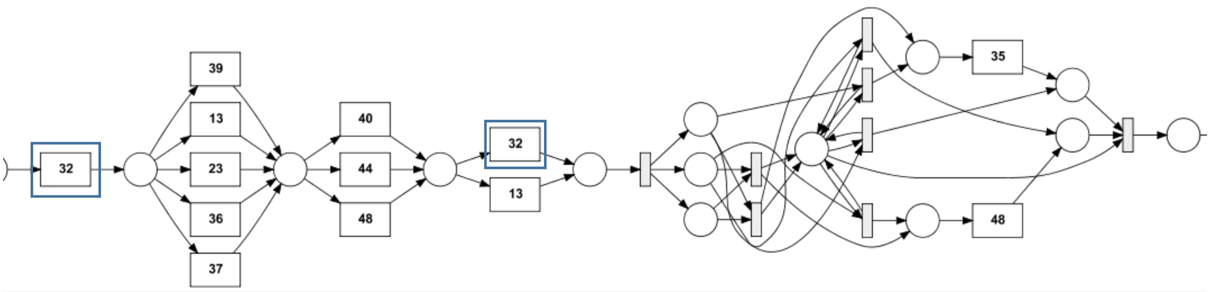


Figure 31 BU2 control flow from ETM in the delivery and distribution sub process

Billing and cash-in sub process

For these sub processes, the difference between the control flow from the ETM and the control flow from the IMi is on the number of activities. The control flow from the IMi has more activities compared to the control flow from the ETM such that it can handle more behaviour and increase the replay fitness value. Therefore, we use the control flow from the IMi for the combined control flow.

Figure 32 shows the combined control flow for BU2. After we compared and combined the control flow of ETM and inductive, we measure the quality of the combined control flow and compare the result with the candidate models in Table 6. The result of replay fitness and precision values of the combined control flow can be seen in Table 7 below.

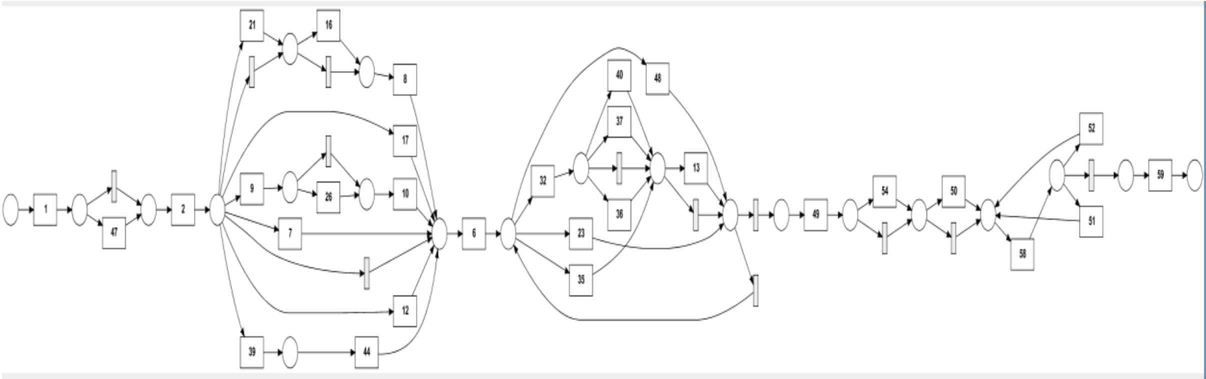


Figure 32 the combined control flow of BU2

Table 7 Comparison fitness and precision value of BU2 control flows

Control Flow	Replay Fitness	Precision
BU2-combined	0.935	0.544
BU2-IMi	0.938	0.440
BU2-ETM	0.838	0.877

The result shows that the replay fitness value of the combined control flow is nearly as high as the control flow from the IMi but with better precision value. The better precision value is achieved due to some activities sequences adaption from the control flow from the ETM. However, the precision value of the combined control flow is lower than the precision value of the control flow from the ETM. This due to the combined control flow has a loop structure that can significantly decrease the precision value. Learning from the result of the trace pattern analysis, we know that the OTC process has many variants. On the other hand, we also learn that the high precision value of the control flow from the ETM is achieved by omitting several activities such that the complete behaviour of the process and the variants of the OTC process could not be discovered on the control flow. It is shown by the lower replay fitness value of the control flow from the ETM compared to the combined control flow. Therefore, we picked the combined control flow.

3.3.5. BU3 control flow combination

For BU3, we find the candidate control flow from the ETM when we use the top 150 trace patterns and 100 generations. Meanwhile, we find the candidate control flow from the IMi when we use also the top 150 trace patterns. Table 8 shows the quality of the candidate control flows from the experiment result of BU3 and Figure 33 and 34 show the control flow from the ETM and the IMi respectively. To simplify the combination process, we chunk the control flow according to the sub process in Figure 2 and combine the sub process.

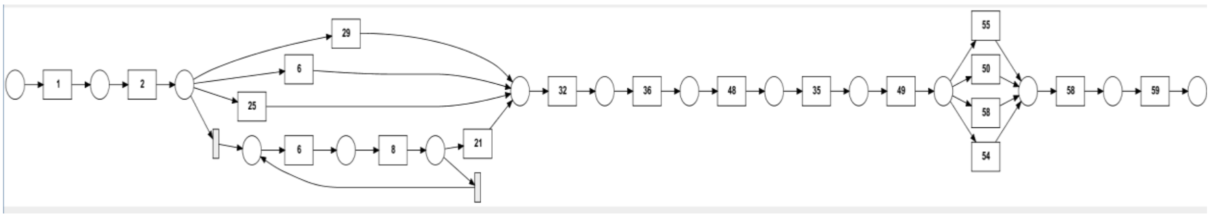


Figure 33 ETM candidate control flow for BU3

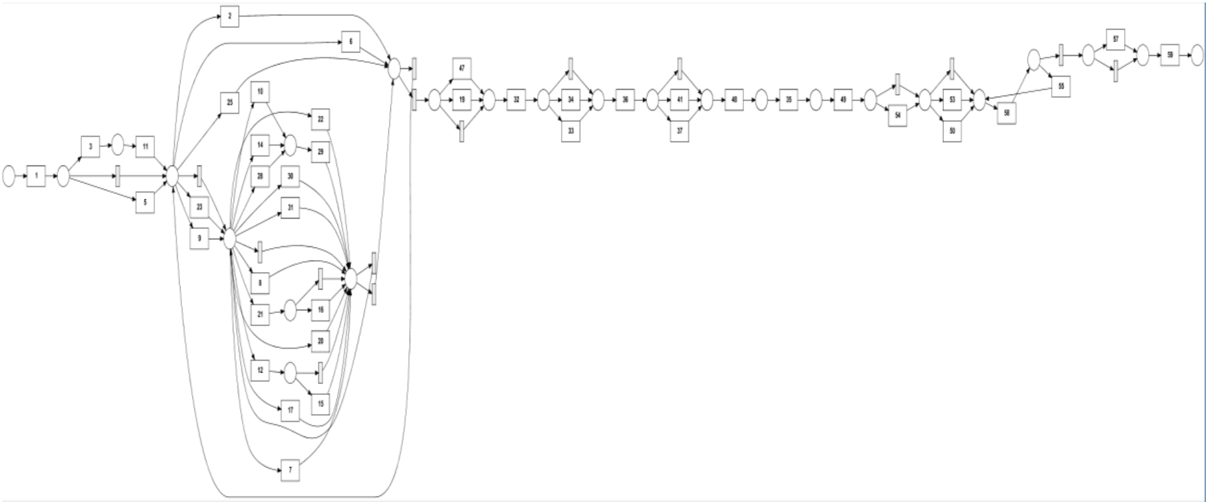


Figure 34 IMi candidate control flow for BU3

Table 8 BU3 model candidates

PM	Algo	TracePattern	Generations	Fitness	Precision
BU3	ETM	150	100	0.807	0.996
	IMi	150	infrequent	0.907	0.452

Order intake sub process

As we can see in Figures 35 and 36, the control flow from the IMi for the *order intake* sub process is much more complex compared to the control flow from the ETM. The control flow from the IMi has two level loops and the activity nr.2 and the activity nr.6 can be executed in arbitrary order (see red box in Figure 35). On the other hand, the control flow from the ETM does not have any loop structure and the activity nr.2 should be executed before the activity nr.6 (see green box in Figure 36). Thus, we adapt the activity sequence of the activity nr.2 and the activity nr.6 by taking out the activity nr.2 from outer loop. Besides that, we also adapt the set and release status pair activity from the control flow of the ETM and the trace pattern analysis such that every release activity can only be triggered after its set activity pair being executed. Moreover, based on exploration on the trace pattern, in most of the cases, the outer loop in the control flow from the IMi only exists to execute the activity nr.2 and the activity nr.6. Since we already discovered a clear sequence between these two activities in the control flow from the ETM, we remove the outer loop in the combined control flow such that the combined control flow has only one loop. Figure 37 shows the combined control flow of the *order intake* sub process.

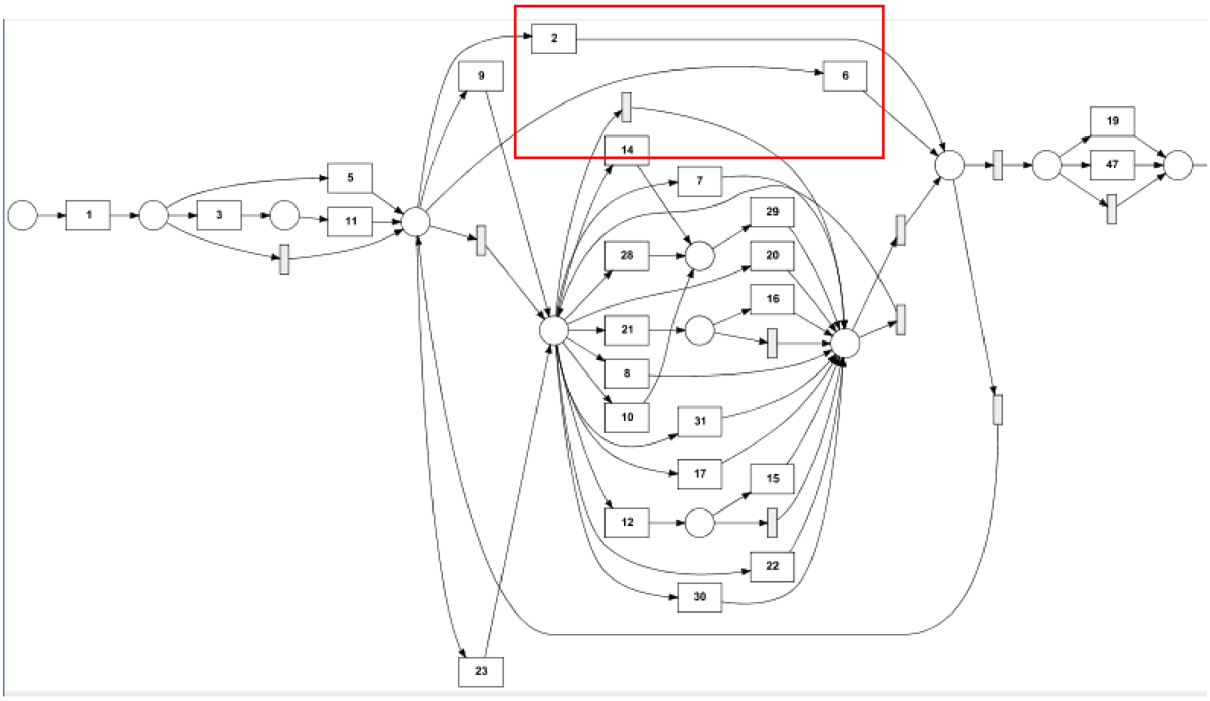


Figure 35 BU3 control flow from IMi in the order intake sub process

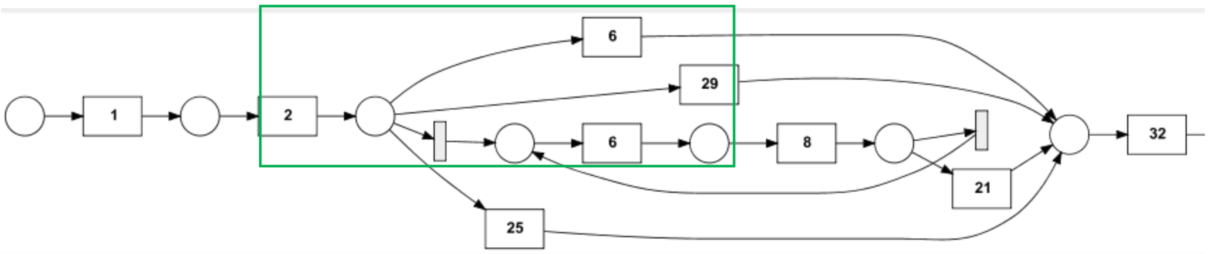


Figure 36 BU3 control flow from ETM in the order intake sub process

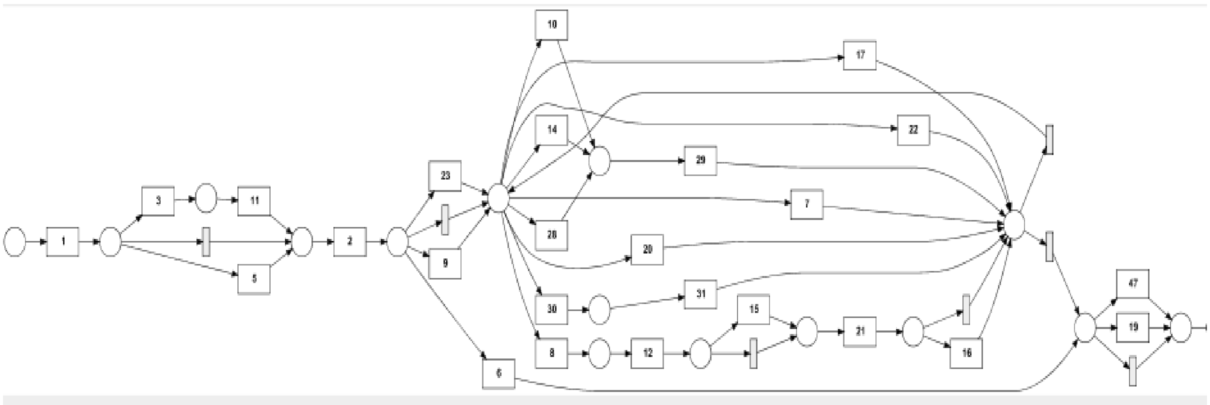


Figure 37 BU3 combined control flow in the order intake sub process

Delivery, distribution, billing, and cash in sub process

For these four sub processes, the difference between the two algorithms is the number of activities. We use the control flow from the IMi for the *delivery, distribution, and billing* sub process since it has more activities such that it can represent more behaviour. Meanwhile, there is no difference between the control flow from the ETM and the control flow from the IMi for the *cash in* sub process.

After we compared and combined the control flow from the ETM and IMi, we measure the quality of the combined control flow and compare the result with the candidate control flows. The result of replay fitness and precision values of the combined control flow can be seen in Table 9 and Figure 38 shows the combined model.

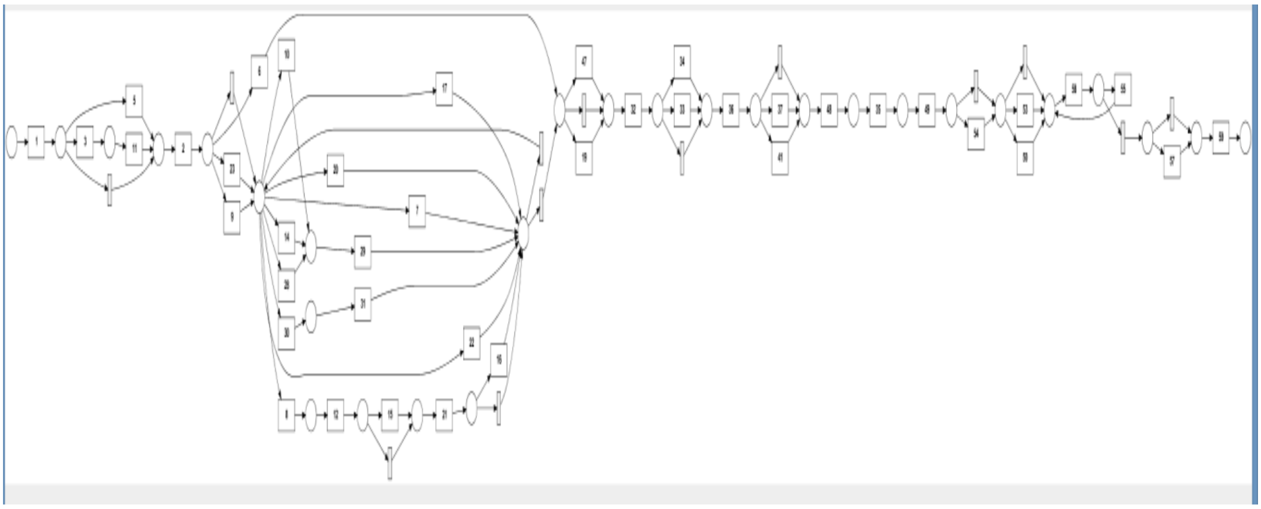


Figure 38 the combined control flow of BU3

Table 9 the comparison of fitness and precision value of BU3 control flows

	Fitness	Precision		Fitness	Precision		Fitness	Precision
BU3-combined	0,864	0,780	BU3 - ETM	0,807	0.996	BU3 - IMi	0,907	0,452

The replay fitness value of the combined control flow is slightly less than the control flow from the IMi, but it has much better precision value compared to the control flow from the IMi such that it makes the combined control flow is a better control flow compared to the control flow from IMi. On the other hand, the precision value of the combined control flow is much lower than the precision value of the control flow from the ETM. Nevertheless, we choose the combined control flow since the precision value of the control flow from the ETM is almost a perfect score that can lead to an overfitting model.

3.3.6. Property of Control Flow

Within the control flow perspective, every task in the control flow has several properties. One of the most important properties of the control flow is time property. In [25], the time property includes processing time, waiting time, transferring time, queue time, validation time, and rework time. Due to data limitation, we only consider processing time and waiting time for this study.

Processing time

Processing time is the actual time a resource spent on an activity. Since we only have the completion time of the activity in the log, we assume the resource directly continues to do the next activity after she finishes one activity. In order to get the processing time, we sort the log by username and date time. Figure 39 shows the log snippet that is sorted by username and date time.

process_name	username	datetime	duration
Release Overall status of credit checks on customer order	USER1	28/11/2014 10:45	0:00:20
create sales Standard Item	USER1	28/11/2014 10:46	0:00:46
Release Overall status of credit checks on customer order	USER1	28/11/2014 10:46	0:00:09
create sales Standard Item	USER1	28/11/2014 10:47	0:00:47
Release Overall status of credit checks on customer order	USER1	28/11/2014 10:47	0:00:21
create Outbound delivery	USER1	28/11/2014 10:51	0:03:21
create Outbound delivery	USER1	28/11/2014 10:51	0:00:10
create Outbound delivery	USER1	28/11/2014 10:51	0:00:04
Change Current Qty Field for Arithmetic Operations in Doc.Process. on delivery item	USER1	28/11/2014 10:52	0:00:35
create Post goods issue item	USER1	28/11/2014 10:52	0:00:37
Change Current Qty Field for Arithmetic Operations in Doc.Process. on delivery item	USER1	28/11/2014 10:53	0:00:51
create Post goods issue item	USER1	28/11/2014 10:53	0:00:25
Change Current Qty Field for Arithmetic Operations in Doc.Process. on delivery item	USER1	28/11/2014 10:54	0:00:31
create Post goods issue item	USER1	28/11/2014 10:54	0:00:12
Change Order quantity in sales units on customer order	USER1	05/12/2014 09:08	
Change Requested delivery date on Sales document Header	USER1	05/12/2014 11:02	1:53:48

Figure 39 log snippet to find processing time

The activity processing time is between 0 seconds and 11.99 hours. The activity with processing time of 0 is the activity that has similar end time with the subsequent activities. This can happen since the log we have is a redo log. The redo log records every *cascade step* of one real activity the resource has done in the system. The *cascade step* is the immediate steps that are done by the system after one activity. For example, the resource real action is changing the route of shipment. The log does not only record the 'change route of shipment' activity, but also the 'delivery item changes' activity since this change also applies to all items in the delivery related to that shipment. Another example of a *cascade step* is the 'create G/L account document' activity. This activity is regularly executed in a batch so that the time information for the orders in one batch will always be the same. In extreme case such as an activity with the average of processing time of 0 means that this activity is the *cascade step* for all occurrences in the log.

On the other hand, we have maximum processing time of 11.99 hours. This extreme case can happen when a resource does some activities at the first hour, continues doing anything that was not recorded in the system, and later on she does some activities at the end of the day. This case indicates that there is an outlier in the dataset. Therefore, we have to omit the outlier from the log before we can calculate the processing time. To find the outlier, we use the interquartile range (IQR) value. The IQR value is the difference between the first quartile and the third quartile of a set of data. Using IQR value, we can set the lower limit and the upper limit of the processing time that is not considered as the outlier. After we cleanse the log from the outlier, we can analyze and find the average and the probability distribution of the processing time for every activity. The detailed result of the processing time for every activity can be found in Appendix C.

Waiting time

If we only consider the processing time as our time property in the control flow, we can never get the total throughput time nearly as in Table 2 for every business unit. Therefore, we should also look into the waiting time. The waiting time is the time span between the completion time of one activity until the start time of the subsequent activity. One of the possible reasons why the waiting time can happen is the activity waits for a third-party response before it can be triggered. One example is the waiting time between the create invoice / create accounting process activity and the create customer payment activity; BU1 and BU3 have around 44 days of waiting time while BU2 has 32 days. The 'create customer payment' activity can only be executed if the customer already paid the goods and these waiting times portray this condition.

In order to discover the waiting time, we sort the log by trace id and date time. Figure 40 shows a snippet of our log that is used to get the waiting time. From this figure, we can see there is a span of 192.9 hours between the 'create sales standard item' activity and the 'change net price on customer order' activity. The span of 192.9 hours is the waiting time of the 'change net price on customer order' activity. However, the span of 192.9 hours already includes the actual processing time of the 'change net price on customer order' activity because we only have the completion time of the activity. Therefore, when we calculate the average of the waiting time, we deduct the waiting time with the average of processing time of the related activity if the waiting time is bigger than the processing time. If the waiting time is smaller than the processing time, we assume there is no waiting time in between. For example, the average processing time of the 'rail' activity 'is 1.17 minutes while in the snippet log, the waiting time is 1.13 minutes. Therefore the duration in the snippet log does not contain the waiting time. Using this log, we can analyse the waiting time for every activity in the business units. Appendix C gives the detailed result of the waiting time.

trace id	process_name	datetime	duration
31	Standard Order	23/04/2014 11:10:01	
31	create sales Standard Item	23/04/2014 11:10:05	0:00:04
31	Change Net price on customer order	01/05/2014 12:03:58	192:53:53
31	create Outbound delivery	07/05/2014 08:03:37	139:59:39
31	Rail	07/05/2014 08:04:45	0:01:08
31	Change Current Qty Field for Arithmetic Operations in Doc.Process. on delivery item	07/05/2014 08:09:16	0:04:31
31	create transfer order	07/05/2014 08:09:18	0:00:02
31	Change Current Qty Field for Arithmetic Operations in Doc.Process. on delivery item	07/05/2014 08:09:20	0:00:02
31	create Post goods issue item	07/05/2014 08:09:37	0:00:17
31	create Pro Forma Inv f Dlv	07/05/2014 08:10:50	0:01:13
31	create accounting process	09/05/2014 23:15:06	63:04:16
31	create Customer payment	06/06/2014 06:35:59	655:20:53

Figure 40 the snippet log to find the waiting time

Furthermore, we need to model this waiting time in all control flows that we discovered beforehand because we found many cases with substantial time span between two activities. The waiting time indicates there are some unrecorded activities happening between two recorded activities. Therefore, in this study, we model the waiting time as a silent transition. The decision to model the waiting time as a silent transition because it has specific time distribution but it does not need the resource. This new type of silent transition is applied in all control flows. Figures 41 shows an example of the waiting time being modelled as silent transition. The silent transition in the red rectangle is the normal silent transition while the silent transition in the blue rectangle is the waiting time. In this case, the silent transition in the blue rectangle indicates the waiting time between the 'standard order' activity and the 'create sales standard item' activity. Furthermore, we modify the control flows that we discovered in the previous sections by adding this 'waiting time' transition as the predecessor activity of the activities with waiting time.

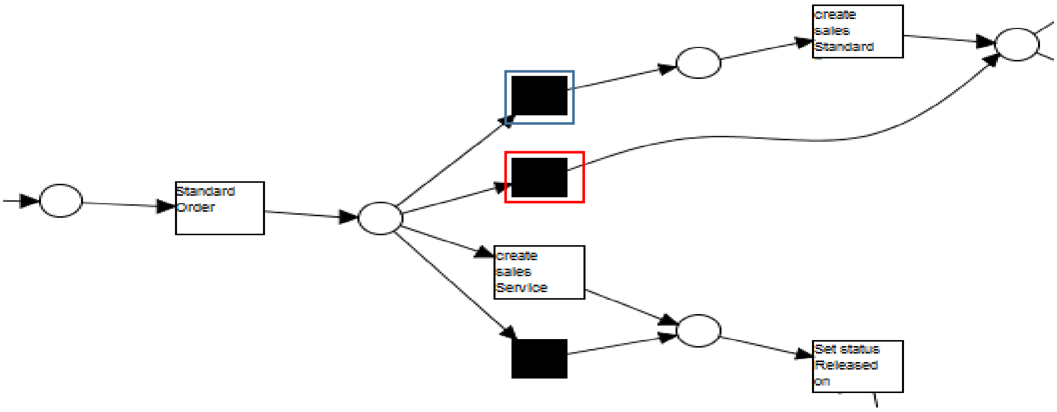


Figure 41 example of control flow with silent transition as waiting time

3.4. Resource perspective

Every manual activity needs a resource in order to be executed. In total, there are 359 unique resources where 181 resources have task responsibility at BU1, 59 resources have task responsibility at BU2, and 147 resources at BU3. After knowing the specific resources for each business unit, we analyse the relation between resources and activities in the control flow. There are two important characteristics of the resource perspective can be observed: work schedule and task responsibility.

Work schedule

Each resource has its own work schedule. Since DSM is a multinational company and operates around the clock, we observe many work schedule patterns. The work schedule pattern of a resource can be derived by discovering the first and last hour of activity the resources executed in the log every day throughout the last three months. We take three months as restriction with the assumption that a resource who does not do any activities in the log within the last three months can be assumed that she is not responsible anymore for the OTC process. After we get this information, we look into the mode value of the hour both for the first and the last hour for every day in a week. The mode value is the most frequent value in one dataset. Hence, the mode value of the first hour becomes the starting hour of work schedule and the mode value of the last hour becomes the completion hour of work schedule. By analysing the mode value, we can discover the regular working times of one resource. Not for all resources the work schedule can be easily determined due to a lack of data as can be seen in Figure 42.

	Monday				Tuesday				Wednesday				Thursday				Friday				Saturday				Sunday			
	1	2	3	mod	1	2	3	mod	1	2	3	mod	1	2	3	mod	1	2	3	mod	1	2	3	mod	1	2	3	mod
USER1	14	14	13	14	14	14	14	14	13	14	13	13	14	14	14	14	14	13	13	13								
	22	23	22	22	23	22	22	22	23	22	23	23	22	23	22	22	22	22	22	22								
USER2									10			10																
									10			10																

Figure 42 example of work schedule pattern

In Figure 42, we can see there are two resources with extreme different data. We can straightforwardly discover that the work schedule pattern for USER1 is Monday – Friday from 14.00 – 22.00. On the other hand, it is difficult to determine the work schedule pattern for USER2 since the log only records him as a resource in the 1st month on Wednesday. Therefore, we consider USER2 as an invalid resource in the OTC process. A valid resource is a resource that has sufficient amount of record in the last three months of the log. A sufficient amount of record means a resource has done at least one activity related to the OTC process every month in the last three months.

While discovering the work schedule pattern, we also notice the 'special' resource. The 'special' resource is a system resource or a shared resource which the work schedule exceeds 20 hours per day and appears all day. For these resources, we set their work schedule to always available. Appendix D gives the complete overview of the work schedule for every resource.

Task responsibility

A resource usually can have more than one task responsibility and vice versa. With this in mind, we try to mine the social network within one business unit using social network analysis. Social network analysis is a method or technique to find the interpersonal relationship in one environment that is usually presented as a graph or metric [15]. This analysis can discover the group of resource which has same task responsibility by using the similar task metric. Regrettably, the result of this analysis could not give clear separation between groups, as can be seen in Figure 43, such that it is difficult to discover the group of resource with the similar task responsibility. Consequently, we set the task responsibility to the lowest granularity which is the resource. To discover the task responsibility of the resources, we use the log to create groups of resources according to the activity name since we have more resources than activities. Appendix D gives more detail information about the task responsibility for every resource.

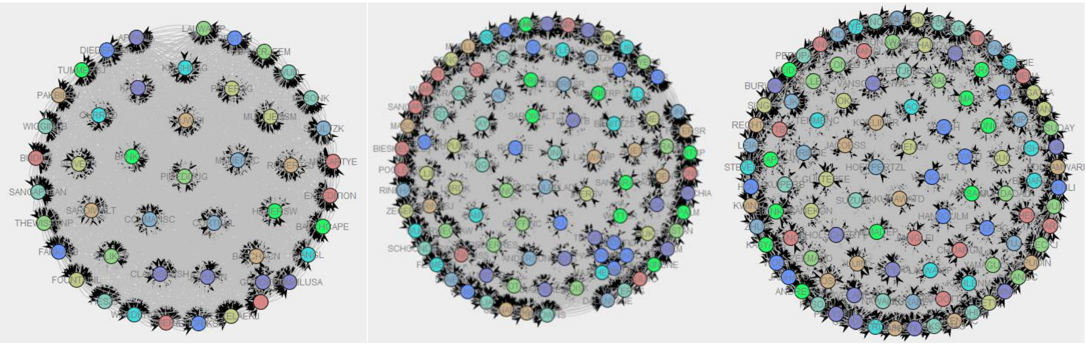


Figure 43 (left) BU2's SNA. (middle) BU1's SNA. (right) BU3's social network analysis

3.5. Data perspective

In data perspective, we discover the probability of one task is being executed in the control flow. In order to get that probability, we calculate the fraction of the occurrence of the specific activity and the total of the occurrence of all possible activities in the XOR-choice. Figure 44 shows one of the example of the data perspective of the activity in the *Order Intake* sub process of BU1. In the Figure 44, we have one variable, i.e. *xor7*, to determine which path the XOR-choice should be picked. In order to get the probability, we should find the total occurrences of each activity first in the log. The total occurrence of activity nr.8 is 3,846 and the total occurrence of activity nr.14 is 1,790. Hence, the total occurrences of activity nr.8 and nr.14 in the log is 5,636. Afterwards, we can calculate the probability of activity nr.8 and activity nr.14 by dividing the total occurrence of the specific activity with the total occurrence of both activities. Table 10 shows the probability of every value for this variable. Appendix E gives the complete overview of the data perspective for each business unit.

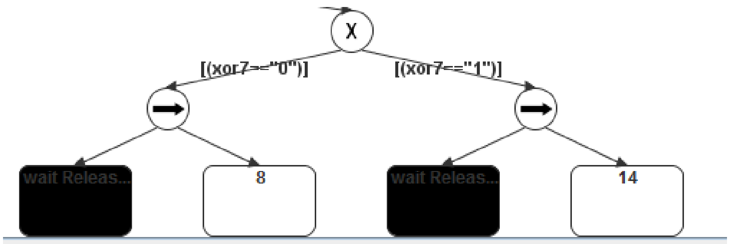


Figure 44 an example of XOR-choice in BU1 control flow

Table 10 XOR7 possible value and its probability

Variable	Value	Probability
Xor7	0	0.682
	1	0.318

3.6. Environment perspective

Within environment perspective of the process model, there is only one property, i.e. the arrival process. The arrival process property can be acquired by dividing the timespan of the log and the total amount of the 'standard order' activity in the log. Table 11 shows the average inter-arrival time of the order for each business unit.

Table 11 Average inter-arrival time of the order

BU	Avg inter-arrival time
BU1	0.173 hours
BU2	0.529 hours
BU3	0.162 hours

3.7. Experiment perspective

The experiment perspective will be the same for each business unit. The experiment perspective has only one property, i.e. the simulation property. We set the simulation property of the experiment perspective with FIFO queues, push allocation, 8,760 hours replication length as a representation of one year run-time with 10 percent out of the replication length as warm-up period, and 30 replications.

3.8. Validation process

After we discover all the perspective of the process model, we conduct two processes of validation. The first process is the control flow validation. This process involves the process expert to validating the process. The second process is the process model validation. The second one focuses to validate the time behaviour of the process, i.e. the throughput time, by using the simulation functionality of PETRA.

3.8.1. Control flow validation

In order to validate our results, we asked the opinion of the process expert about the result. From his expert opinion, our results are indeed plausible control flows in the DSM case. One of the examples where our result depicts the real situation in DSM is the position of the 'change net price on customer order' activity in the BU2 control flow. In BU2, the 'change net price on the customer order' activity is in the middle of the *distribute* sub process while the other two business units have this activity in the middle of the *order intake* sub process. From the original log, this activity is listed on the file 'change order' which means it relates to the *order intake* sub process; yet, in BU2, this activity is executed in the *distribute* sub process. From the validation process with the process expert, apparently, this activity position can be explained due to the pricing regulation that is applied in BU2. BU2 uses a different method of pricing because of the characteristic of the products of BU2 which heavily depend on oil prices. Therefore, the price might change after the shipment is created due to oil price changes. Another example where our result depicts the real situation in DSM is the shipment type which only appears in one specific business unit, e.g. BU2 has the rail shipment while the other business units does not have because in BU2 the rail shipment is one of the most common shipment method in BU2. This situation also applies on BU3 with the courier shipment and BU1 with the sea shipment.

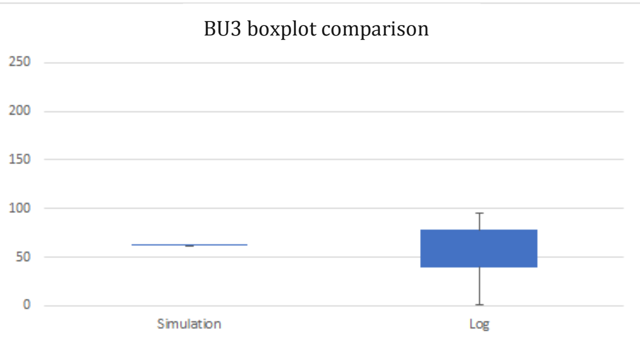
3.8.2. Process model validation

The control flow itself is not enough to validate the whole process since the process model has several other perspectives. To further strengthen the validation from the process experts, we conduct the process model validation using the simulation capability of PETRA. We develop a simulatable process model for each business unit using all the perspectives have been discovered in previous sections. The process model validation is an important step to measure how close our models is to reality and we choose the throughput time as our validation measurement.

The result of the first experiment is considerably unacceptable for some business units as shown in Table 12 because they have a significant difference of the throughput time resulted from the simulation and the log. The only business unit that has a close result of the throughput time between the simulation result and the log is BU3 as shown in Figure 45. Therefore, we should look into detail the model once again and try to better align the model for BU1 and BU2.

Table 12 Comparison of throughput time between log and simulation - first trial

Throughput time	BU1 (days)	CI 95 %	BU2 (days)	CI 95 %	BU3 (days)	CI 95 %
Simulation	56.612	0.0163	45.469	0.019	61.413	0.032
Log	63.622		52.807		61.773	



	Avg.throughput time simulation (days)	CI 95 %	Avg. throughput time log (days)
BU3	61.41291432	0.032887	61.77294544

Figure 45 BU3 boxplot comparison and detail of throughput time comparison

BU2

In order to better align the process model with the reality, we performed the following steps:

1. Compare the number of occurrences every activity on the log and the simulation result
2. If there are some activities with significant differences, examine the trace pattern to find the best position for that particular activity.
3. Relocate the activity to the best position.
4. Run the simulation and compare the throughput time.

For BU2 case, Table 13 shows two activities with a significant gap between the number of occurrences in the log and the simulation result.

Table 13 comparisons of occurrence between simulation and log for intermodal and change material staging

Activity	Number of occurrence in log	Number of occurrence in simulation
Intermodal	760	8
Change material staging	2,566	8

In the original model, the 'intermodal' activity and the 'change material staging' activity are located after the 'create sales standard item' activity as can be seen in the top figure of Figure 46 (see red box). By examining the top 150 trace pattern of BU2, we conclude that these two activities have been misplaced since these activities are executed after the 'create outbound delivery' activity in most of the trace patterns. Therefore, we move the 'intermodal' activity after the 'create outbound delivery' activity as one of the options of the shipment. Meanwhile, the 'change material staging' activity is placed after the shipment creation and also we add a silent transition to handle the log without this specific change (see blue box in the bottom figure of Figure 46). Afterwards, we run the simulation again.

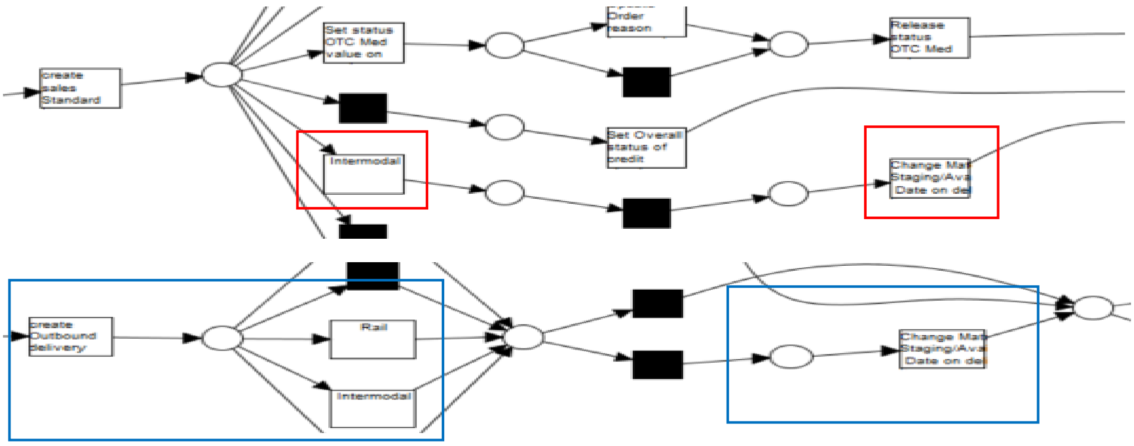
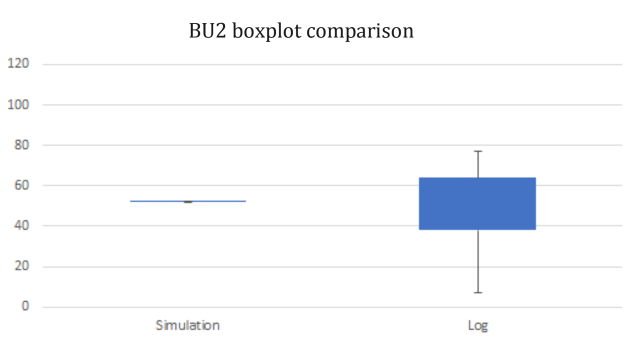


Figure 46 (top) control flow before intermodal is replaced. (bottom) control flow after intermodal is replaced

The activity relocation leads to a positive result as can be seen in Figure 47. The simulation boxplot is clearly in the range of the log boxplot. Therefore, we cannot conclude our BU2 model and reality has a significant difference.



	Avg.throughput time simulation (days)	CI 95 %	Avg. throughput time log (days)
BU2	52.36403713	0.069006	52.80749103

Figure 47 BU2 boxplot comparison and detail of throughput time comparison

BU1

We did the similar steps as BU2 for BU1 to better align the process model with the reality and we found an activity with a significant gap of occurrences as can be seen in Table 14. The ‘change requested delivery’ activity is an activity within the sub process sales document changes.

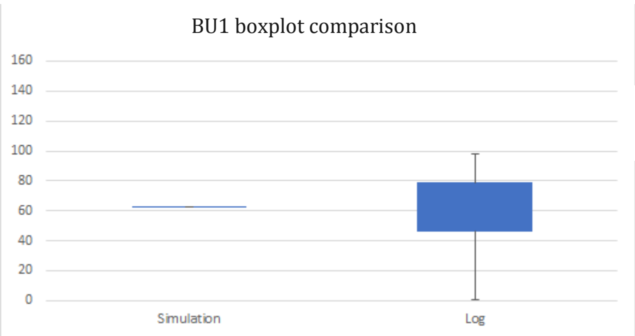
Table 14 the comparison of the number of occurrences between simulation and log for change requested delivery

Activity	Number of occurrence in log	Number of occurrence in simulation
Changed requested delivery	11,934	4,449

Apparently, we could not find the best position for this activity by examining the trace pattern of the BU1 since this activity can be executed anywhere on the sub process sales doc changes. Therefore, we propose three new steps:

1. Check the silent transition in the model and check in which silent transition has a significant number of ‘move on log’ occurrence for the activities in question.
2. Change the silent transition into a manual activity with a similar name as the activity in question while the activity in question is changed to a silent transition.
3. Run the simulation. Try the first option: only one activity at the new position. If the result is not good, try the second option which is the control flow has two activities with similar name in different positions by keeping the activity in question as it is.

In the BU1 case, the first option already yielded a promising result as shown in Figure 48. Hence, we cannot conclude our BU1 model and reality has a significant difference.



	Avg.throughput time simulation (days)	CI 95 %	Avg. throughput time log (days)
BU1	62.46842005	0.042567	63.62263726

Figure 48 BU1 boxplot comparison and detail of throughput time comparison

Thus, Figure 49 to 51 show the final process model after the validation process for BU1, BU2, and BU3 respectively.

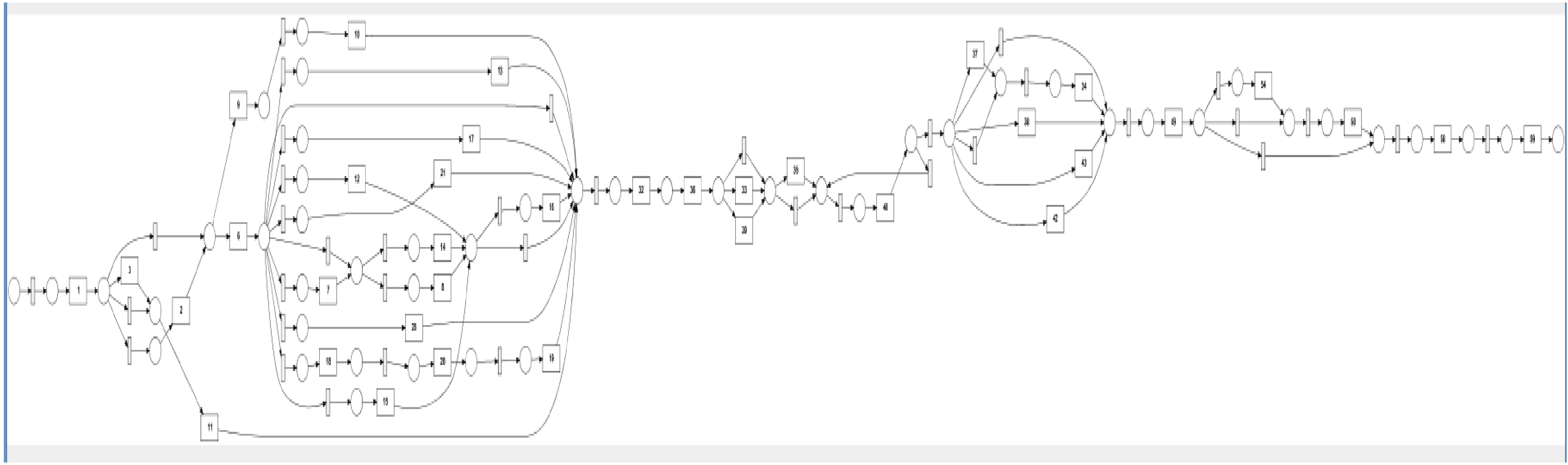


Figure 49 BU1 final control flow

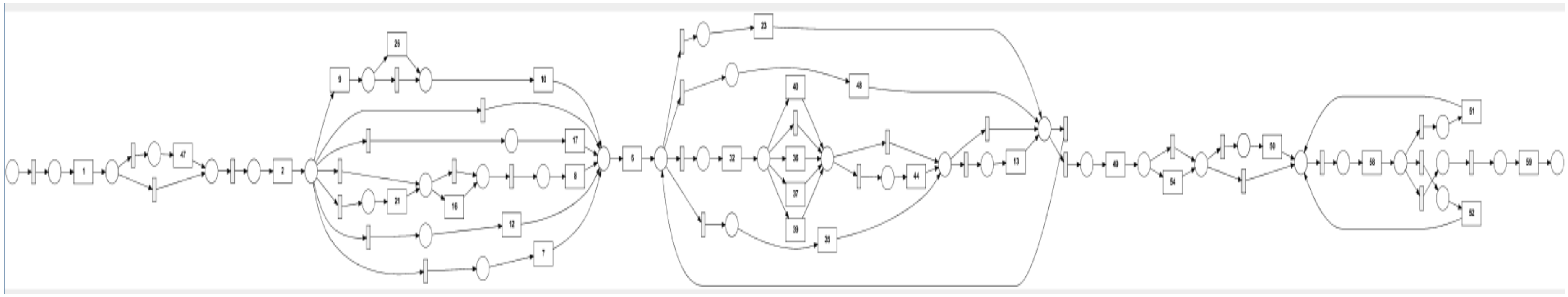


Figure 50 BU2 final control flow

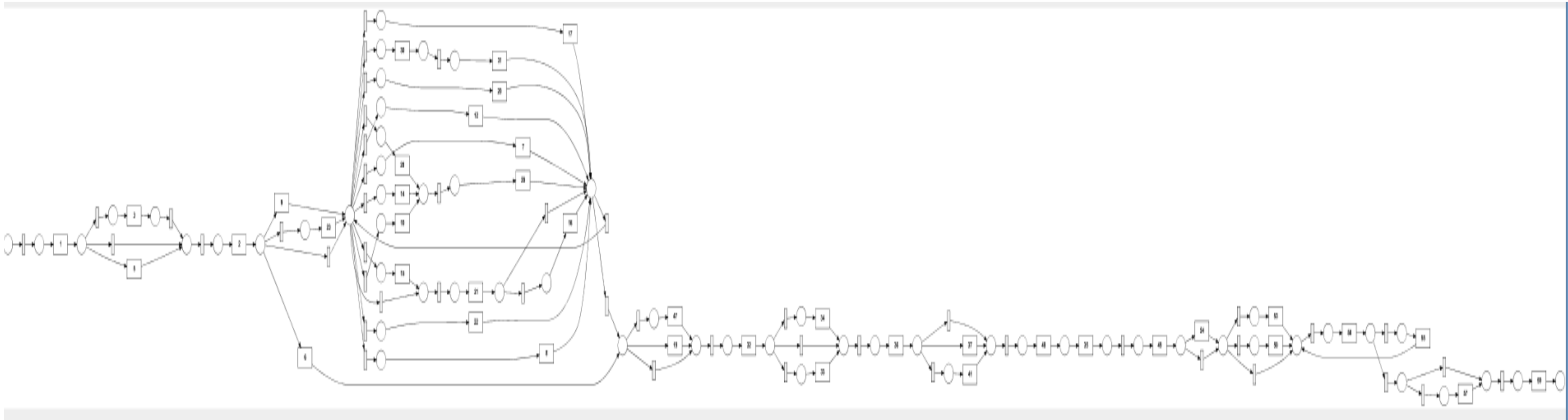


Figure 51 BU3 final control flow

3.9. Conclusion

The goal of this chapter is to discover the process model of the OTC process in each business unit that closely represent the real life situation. In order to achieve the goal, we already presented the steps to discover the control flow perspective, the resource perspective, the data perspective, the environment perspective, and the experiment perspective. Furthermore, we already showed that the process mining algorithm can discover the control flow perspective for every business unit with good replay fitness and precision values even though we only use a part of the log. We also have shown the steps to combine the control flow resulted from two different mining algorithms such that it can improve the precision value while still maintains high replay fitness value. Within the control flow perspective, we also presented the steps to discover the properties of the control flow, i.e. the time property. Moreover, the obtained process models are also verified by an expert which states that these control flows are plausible work flows in DSM with several examples where the control flow indeed depicts the real life situation in DSM according to the process expert's opinion. Finally, we showed that our process models already closely represented the real life situation because there are no significant difference between the throughput time from the simulation and the log. Thus, we can conduct our next step with the resulted process model as a base.

Chapter 4. Process Variants Comparison and Analysis

As explained in Chapter 2, a configurable process model has several perspectives and every perspective can be configured. The configuration point in each perspective is essentially a variant of the process model. Hence, using these perspectives, we conduct an analysis and comparison to find variations between the process models. Due to time limitation, we only analyse and compare two out of three possible perspectives since the environment perspective has only one property to compare and every business unit has the same value of the experiment perspective:

a. Control flow perspective
   In this perspective, we conduct an analysis and comparison of the control flow itself and one important property that belongs to this perspective, i.e. time property. We compare and analyse the resulted control flow and the time property from the results of the previous chapter. The time property deals with all time-related information about an activity in the control flow. The time property has many types, e.g. waiting time, processing time, setup time, and queue time. In our study, we analyse and compare two of them: waiting time and processing time.
b. Resource perspective
   The resource perspective deals with the work schedule of the resource and the responsible tasks of the resource. Every resource has its own work schedule and set of responsible task. We use the result of resource property from the previous chapter to compare and analyse the difference between the business units.

This chapter starts with the analysis and comparison of the control flow perspective in Section 4.1. In Section 4.2, the analysis and comparison of the resource perspective are given. Finally, we conclude this chapter in Section 4.3.

4.1. Control flow perspective

The control flow perspective is the main perspective of the process model since the other perspectives depend on this perspective. The control flow consists of activities and every activity has properties. Therefore, in this section we analyse the control flow in three aspects: the activity, the time property of the activities, and the control flow itself.

4.1.1. Activity analysis

Since we used the log that contains the top 150 trace patterns when we discovered the control flow, not every activity in each business unit log is modelled in the control flow. In total BU1 has 78 different activities in the log, BU2 has 72 different activities in the log, and BU3 has 80 different activities in the log. However, only 34 different activities have been modelled in the control flow of BU1, 30 activities in the control flow of BU2, and 41 activities in the control flow of BU3. The different total number of activities in the control flow shows that there are some activities that only exist in one particular business unit. Table 15 gives the details of the activities within the business units. This configuration point decides which activity should be hidden or blocked. Certainly, the 20 shared activities between three business units should never be blocked or hidden.

Table 15 activity in business unit(s)

Business unit(s)	Total number activity	Activity name
BU1- BU2- BU3	20	Change current qty. in delivery item, change order qty, change requested delivery date, create accounting process, create customer payment, create invoice, create outbound delivery, create post goods issue item, create pro forma inv, create sales standard item, create transfer order, pick up(ex work) shipment, release overall status credit checks, release status availability problem, release status otc med value, road shipment, set overall status credit check, set status otc allowed, set status otc med, standard order
BU1- BU2	2	Change net price, intermodal shipment
BU1- BU3	8	Change gross weight, change route, create pick order, create sales service, release delivery block, release status lead time, set status availability problem, set status released
BU3- BU2	3	Change customer purchase order number, Change schedule line date, and Change shipping condition.
BU1	4	Delivery item changes, sea shipment, set status lead time problem, update batch number
BU2	5	Change material staging, create credit memo, create debit memo, rail shipment, update order reason
BU3	10	Courier shipment, create G/L document, create intercompany billing, create invoice (F1), create pro forma for order, create sales TPO it. GR basedinv, Release status alternative, Release status OTC deliver manually, set status alternative, set status OTC deliver manually

4.1.2. Time property analysis

As well as the previous chapter, we divide the analysis and comparison of the time property on two types of time property: processing time and waiting time.

Processing time

Using the result from the previous chapter, Table 16 shows the general statistics of processing time for every business unit. On average, BU3 can finish its activities faster than the other business units.

Table 16 general statistics of processing time

Business Units	Average processing time of all activities	Std. Deviation	Min processing time	Max processing time
BU1	13.68 mins	13.70 mins	11 secs (delivery item changes)	41.95 mins (change order qty)
BU2	10.36 mins	8.76 mins	27 secs (pick up(ex work) shipment)	28.95 mins (create transfer order)
BU3	9.11 mins	8.51 mins	0 secs (create G/L account document)	33.64 mins (change order qty)

Furthermore, we also compare average processing time for 20 shared activities among the three business units as can be seen in Figure 52. Every business units has different processing time of the activity. In most of the cases, BU1 has the highest processing time while BU3 becomes the lowest. This is in line with previous finding which shows that BU3 has the lowest average of processing time. We can exploit these findings when we optimize the business process on later stage.

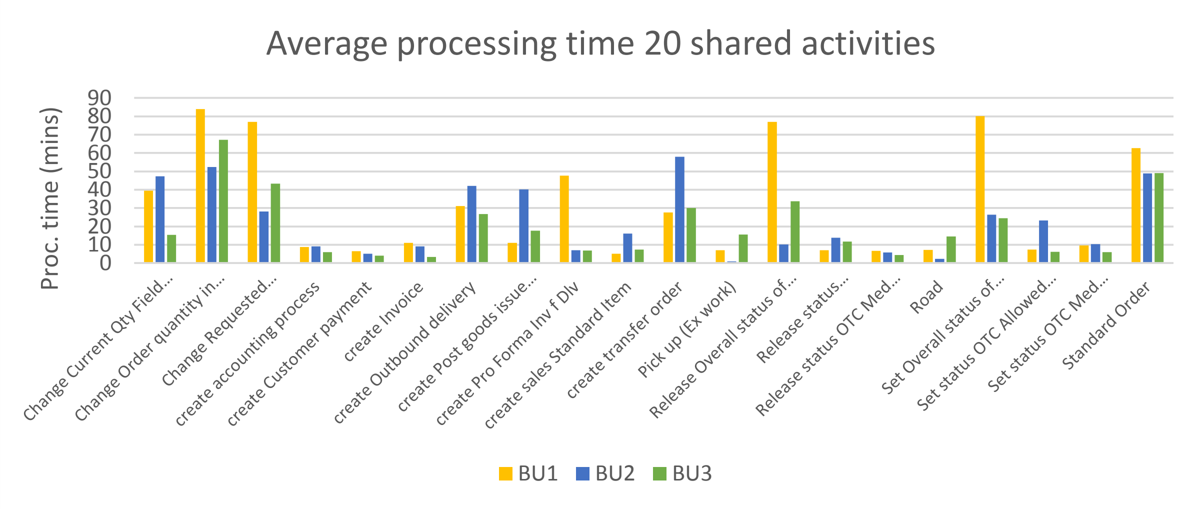


Figure 52 average processing time 20 shared activities bar chart

Waiting time

Using the result from the previous chapter, Table 17 shows the general statistics of waiting time for every business unit.

Table 17 general statistics of waiting time

Business Units	Average waiting time / activity	Std. Deviation	Min waiting time	Max waiting time
BU1	3.15 days	7.72 days	0 sec (change gross, change current qty, update batch number, delivery item changes)	43.92 days (create customer payment) 14.15 days (set overall status of credit check)
BU2	3.95 days	6.38 days	3.09 mins (set status OTC allowed)	31.91 days (create customer payment), 12.37 days (create credit memo)
BU3	3.19 days	7.76 days	0.69 mins (create sales TPO)	43.78 days(create customer payment), 28.05 days (create G/L account document)

From the Table 17, we can see that BU1 has several activities with waiting time of 0. It means these activities never have waiting time to be executed. We also can see that in every business unit the waiting time of the 'create customer payment' activity is the longest waiting time activity.

Furthermore, we also compare the average waiting time for the 20 shared activities among three business units. Every business unit has different waiting time of the activity. However, unlike the processing time, in the waiting time, the 'create Customer payment' activity dominates the other activities. If we compare the waiting time of the 'create Customer payment' activity and the throughput time of the complete process in Table 2, the 'create customer payment' activity takes around 70 percent of throughput time for BU1 and BU3, and 60 percent for BU2. Nevertheless, the 'create customer payment' activity is merely a waiting activity such that we cannot do much about this. Therefore, we only focus on the other 19 activities when it comes to optimizing as can be seen in Figure 53.

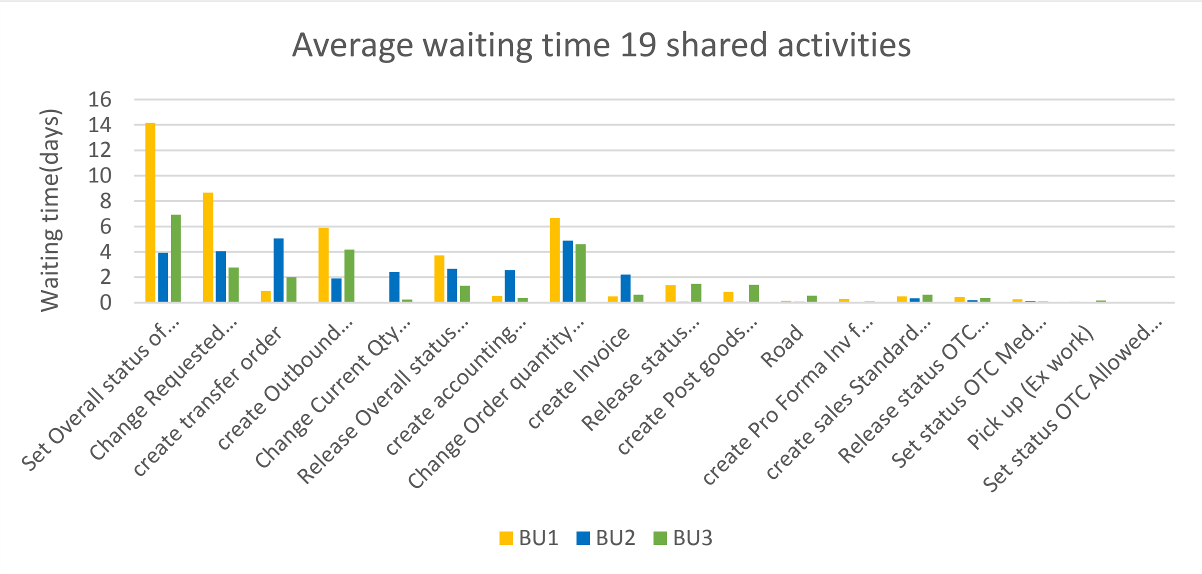


Figure 53 average waiting time 19 shared activities

One noticeable observation can be made from the waiting and processing time: the waiting time dominates the throughput time. From Table 2, we learned the average number of activity per order for each business unit. If we took the maximum processing time for each business unit from Appendix C, i.e. 55 minutes, and multiply with the average number of activities per trace, the processing time ratio over the throughput time in one trace is merely around 0.9 % for BU3, 0.85 % for BU1, and 1.02 % for BU2. These ratios can surely be smaller if we use the precise processing time. This finding indicates that we should put more effort to optimize the waiting time compared to the processing time in the next step.

4.1.3. Control flow analysis

Control flow is the main source of variants in the business process since every business unit can do similar things in several ways. We divide our analysis and comparison into four parts due to the complexity of the complete process.

Sub process sales document creation

The sub process sales document creation starts when the customer orders the goods and ends when the resource or system performs one of the sales type creation activity. There is only a slight difference between the business units as can be seen in Figure 54.

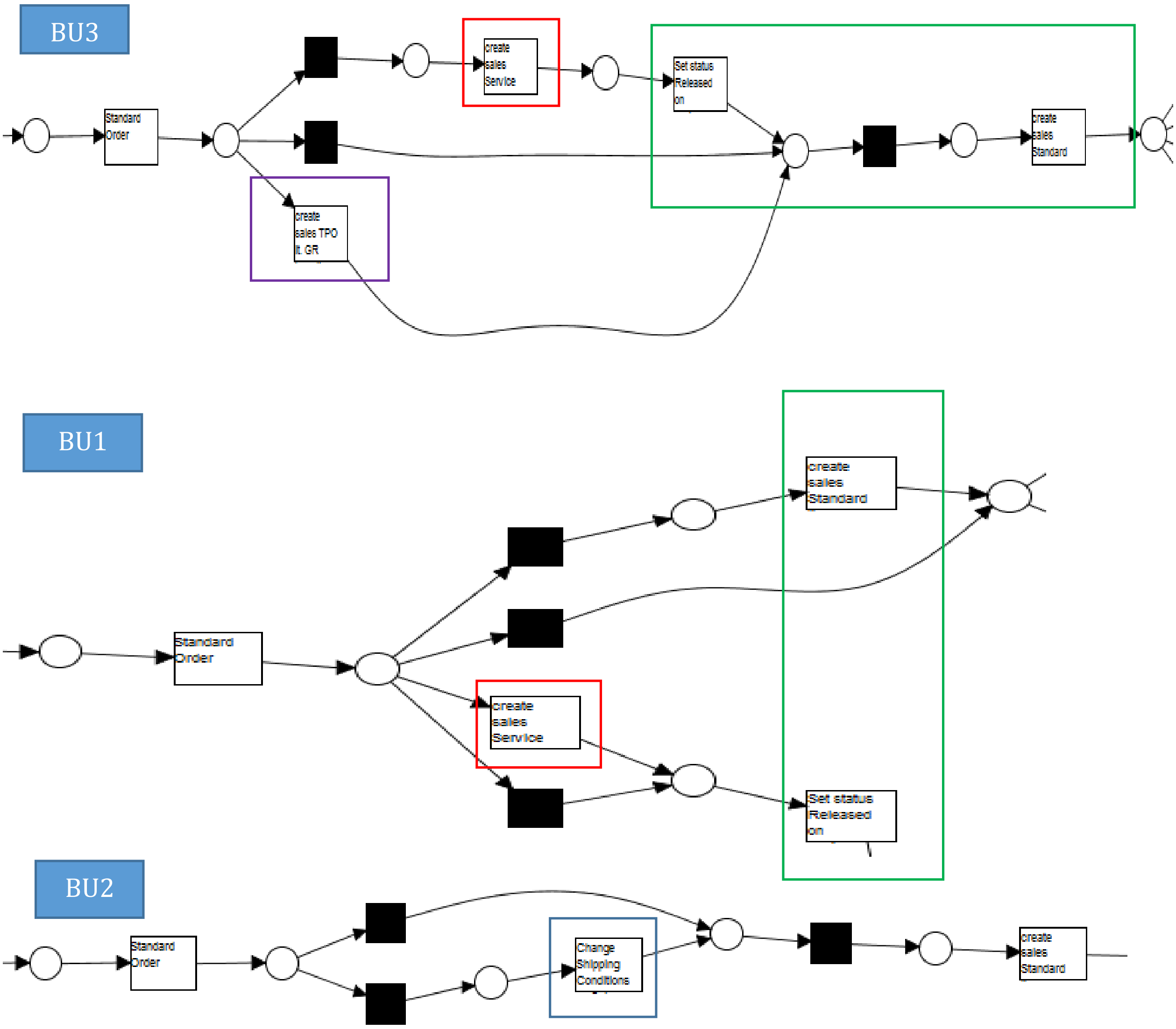


Figure 54 Sub process sales document creation control flow

One point of variation is the sales item variants between the business units. BU3 and BU1 have the sales service as an item (see the red box in BU3 and BU1 control flow) while BU2 does not. On the other hand, even though we already divide the sales item changes as a different sub process, we can discover BU2 can have some changes in early stage which is the 'Change shipping condition' activity (see the blue box in BU2 control flow). This activity can be executed after the 'create standard order' activity although with very low frequency (only 0.5 % of all cases). Besides the sales service, BU3 also has another type of standard order: the 'create sales TPO it. GR basedinv' (see the purple box in BU3 control flow) though with very low occurrence (1% of all cases).

Another point of variation is the activity after creating the sales service. In BU3, we only have one option after creating the sales service which is the 'set status released' activity and it is continued with the 'create sales standard item' activity (see green box in BU3 control flow). However, in BU1, these two different types of sales item are separated into two different paths such that the sales service order does not have the 'create standard item' activity as its succeeding activity (see green box in BU1 control flow). These paths meet again in the sub process delivery. As a consequence, we discover the 'move on log' event in the 'create sales standard item' activity about 6% of total trace in BU3.

Sub process sales document changes

The sub process sales document changes starts after 'create sales standard item' activity and finishes before 'create outbound delivery' activity. It consists of many activities and can be considered as the most complex sub process in the OTC model. In fact, this sub process is not too desirable from the business side yet it is unavoidable because every change in the sales document has negative impact to the throughput time and also the resource utilization. Figure 55 depicts the sub process for every business unit.

In general, there are two types of changes which can be executed: status changes and item changes. In most cases, the setting status activity will be followed by a release for that status except for the 'set status OTC allowed value' activity (activity nr.6). Moreover, the 'set status OTC allowed' activity has different positions in every business unit. In BU3, this activity has a 'highway' route such that every case with this status can directly continue to the 'create outbound delivery' activity (see red rectangle in BU3 control flow) without executing any change activities. In BU1, this activity can be followed by many options of change activity (see green rectangle in BU1 control flow) and one of the options is also the 'highway' route like BU3 by triggering the silent transition option (see blue rectangle in BU1 control flow). Different case happens to BU2, this activity becomes the final activity before an outbound delivery is created (see purple rectangle in BU2 control flow).

The main variation between business units is the number of possibility of change activities in each business process. Table 18 shows what possible activities in which business unit and its frequency. From the total of 23 different activities, we can find 8 activities shared in all business units. This set of activity can be considered as a minimum set of changes activity that should be allowed in one business unit.

Another main difference in this sub process is BU3 has a loop structure while other business units do not. The loop structure indicates BU3 has executed more than two sales doc changes in most of the orders. This loop structure gains its justification by investigating the number of orders with more than two changes as can be seen in Table 19. BU3 has almost 47 % of the total order with changes more than two while BU1 has 30 % and BU2 only has 5 % of the total order. BU2 has the least percentage of the order with more than two changes because 95.8% of the orders go directly to the 'set status OTC allowed' activity as can be seen in Table 18.

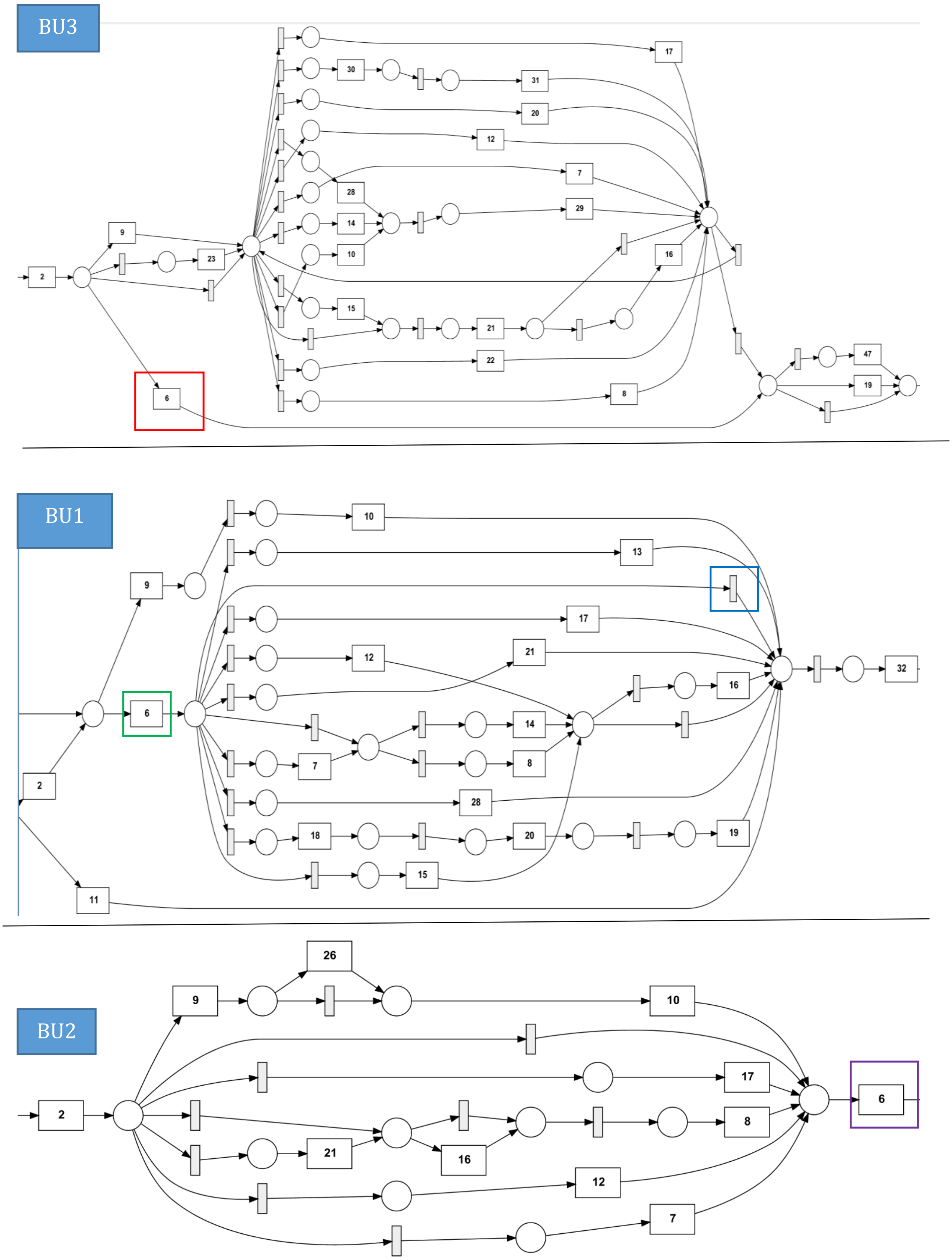


Figure 55 Sub process sales document changes control flow. (top) BU3, (middle) BU1, (bottom) BU2

Table 18 activities in sub process sales document changes

Activity	BU3- total occurrence / total records (%)	BU1 - total occurrence / total records (%)	BU2 - total occurrence / total records (%)
Change customer purchase order number	6.10%		
Change net price		6.88%	
Change order quantity	6.49%	5.10%	3.92%
Change requested delivery date	13.62%	16.62%	
Change route	6.31%	2.95%	
Change schedule line date	15.55%	9.26%	3.85%
Change shipping conditions	1.25%		
Create pro forma order	3.89%		
Release delivery block	2.74%	6.46%	
Release overall status of credit check	21.18%	12.01%	10.03%
Release status alternative	30.38%		
Release status availability	7.61%	12.76%	1.61%
Release status lead-time	0.77%	2.65%	
Release status OTC delivery manually	23.22%		
Release status OTC med	7.96%	9.07%	2.77%
Set overall status of credit check	7.57%	4.37%	1.62%
Set status alternative	21.10%	5.09%	
set status availability		4.41%	
set status lead-time problem		2.36%	
Set status OTC Allowed	91.47%	83.62%	95.84%
Set status OTC delivery manually	12.24%		
Set status OTC Med value	8.32%	9.06%	2.77%
Update order reason			0.63%

Table 19 comparison of total order with different number of changes

Num of Changes	BU1	%	BU2	%	BU3	%
<= 2	17,729	69.1	10,323	94.9	14,614	53.2
> 2	7,927	30.9	549	5.1	12,852	46.8

Since this sub process has a complex structure, it becomes interesting to find out which activity has big impact on the throughput time. In Table 18, we also can observe which activity has a high number of occurrences. Using this occurrence and also the waiting time from the previous section, we can calculate the impact value by multiplying the occurrence and the normalized value of the waiting time (the calculation can be seen in the Appendix F). From this calculation, Table 20 shows top 5 activities for each business unit that has a big impact on throughput time. There are two change activities that always appear in the top 5 of business units: 'Set overall status of credit checks' and 'change schedule line date'. This finding can be a good insight for the next step.

Table 20 top 5 activities with big impact in sub process sales document changes

Rank	BU1	BU2	BU3
1	Change Requested delivery date	Change Schedule line date	Set Overall status of credit checks
2	Change Schedule line date	Release Overall status of credit checks	Release status OTC Deliver manually
3	Set Overall status of credit checks	Change Order quantity	Change Schedule line date
4	Set status Alternative	Set Overall status of credit checks	Change Requested delivery date
5	Release Overall status of credit checks	Release status OTC Med value	Set status Alternative

= shared activity in all business units = shared activity between BU1 – BU3 = shared activity between BU1-BU2

Sub process delivery – shipment – transfer order

This sub process starts from the 'create outbound delivery' activity and ends before the 'post goods issue item' activity. The main difference in this sub process between the business units is the ordering between the activities. Figure 56 shows that in BU2, we could have the order either (delivery-shipment) – transfer order or transfer order – (delivery – shipment) due to the loop structure. However, in BU3 and BU1, we have a clear ordering between delivery, shipment, and transfer order. In BU3, the order is delivery – shipment – transfer order while in BU1, the order is delivery – (shipment) – transfer order – shipment. The first shipment activity in BU1 consists of road and intermodal shipment while the second one consists of sea and pick up (ex work) shipment. In total, BU3 has 3 different types of shipment: courier, road, and pick up (ex work), BU2 has 4 types: road, pick up (ex work), rail, and intermodal, and BU1 has also 4 types: road, intermodal, sea, and pick up (ex work).

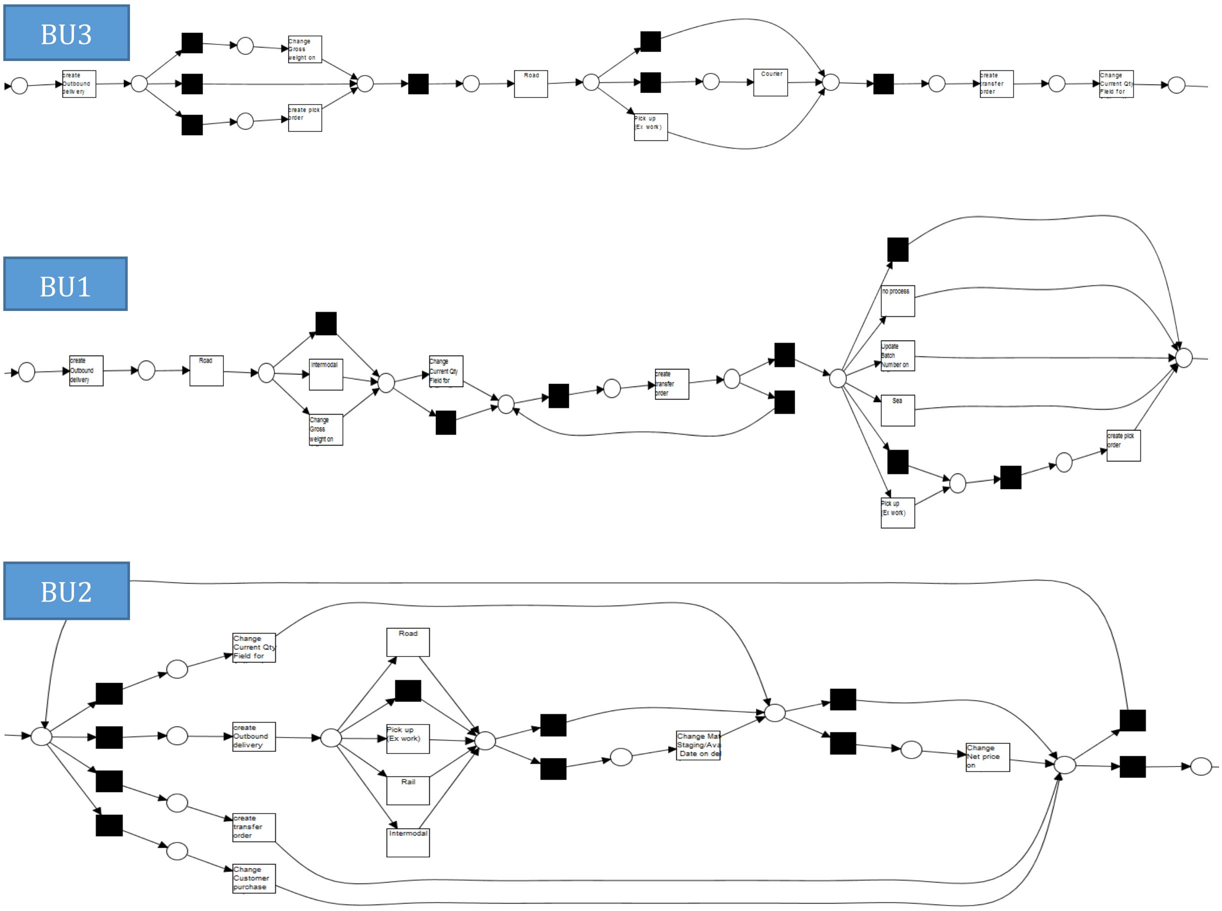


Figure 56 sub process delivery-shipment-transfer order control flow. (top) BU3, (middle) BU1, (bottom) BU2

The loop structure in BU2 can be justified by calculating the total number of deliveries for every order in each business unit. Table 21 shows that BU2 has almost 35% of total order that has more than 1 delivery while the other two business units only has 10,8% at most. Besides the BU2 loop, another loop structure discovered is the 'transfer order' activity in BU1 model.

Table 21 comparison of total order with different number of delivery

Number of deliveries	BU1	BU2	BU3
1	22,264 (89.2%)	7,077 (65.24 %)	24,381(90.04 %)
> 1	2,694 (10.8%)	3,770 (34.76 %)	2,699 (9.96 %)

Sub process post goods issue item – invoicing – accounting process – customer payment

Figure 57 shows the *post goods issue item-invoicing-accounting process-customer payment* sub process of BU3, BU1, and BU2. The main differences are on the invoicing process and accounting process. In the invoicing process, BU3 has two types of invoice: invoice F1 and normal invoice (see green box in BU3 control flow). Thus, the other two business units only have the normal invoice (see green box in BU1 and BU2 control flow). In the accounting process, BU3 and BU2 have a loop structure (see red box in BU3 and BU2 control flow) while BU1 does not. An order in BU3 has a loop in the 'accounting process' activity when there is an intercompany billing while an order in BU2 has loop due to the 'credit memo' activity or the 'debit memo' activity. The last variation point is BU3 has the 'create G/L document' activity while the other two do not (see the blue box in BU3 control flow).

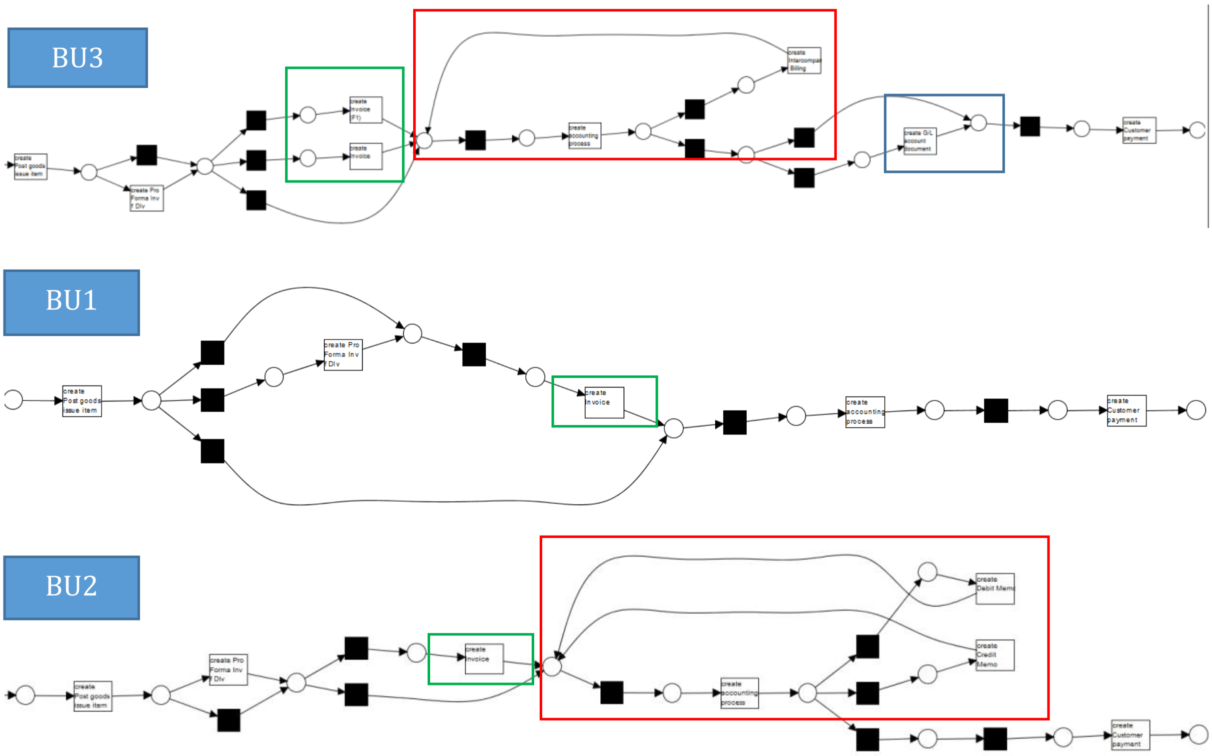


Figure 57 sub process post goods issue item-invoicing-accounting process-customer payment control flow. (top) BU3, (middle) BU1, (bottom) BU2

4.2. Resource perspective

Using the result from the previous chapter, we can discover 52 different work schedule patterns. Table 22 shows the comparison of the total resources of each work schedule pattern for every business unit. The generic encoding of work schedule pattern is 'xx'-'yy' where the 'xx' is the start of working time and the 'yy' is the end of working time. The work schedule patterns that follow the generic encoding have the working schedule: Monday to Friday from the 'xx' to 'yy'. Furthermore, for the work schedule patterns that do not follow the generic encoding, Table 23 gives the detailed explanation.

Table 22 work schedule patterns

Work schedule pattern	BU1	BU3	BU2	**Work schedule pattern**	BU1	BU3	BU2	**Work schedule pattern**	BU1	BU3	BU2
Always available	23	9	13	**02 – 17 (mo-sat)**	1	0	0	**07 - 15**	14	1	0
10 - 23	0	2	0	**02 – 22**	1	0	0	**07 - 17**	0	2	0
01 - 11	7	0	0	**02 - 09**	1	2	0	**07 - 20**	0	0	1
01 - 13	2	0	0	**03 - 11**	3	7	1	**07 - 23**	1	0	0
12 - 20	0	2	0	**03 - 15**	1	0	0	**08 - 12**	0	0	1
13 - 21	1	5	3	**03 - 16**	1	1	0	**08 - 15**	0	2	0
13 - 23	1	0	2	**03 - 09**	1	0	1	**08 - 16**	22	29	12
14 - 21	0	3	0	**04 - 12**	3	0	0	**08 - 17**	2	0	0
14 - 22	5	10	2	**04 - 16**	1	0	0	**08 - 20**	0	1	0
15 - 23	5	12	2	**04 - 18 (mo-fri) 04- 15 sat**	1	0	0	**08 - 22**	1	0	2
17 - 23	0	3	0	**05 - 13**	10	2	1	**08 - 23**	1	1	0
01 - 09	3	0	0	**05 - 14**	1	0	0	**09 - 15**	0	0	1
01- 09 (mon-fri, sun)	0	2	0	**05 - 15**	1	0	0	**09 - 16**	0	6	0
02 - 10	26	10	2	**05 - 19**	1	0	0	**09 - 17**	27	34	10
02 - 10 everyday	1	0	0	**05 - 21**	1	0	0	**09 - 17 (mo-sat)**	0	0	1
02 - 12	4	0	0	**05 - 22**	1	0	0	**09 - 20**	0	0	1
02 - 12 everyday	0	0	1	**06 - 14**	4	0	2	**09 - 21**	0	1	0
02 - 17	1	0	0	**06 - 18 everyday**	1	0	0	**Total Resources**	181	147	59

Table 23 work schedule pattern legend

Work schedule pattern	**Explanation**
Always available	24 hours a day, 7 days a week
01 – 09 (mon-fri, sun)	Monday to Friday, and Sunday, 01.00 – 09.00 (8 hours)
02 – 10 everyday	Monday to Sunday, 02.00 – 10.00 (8 hours)
02 – 12 everyday	Monday to Sunday, 02.00 – 12.00 (10 hours)
02 – 17 (mo-sat)	Monday to Saturday, 02.00 – 17.00 (15 hours)

Work schedule pattern	Explanation
04 – 18 (mo-fri) 04- 15 sat	Monday to Friday, 04.00 – 18.00 (14 hours), Saturday 04.00 – 15.00 (11 hours)
06 – 18 everyday	Monday to Sunday, 06.00 – 18.00 (12 hours)
09 - 17 (mo-sat)	Monday to Saturday, 09.00 – 17.00 (8 hours)

4.3. Conclusion

We have shown the variants between the three business processes of three different business units in the control flow perspective and the resource perspective. The difference in the control flow perspective starts with the different activities which can be executed within the business units. We have shown that every business process has specific activities which can only be executed in that business unit. From the time property, we presented the variant in processing and waiting time of 20 shared activities and showed that the waiting time has a much bigger contribution to the throughput time compared to the processing time. From the control flow itself, the variants lie in the detail of the sub processes. We found three types of variant: the variants of choice, the variants of activity ordering, and the variants of loop structure. Furthermore, in the resource perspective, we have shown the comparison of the total number of resources in different work schedule patterns within business units. The result of this chapter will be used as the configuration points to develop the configurable process models in the next step.

Chapter 5. Process Variants Optimization

In this chapter, we used the process model variants resulted in Chapter 3 and the configuration points from Chapter 4 to develop the configurable process models as input for PETRA. Using the configurable process models, we investigate the applicability of PETRA within DSM. Moreover, by conducting this investigation, we expect to propose the better process model compared to the original process model for each business unit. In order to validate the proposed process models from PETRA perform better from the original process model, we use the Key Performance Indicators (KPIs) from DSM ARIS PPM that is explained in section 5.1. Next, we explain the optimization scenarios which are presented in 5.2 to 5.5. Finally, we conclude our optimization process in section 5.6.

5.1. Key Performance Indicators (KPIs)

In order to measure our models, some KPIs taken from DSM ARIS PPM are introduced:

a. First time right order: the number of orders without any change activity
b. Throughput time end to end process: the time needed from order intake until customer payment
c. Throughput time from order creation to post goods issue item: the time needed from order intake until goods shipped to customer.

We only use the last two KPIs since the first KPI does not serve the purpose of this study. The first KPI would drive the simulation process to only run some specific activities since some activities have a higher priority than the other activities. Hence, the simulation does not run based on the discovered behaviour from the log but from the priority that we already predefined for each activity.

5.2. Complete process model scenario

In this scenario, we try to combine the complete process models from the three business units into one configurable process model for each business unit. This configurable process model takes the variant in the control flow perspective and the time property of it such that the 20 shared activities has three options of the processing and waiting time. Unfortunately, it results on too many variants which do not fit in the main memory. As a consequence, PETRA could not give any outcomes. Therefore, we conduct several other scenarios to optimize the OTC process for each business unit:

a. Sub process scenario
   Since we cannot optimize the process using the whole process we propose to chunk the process into sub processes and try to find the best model of the sub process.
b. As-Is scenario
   In this scenario, we keep the original control flow and try to configure the waiting time of several activities. We expect that we still can optimize the process without changing the control flow since the waiting time has a much bigger contribution to the throughput time compared to the processing time. Hence, if we omit the 'waiting time' silent transitions from the control flow and simulate the process model, the result of the throughput time is very small compared to the throughput time of the log.
c. Reordering activity scenario
   Reordering activity is one of the BPR heuristic rules that can be used when we want to redesign one business process. This rule is the inspiration for this scenario. We assume that every business unit has a similar goal within one sub process such that we can adopt

the working style from one business unit to another business unit. In this scenario, we use the variant of activity ordering to optimize the process models.

In each scenario, we developed specific configurable process models according to the variants chosen. These configurable process models are analysed and optimized using PETRA. Section 5.3 to 5.5 explain the result of the *sub process* scenario, the *As-Is* scenario, and the *reordering activity* scenario respectively.

5.3. Sub process scenario

In order to find the sub process that has the biggest impact to the throughput time, we conduct an analysis of the throughput time for each sub process. Our analysis result shows that the *customer payment* sub process contributes more than 60% of throughput time as shown in Table 24. This sub process indeed has a big impact if we can optimize it. However, it is nearly impossible to optimize this sub process since the task completion heavily depends on the third-party response. In this case when the customer sends the payment. Hence, we neglect the possibility to optimize this sub process. The other sub processes, such as *sales doc creation* and *post goods issue item*, have a very low contribution to the throughput time such that can be neglected. The rest of the sub processes are the *sales doc changes*, the *delivery-shipment-transfer order*, and the *invoice*. From these three sub processes we choose the top 2 of the throughput time contribution to be optimized: the *sales doc changes* and the *delivery-shipment-transfer order*. Subsequently, for every sub process that we choose, we create a configurable process model for each business unit and run in PETRA to determine the best model. In this scenario, we use the variant in the control flow perspective and the time property when we develop the configurable process models.

Table 24 comparison of sub process throughput time

Sub process	Throughput time BU2 (days)	Throughput time BU1 (days)	Throughput time BU3 (days)
Sales doc creation	0.3251 (0.61%)	0.336 (0.54%)	0.623 (1.01%)
Sales doc changes	2.23 (4.19 %)	6.841 (10.89 %)	5.663 (9.19%)
Delivery-shipment-transfer order	15.012 (28.21%)	9.655 (15.38%)	7.156 (11.61%)
Post goods issue	0.0013 (0.002%)	0.926 (1.47%)	1.459 (2.37%)
Invoice	3.613 (6.79%)	0.915 (1.46 %)	1.995 (3.24%)
Customer payment	32.028 (60.19 %)	44.113 (70.26 %)	44.731 (72.58%)
Total Throughput time	53.2094	62.786	61.627

1. Sales doc changes

Based on the results from Chapter 4, we learn two main differences: different set of possible changes of the order in each business unit and loop structure in the BU3 process model. The loop structure in general will increase the frequency of change activities in this sub process. On the other hand, we learn that every change is an undesirable action in the OTC process. From this fact and the impact of the loop structure, we omit the loop structure from BU3 when we create the configurable process model. Moreover, our decision to omit the loop structure is in line with the 'eliminate the unnecessary task' rule in the BPR heuristic rules. Furthermore, we only consider the activity with sufficient impact to the throughput time when we create the configurable process model. Therefore, we propose these guidelines to pick the activity that will be included in the configurable process model:

- It must have considerably long waiting time due to the fact from Chapter 4 that the waiting time has a bigger impact compared to the processing time.
- The percentage of occurrence must at least 2% out of total record in one or more business units.

Table 25 shows all change activities in the three business units. Activities with yellowish background are not included in the configurable process model based on the guidelines.

Besides the activity, we also consider the waiting time as the configuration point by having activities with two or more options of waiting time distribution. We choose two activities with the biggest gap of waiting times and high occurrence from Table 25, which are: the ‘change requested delivery’ activity and the ‘set overall status credit check’ activity. The ‘change requested delivery’ activity has two options of waiting time distribution and the ‘set overall status credit check’ has three options of waiting time distribution. For the resource perspective, we assume the resources in one business unit belongs to that business units and they cannot work for other business units such that we do not consider resource as a configuration point. We also do not consider the probability of choice as a configuration point since the customer characteristic of one business unit is different from others, such that we could not easily adopt the probability choice from one business unit to another. Therefore, we could not have one configurable process model for this sub process because several perspectives of the process model could not be combined. Consequently, we develop one configurable process model for each business unit. This configurable process model is developed with different value of choice probability, the resources involved, and inter-arrival rate.

Table 25 comparison of activity waiting time and occurrence in sub process sales document changes

Activity	BU3 (WT day)	Total occurrence / total records (%)	BU1 (WT day)	Total occurrence / total records (%)	BU2 (WT day)	Total occurrence / total records (%)
Change net price			5.71	6.88%		
Change order quantity	4.59	6.49%	6.65	5.10%	4.86	3.92%
Change requested delivery date	2.74	13.62%	8.65	16.62%		
Change route	2.92	6.31%	1.63	2.95%		
Change schedule line date	3.34	15.55%	5.42	9.26%	7.24	3.85%
Change shipping conditions	5.04	1.25%				
Create pro forma order	1.58	3.89%				
Release delivery block	0.92	2.74%	2.39	6.46%		
Release overall status of credit check	1.32	21.18%	3.72	12.01%	2.66	10.03%
Release status alternative	0.56	30.38%				
Release status availability	1.48	7.61%	1.39	12.76%	0.06	1.61%
Release status lead-time	0.05	0.77%	0.54	2.65%		
Release status OTC delivery manually	2.24	23.22%				
Release status OTC med	0.37	7.96%	0.44	9.07%	0.21	2.77%
Set overall status of credit check	6.96	7.57%	14.6	4.37%	3.91	1.62%
Set status alternative	1.66	21.10%	9.1	5.09%		
set status availability	1.94	7.30%	2.76	4.41%		

Activity	**BU3 (WT day)**	**Total occurrence / total records (%)**	**BU1 (WT day)**	**Total occurrence / total records (%)**	**BU2 (WT day)**	**Total occurrence / total records (%)**
Set status lead-time problem			1.26	2.36%		
Set status OTC Allowed	0.003	91.47%	0.015	83.62%	0.001	95.84%
Set status OTC delivery manually	1.61	12.24%				
Set status OTC Med value	0.09	8.32%	0.876	9.06%	0.12	2.77%
Update order reason					0.12	0.63%

= the neglected activity.

Given the process models of the *sales doc changes* sub process for every business unit and the configuration points, we create manually the configurable process model for each business unit. The configurable process trees can be seen in Figure 58. For every business unit, 2048 variants have been analysed using PETRA.

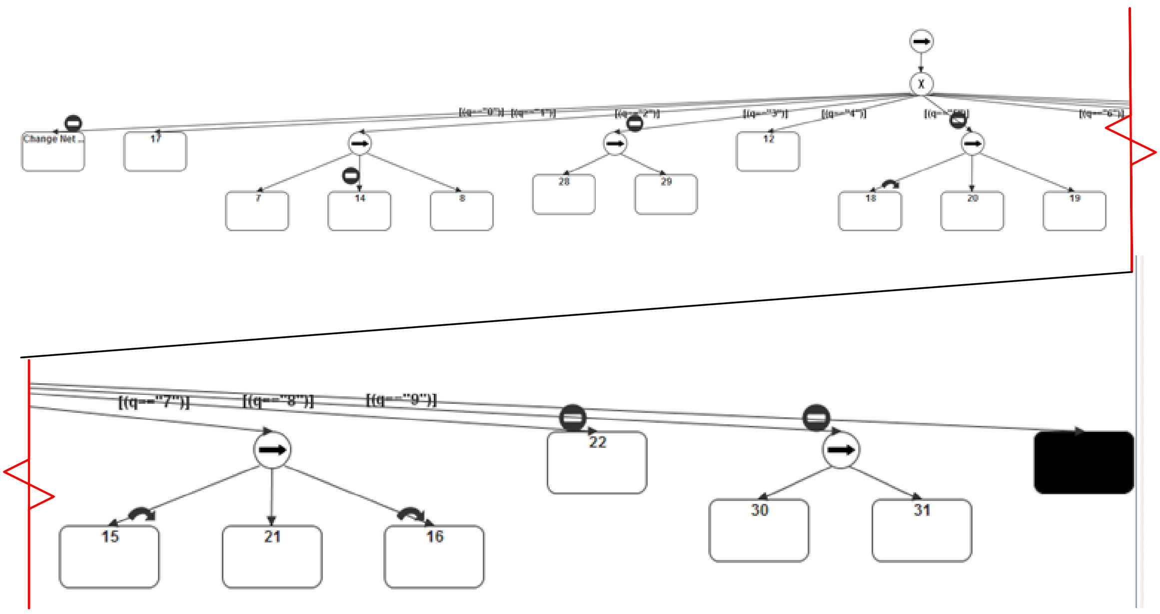


Figure 58 configurable process model of sub process sales document changes

Result

In general, PETRA yielded process models with shorter throughput time compared to original one. However, this is achieved by allowing less occurrence of the change activities. Another result, PETRA gives the shorter waiting time for two activities with configurable waiting time. Table 26 gives an overview of the result for every business unit and Figure 59 shows the best process tree for every business unit

Table 26 optimization result of sub process sales document changes

BU3	• Best throughput time 3.32 days with CI 95% = 0.017 days compared to 5.663 days • No loop structure • Does not have the 'set-release OTC deliver manually' activity, the 'set-release status availability problem' activity, the 'release delivery block' activity, and the 'set status lead time problem' activity. • Set overall status avg. waiting time = 113.513 hours (taken from BU2)
BU1	• Best throughput time 2.61 days with CI 95 % = 0.014 days compared to 6.841 days • Does not have the 'change net price' activity, the 'release delivery block' activity, the 'release status avail' activity, the 'set status avail' activity, and the 'add create pro forma' activity. • Set overall status avg. waiting time = 117.816 hours (taken from BU2), change req. delivery avg. time= 58.117 hours (taken from BU3).
BU2	• Best throughput time = 0.418 days with CI 95% = 0.005 days compared to 2.23 days • Does not have the 'release status availability' activity • Change req. delivery avg. waiting time = 55.693 hours (took BU3)

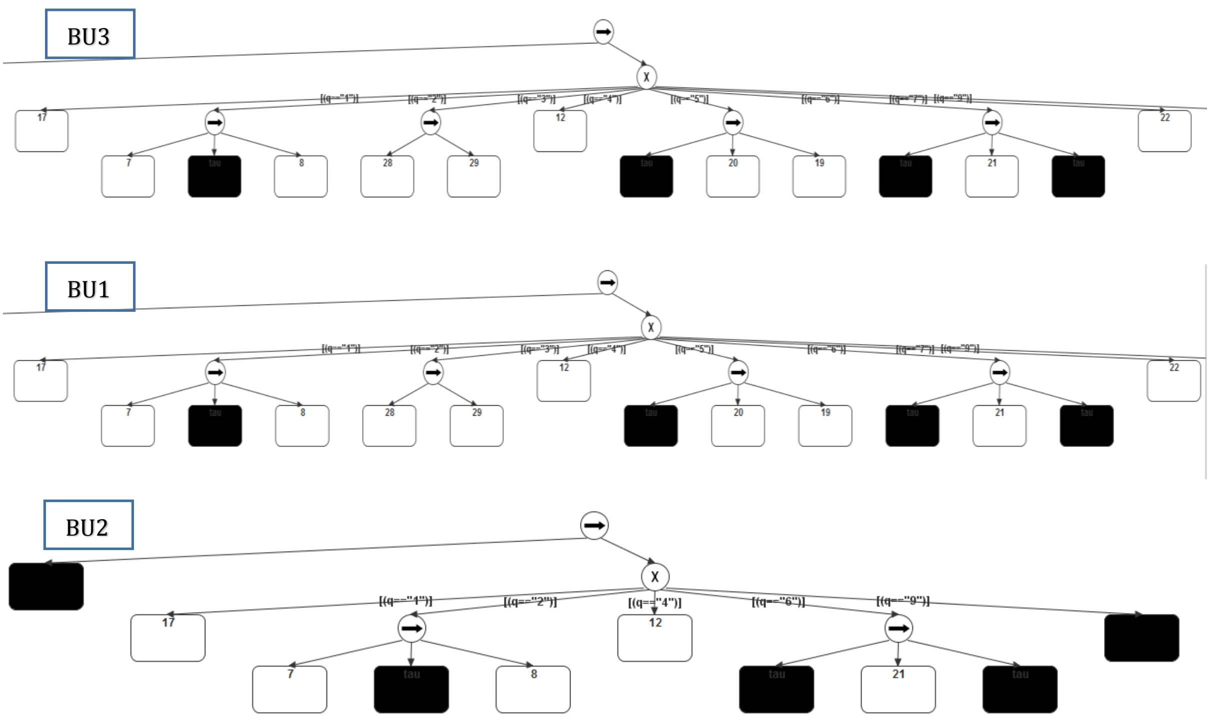


Figure 59 optimized control flow of sub process sales document changes. (top) BU3, (middle) BU1, (bottom) BU2

5.3.2. Delivery – Shipment – Transfer order

The main difference in this sub process is the loop structure in BU2 and BU1. BU2 has a loop structure in the whole sub process while BU1 has a loop structure only for the 'create transfer order' activity. Furthermore, BU3 does not have a loop structure at all. It can be shown by total occurrences of the 'create outbound delivery' activity and the 'create transfer order' activity in Table 27. The total occurrence of the 'create transfer order' activity in the BU1 event log is twice the number of the traces such that it justifies the loop structure for the 'create transfer order' activity in the BU1 process model. Meanwhile, the total occurrence of the 'create outbound delivery' activity in the BU2 event log is almost 1.5 times of the number of the traces. It justifies the loop structure in the BU2 process model. The loop structure in this sub process indicates the rework pattern. Therefore, we tried to optimize this sub process by omitting the loop structure for both business units. In contrast with the sales item changes, we include all possible activities from three business units and does not configure the waiting time. With the same reason as the *sales doc changes* sub process, we create one configurable process model for each business unit with different values of choice probability, the resources involved, and inter-arrival rate. The configurable process trees can be seen in Figure 60.

Table 27 comparison of occurrence and total order for create outbound delivery and cerate transfer order activity

Activity	BU1– total activities	BU1– total orders	BU2 – total activities	BU2 – total orders	BU3 – total activities	BU3 – total orders
Create outbound del	28,868 (1.1 times per trace)	25,256	16,197 (1.5 times per trace)	10,872	24,733 (0.9 times per trace)	27,466
Create transfer order	48,671 (1.9 times per trace)	25,256	13,210 (1.2 times per trace)	10,872	29,860 (1.1 times per trace)	27,466

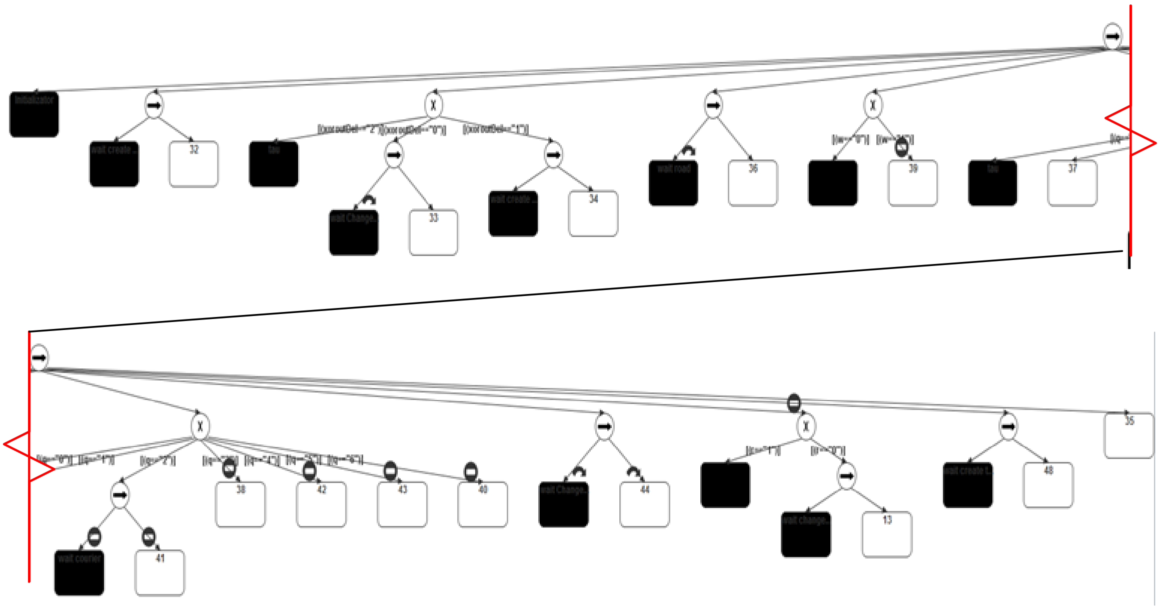


Figure 60 configurable process models of sub process delivery-shipment-transfer order

In total, 2048 variants have been analyzed using PETRA and PETRA yielded process models with better throughput time compared to the original. For BU1, the best result is 6.49 days (with 0.011 days as 95 % confidence interval) compared to 9.655 days in the original model, but it does not have the loop structure and the 'change gross weight' activity. For BU3, the best result is also 6.49 days (with 0.014 days as 95% confidence interval) compared to 7.156 days in the original model, but it does not have the 'courier' activity. Meanwhile, for BU2, the best result is 3.36 days (with 0.007 days as 95% confidence interval) compared to 15.012 days in the original model, but the process model does not have the loop structure, the 'change net price' activity, the 'intermodal' activity, and the 'change material staging' activity anymore. Figure 61 to 63 show the optimized process tree of sub process delivery-shipment-transfer order for each business unit.

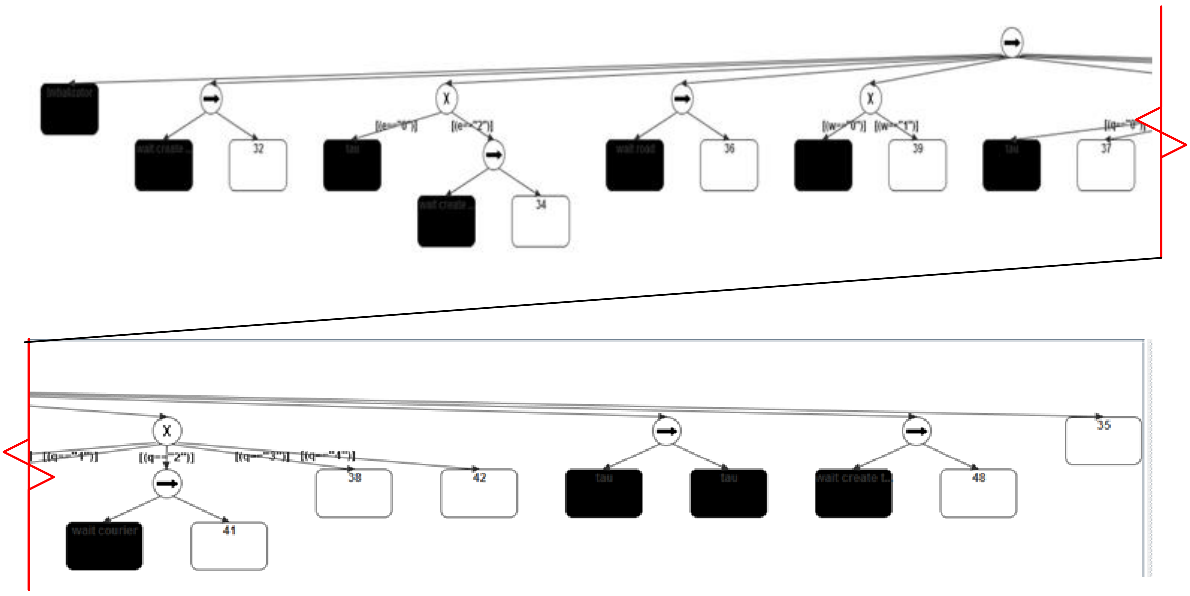


Figure 61 BU1's optimized control flow of sub process delivery-shipment-transfer order.

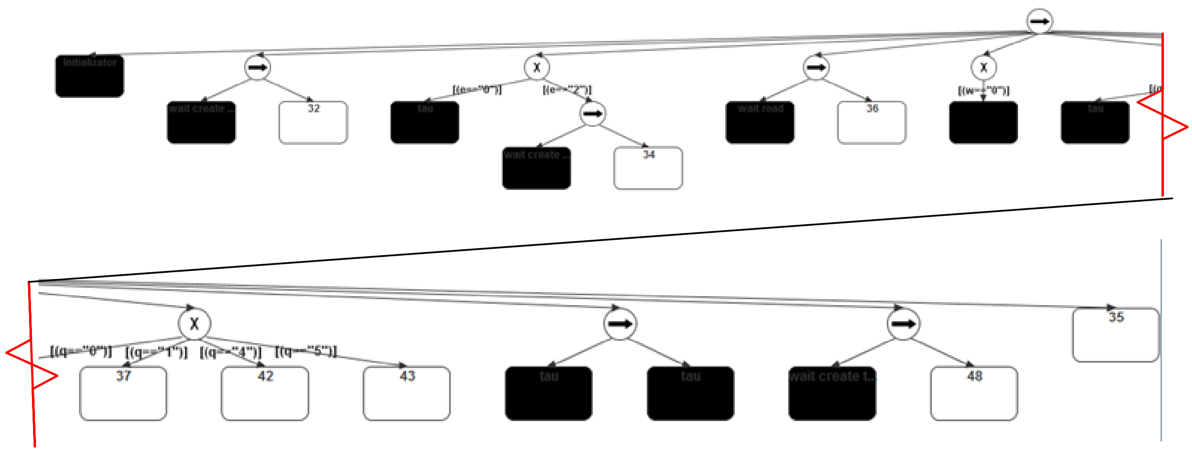


Figure 62 BU2's optimized control flow of sub process delivery-shipment-transfer order.

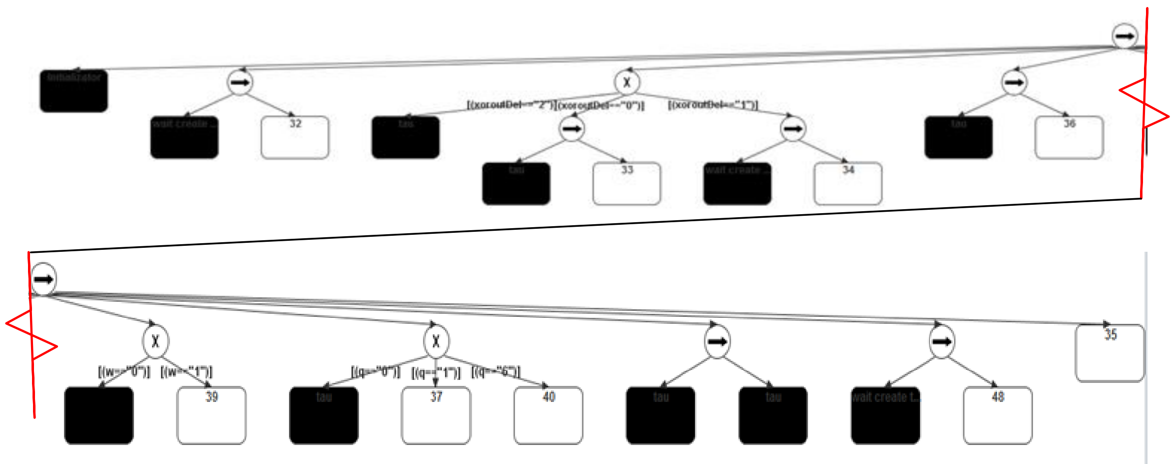


Figure 63 BU3's optimized control flow of sub process delivery-shipment-transfer order.

To conclude, this sub process scenario can yield a process model with better throughput time in sub processes which eventually will also reduce the throughput time for the whole process. Moreover, it can be used to simplify the optimization process and to gain more specific insight from one sub process. However, since we combine the process models from three variants, PETRA tends to yield results with the simplest process models out of the three variants. The result also shows that the activity with low probability will be omitted in the best solution and the lowest waiting time setting will be used. Nevertheless, this scenario has one main drawback which is the business unit losses some of its flexibility to handle the process since some activities are omitted from the process model, especially activities related to changing the document either sales document or delivery document.

5.4. As-Is scenario

We learned from the previous chapter that the waiting time has the biggest portion of the throughput time, hence we can use this finding to optimize the business process. In contrast to the sub process scenario, we keep the structure of the control flow of each business unit as it is to maintain the behaviour of the process. Subsequently, we analyse the waiting time for 20 shared activities among business units and choose the top 3 activities with high waiting times and a big gap between the maximum and minimum waiting time: the ‘Set overall status of credit checks’ activity, the ‘change requested delivery date’ activity, and the ‘create transfer order’ activity. A big gap amongst the waiting time of an activity indicates there are different approaches to cope with the waiting time of the specific activity between business units. We deliberately excluded the ‘create customer payment’ activity since it is almost impossible to optimize the waiting time for customer payment and also it is more dependant to the customer rather to the company. Table 28 shows the list of activities in consideration. Furthermore, we create a configurable process model and configure the waiting time for these activities with three different time options such that every business unit has 27 variants of business process.

Table 28 top activities with high waiting time

Activity name	BU1 (days)	BU3 (days)	BU2 (days)
create Customer payment	43.91	43.78	31.90
Set Overall status of credit checks	14.15	6.92	3.92
Change Requested delivery date	8.66	2.75	4.06
create transfer order	0.91	2.01	5.07
create Outbound delivery	5.88	4.18	1.90
Release Overall status of credit checks	3.71	1.33	2.66
create accounting process	0.51	0.37	2.56
Change Order quantity in sales units	6.68	4.61	4.87
create Invoice	0.48	0.62	2.20

= activity with longest waiting time, not chosen. = activity chosen for this scenario.

From all 27 variants, we have that the best model of every business unit adopts the minimum waiting time. Therefore, every business unit adopts waiting time of the ‘set overall status of credit checks’ activity (SOSCC) from BU2, the waiting time of the ‘change requested delivery date’ activity (CRDD) from BU3, and the waiting time of the ‘create transfer order’ activity (CTO) from BU1. The throughput time reduction of this scenario is not as much as the *sub process* scenario as can be seen in Table 29. Table 29 presents the throughput time of top three process model for each business unit and also the chosen waiting time for each activity. Another interesting finding is that every business unit always uses the waiting time of the ‘create transfer order’ from BU1 in the top three process model. This is due to the fact that the ‘create transfer order’ activity is a must-do activity in the business process. Therefore, even though the difference between the minimum and the second minimum in waiting time of the 'create transfer order' activity is only around 1 day, it costs the throughput time also 1 day longer as shown in Table 30.

Table 29 top optimized model of as-is approach.

BU	Rank	Throughput time (days)	Std. Dev.	CI 95 %	SOSCC	CRDD	CTO
BU1	1	60.28	0.13	0.085	BU2	BU3	BU1
	2	60.38	0.095	0.064	BU3	BU3	BU1
	3	60.5	0.16	0.11	BU2	BU2	BU1
BU2	1	47.88	0.2	0.088	BU2	BU3	BU1
	2	47.95	0.19	0.11	BU3	BU3	BU1
	3	48.11	0.15	0.099	BU1	BU3	BU1
BU3	1	60.12	0.09	0.061	BU2	BU3	BU1
	2	60.33	0.16	0.11	BU2	BU2	BU1
	3	60.54	0.088	0.059	BU3	BU2	BU1

Table 30 optimized model with BU3 waiting time for change transfer order

BU	Rank	Throughput time (days)	Std. Dev.	CI 95 %	SOSCC	CRDD	CTO
BU1	9	62.37	0.11	0.072	BU2	BU3	BU3
BU2	10	49.25	0.18	0.13	BU2	BU3	BU3
BU3	6	61.17	0.12	0.083	BU2	BU3	BU3

Even though this approach can reduce the throughput time, we also notice that the waiting time happens between two related activities. In order to reduce the waiting time of an activity, we have to consider that activity and the preceding activity. Therefore, we conduct a preliminary analysis of the predecessor activity of the top three waiting time activities in this scenario. Table 31 shows the top 2 preceding activities that contribute the highest waiting time for each top waiting time activity. For example, if the 'set overall status of credit check' activity is preceded with the 'set status OTC allowed' activity, the average waiting time in between of these two activities are 18.1 days. For the 'set overall status of credit check' activity and the 'change requested delivery date' activity, every business unit has the same preceding activities that contribute high waiting time. On average, BU2 has the least waiting time for the 'set overall status of credit check' activity and BU3 has the least waiting time for the 'change requested delivery date' activity. Using this preliminary analysis, a further investigation can be conducted, e.g. learn the working style of BU2 to handle the waiting time between the 'set overall status of credit check' activity and the 'release overall status of credit check' activity and investigate whether it is applicable to the other business units since it has big difference of waiting time between business units for this pair of activity.

Table 31 top preceding activities for three activities with configurable waiting time

Activity	Preceding activity	WT BU1 (days)	WT BU2 (days)	WT BU3 (days)
Set overall status of credit check	Set status OTC allowed	18.1	3.88	6.11
	Release overall status of credit check	12.09	1.01	6.94
Change requested delivery date	Set status OTC allowed	10.78	8.54	1.89
	Change requested delivery date	8.77	5.27	6.75
Create transfer order	Road shipment	3.67		2.09
	Pick up (ex work) shipment		8.79	2.83
	Rail shipment		16.31	

5.5. Reordering activities scenario

Based on the result from Chapter 4 and considering the input from the process expert, we found at least two situations where we can use this scenario for optimization. In contrast to previous scenarios, we only change the ordering of the activities while we still maintain all the original activities and waiting time in between.

5.5.1. BU2 delivery-shipment-transfer order sub process.

Even though the loop structure in this sub process depicts the rework pattern, we cannot be sure that the loop structure in this sub process is an undesirable condition as in the *sales document changes* sub process. Therefore, it becomes interesting to take another approach when optimizing this sub process. This scenario applies specifically in BU2 because BU2 has the longest throughput time for this sub process. From three business units, BU1 and BU3 already have a clear ordering between delivery, shipment, and transfer order. On the other hand, BU2 only has a clear ordering between delivery and shipment where the delivery is a predecessor activity of the shipment. In our investigation, we found three parts that cause confusion for the activity ordering in BU2, which are: the 'change current qty. in delivery shipment' activity, the 'create transfer order' activity, and the 'delivery-shipment' sub process. BU3 allows the 'change current qty. in delivery item' activity to be executed after the 'create transfer order' activity while BU1 allows the other way around. In contrast with the other two business units, BU2 does not have any clear ordering. After investigating the situation, we found that BU2 has 12 different possible orderings. It includes no loop/loop condition, do after or before the (delivery-shipment) activity condition, and the parallel execution of two activities: the 'change current qty. in delivery shipment' activity and the 'create transfer order' activity. The parallel process can be applied because BU1 and BU3 give different ordering between these two activities. From this fact, we can conclude that there is no dependency relationship between these two activities and we can execute these activities in parallel. Table 32 shows the average throughput time for every possible ordering. "in back" denotes the tasks are executed after the *delivery-shipment* sub process while "in front" denotes the other way around. From the result, we can conclude several things:

a. Having loop/no loop structure gives a big impact in term of throughput time
b. The parallel processing can slightly reduce the throughput time up to 0.83 days compared to the sequential processing. For example, the ordering nr.6 is the parallel process and the ordering nr. 2 and 10 are the processes with the sequential ordering of the ordering nr.5. The throughput time of the ordering nr.2 is 52.89 days, and the ordering nr.10's throughput time is 52.96 days, while the ordering nr.6's throughput time is 52.13 days.
c. The ordering between the 'change current qty. in delivery item' activity and the 'create transfer order' activity has no significant effect for the throughput time.

d. Putting the 'create transfer order' activity as early as possible position or as late as possible position in the process also does not provide a significant effect for throughput time.
e. Nevertheless, in BU2's situation, this scenario can have similar effect even better compared to the *as-is* scenario if it has parallel process and no loop structure.

Table 32 throughput time result for all possibility current qty-transfer order ordering

Nr.	Ordering	Throughput time (days)	CI 95 % (days)
1	Current qty - transfer order (in back) no loop	47.59	0.118
2	Current qty - transfer order (in back) loop	52.89	0.083
3	Current qty – transfer order (in front) no loop	47.63	0.095
4	Current qty – transfer order (in front) loop	52.83	0.135
5	current qty AND-join transfer order (in back) no loop	47.18	0.107
6	current qty AND-join transfer order (in back) loop	52.13	0.143
7	current qty AND-join transfer order (in front) no loop	47.19	0.106
8	current qty AND-join transfer order (in front) loop	52.18	0.154
9	transfer order - current qty (in back) no loop	47.60	0.092
10	transfer order - current qty (in back) loop	52.96	0.127
11	transfer order - current qty (in front) no loop	47.66	0.121
12	transfer order - current qty (in front) loop	52.91	0.130

5.5.2. BU1 and BU3 change requested delivery date activity

From a discussion with a business expert and the impact analysis in the Chapter 4, we conclude that one of the biggest impact activities is the 'change requested delivery date' activity. Therefore, it becomes interesting to know the effect on the throughput time if we have this change as early as possible and as late as possible in the OTC process. We configure BU1 and BU3 control flow to this new ordering and have 2 variants for each business unit. Table 33 shows the result of this reordering scenario. In BU1, reordering the activity either as late or as early as possible can reduce the throughput time as much as 0.2 days compared to the original process models. In BU3, having the 'change requested delivery date' activity as early as possible can reduce the throughput time as much as 0.5 days compared to the original process model. Meanwhile, putting the activity as late as possible does not have a significant impact to the throughput time compared to the original model.

Table 33 throughput time for all possibilities of change requested delivery date ordering

Business unit	As early as possible	CI 95%	As late as possible	CI 95%	Original	CI 95%
BU1	62.28	0.060	62.29	0.093	62.46	0.042
BU3	61.09	0.070	61.42	0.071	61.41	0.032

5.6. Conclusion

In this chapter, we investigated the applicability of PETRA by conducting four different scenarios to optimize business process using PETRA. The first scenario could not give any result since it produced too many variants such that it did not fit in main memory. As a consequence, PETRA did not give any result. Contrary to the first scenario, the other three scenarios give a better model for every business unit. In the *sub process* scenario, we tried to minimize the rework pattern and unnecessary task by hiding and blocking activities with low probability or activities that only exists in one or two business unit(s) and deleting the loop structure. The result is significant and even for BU1 it can reduce the throughput time by almost 4.3 days for the *sales doc changes* sub process alone. In total, from the two sub processes we optimized, BU1's throughput time can be reduced by 7.4 days, BU2 can be reduced by 13.4 days, and BU3 can be reduced by 4 days. However, this approach has a drawback that the model becomes less flexible. For example, BU2 could not have the 'change net price' activity anymore in delivery-shipment-transfer order sub process. In fact, this change is a common activity in BU2 due to the nature of the product that depends on the oil price.

The *as-is* scenario is conducted based on the fact that the waiting time has a major portion of the throughput time and we want to maintain the behaviour of the process for each business unit. The motivation for this approach is if we can adopt the waiting time from another business unit, we can also reduce the throughput time. We picked three different activities that have considerably long waiting times and a big gap between the minimum value and maximum value among three business units. From the result, we can conclude that this scenario also can produce an optimal business process. In total, using this approach, BU1's throughput time can be reduced by 2.2 days, BU2 can be reduced by 4.48 days, and BU2 can be reduced by 1.3 days. However, the reduction of the throughput time is not as big as the *sub process* scenario. For example, in BU2, the reduction is only around 4.48 days compared to 13.4 days reduction of BU2 in the *sub process* scenario. Nevertheless, this scenario can overcome the drawback from the *sub process* scenario because it maintains the original structure of the control flow such that it has more flexibility compared to the *sub process* scenario. In a real situation, we need further investigation to ensure whether the waiting time of one business unit is applicable to other business units. To help the investigation, we provide the preliminary investigation that gives the top 2 preceding activities that contributes to a large waiting time.

Our last scenario is inspired from one of the BPR rules: reordering. We reordered several activities in one business unit by adopting other business unit behaviour in two cases. We showed that even though the reordering applied, the throughput time cannot be reduced significantly in most cases. To conclude this chapter, from the business aspect, we have shown by exploiting the waiting time, limiting the activities, omitting the loop structure, or reordering the activities, the business units can gain some reduction in throughput time even though the applicability of this optimal business process needs further investigation. From the scientific aspects, we have shown that PETRA is applicable in DSM case to some extent. Moreover, we also showed PETRA's capability to do what-if analysis by doing simulation for every business process possibility. We also already presented the result of PETRA and gave several options of the business process that performs better compared to the current one for business units.

Chapter 6. Related work

The related work is divided into three parts: Business Process Management (BPM) and process mining, configurable process models, and case study of process mining application.

BPM and process mining. The quest of improving business processes has been one of the main issues in the field of industrial engineering. In 1990, the usage of information systems was promoted as a redesign tool that leads to improving the business process [8]. Since then, research on the usage of information systems as a tool for managing business processes becomes very popular. It continues to grow and becomes what we know today as Business Process Management (BPM). BPM is one of the foundations of process mining research besides data mining [9]. Afterwards, process mining emerged as an active research topic and many tools have been developed for process discovery, process analysis, and process enhancement or optimization. In the process discovery, many researches have been conducted to find the best algorithm to find the control flow from the event log, such as α algorithm [16], the heuristic miner [17], the fuzzy miner[18], the inductive miner[4, 5], and the ETM miner[3]. Nevertheless, not all mining algorithms guarantee the soundness of the resulted process model as the soundness is one of the main criteria of a good control flow [3].

Configurable process models. Configurable process models are a young and new research area. This research area is preceded by awareness of people that certain variations exist in one business process. In [29], a method is proposed to combine variants of business process into one reference model in the healthcare domain. Nevertheless, the reference model is different with the configurable process models because the reference model are based on guidelines and evidence and the configurable process models based on the family business process [29]. In [19], a questionnaire based approach is introduced to capture the variability in a high level of abstraction. This approach, in combination with the functional requirements, yields the configurable process models. In [13], the Provop framework is proposed to capture the variability of business process into configurable process models by applying a context-based configuration. However, these two reports do not have a formal definition of configurable process models they used such that it makes difficult to exploit the variants automatically in a tool.

Case study of process mining. Many organizations have applied process mining within their business process as reported in [9]. Still, there are only limited reports on a complete process mining project. A complete process mining project includes all type of process mining task: process discovery, process analysis, and process enhancement or optimization [2]. In [11] and [12], we can learn two different implementations of a complete process mining project in two different business areas: healthcare and government. However, the optimizing part of the project does not exploit the variants of business process. Another report in [14] shows the applicability of variants of business process as an input for the optimization process. However, the tools developed for the project does not provide the complete analytical function to determine the best variants.

Chapter 7. Conclusion

This chapter starts with the summary and conclusion in 7.1 and continues with future work in 7.2

7.1. Summary and conclusion

In this study, we conducted a process model discovery from four perspectives. We presented the step to discover of the work schedule of the resource perspective by taking the mode value of the first and last hour recorded activity of the resources using the event log from the BPM system of DSM. Using the same event log, we also showed how to discover the task responsibility of the resource perspective, and the step to discover the case arrival property of the environment perspective. We also conducted the process mining task with two algorithms for process mining: ETM miner and Inductive miner-infrequent to discover the control flow perspective of the process model. Due to the high variability of the process we only took part of the log as the input file instead of taking a whole log as the input file. Furthermore, we proposed several steps to combine the result of the two algorithms. We showed that the combination of the control flow can maintain a high replay fitness value while also have the acceptable precision value. Furthermore, we also presented the step to discover of the time property of the control flow perspective by sorting the event log in certain ways. In order to validate our process model, we presented the expert opinions and conducted the simulation to compare the throughput time of the process models from the event log and from the simulation result. By doing the process mining task, combining the result of the process mining task, and conducting step to discover the other perspectives, we already answered the first sub research question: "*How can we discover the process models variants within DSM?*"

To answer the second sub research question, we compared and analysed the variants in two perspectives: the control flow perspective and the resource perspective. Within the control flow perspective, we also conducted the comparison and analysis for the time properties. In general, every business unit has the variant of processing and waiting time of each activity. Nevertheless, we found the interesting findings that the waiting time has a much bigger impact to the throughput time compared to the processing time. In the control flow itself, we conducted our comparison and analysis into several sub processes. We found the different way of working in each sub process for each business unit. Furthermore, we found three types of variant: the variants of choice, the variants of activity ordering, and the variants of loop structure. The variants of choice is the difference set of activities can be executed after a specific activity in the business process. In the resource perspective, we presented the comparison of the work schedule pattern of every business unit in term of number of resources. In total, we found 53 different types of work schedule pattern that belongs to 387 resources from all business units. This results are the answer to our second sub research question: "*What are the variation point of the process model variants within DSM?*"

To answer the third sub research question, we conducted four different scenarios that using the resulted process model and the resulted configurations point to develop the configurable process model. The configurable process models are the main input of PETRA. Each scenario exploits the variants that we found beforehand. The first scenario exploited the variants of choice, the variants of loop structure, the variant of activity ordering, and the variants of time property; and applied them to the whole process. Meanwhile, the second scenario exploited the variants of choice, the variants of loop structure, and the variants of time property; and applied them to the sub processes. Furthermore, the third scenario only exploited the variants of the waiting time. Finally, the last scenario exploited the variants of activity ordering. The configurable process models that resulted from the four scenarios are the answer for the third sub research question: "*How can the generated process models variants be used as input for PETRA?*"

Using the result of the sub research questions, we investigated the applicability of PETRA in DSM case. We showed the applicability of PETRA by conducting the optimization process using four scenarios. Three out of the four scenarios could result the better model compared to the original model while the first scenario could not produce any result due to memory limitation. We presented that the best model can be achieved by limiting the choice and omitting the loop structure in the second scenario. The omitted activities are usually the activities with low probability and/or long waiting time. The choice limitation and loop elimination lead to a significant reduction of the throughput time for every business unit. BU1's throughput time can be reduced by 7.4 days, BU2 can be reduced by 13.4 days, and BU3 can be reduced by 4 days. However, we concluded that this scenario has its drawback which is the business unit losses its flexibility to handle many possibility behaviours in future. Furthermore, the third scenario also yielded the process model with less throughput time compared to the original process model. However, the reduction of throughput time is less than the second scenario. BU1's throughput time can be reduced by 2.2 days, BU2 can be reduced by 4.48 days, and BU2 can be reduced by 1.3 days. Nevertheless, this scenario can overcome the drawback of the second scenario since it maintains the original control flow. The fourth scenario is conducted in two sub processes: *sales item changes* and *delivery-shipment-transfer order*. This scenario result showed that the reordering activity in the *sales item changes* sub process could reduce the throughput time with at most 0.83 days while reordering activity in the *delivery-shipment-transfer order* sub process does not have a significant effect on the throughput time.

In conclusion, our study confirmed that PETRA might be applicable to other domain outside the municipality. However, to be fully applied in DSM case, we need to reduce the space of the possible model by selecting some of the configuration points or dividing the complete process into several sub processes. Nevertheless, we validated the capability of PETRA to find the best process model by analysing the configurable process models and adding the process model with the simulation properties. We also showed the scalability of PETRA which analysed 2,048 variants of the possible process model in the scenarios. The results show PETRA could give the model that has a better score on their performance indicator. Moreover, this model could show which part of the business process can be improved in order to get less throughput time. Therefore, the solution we offered in this study can be used for further investigation or to trigger the discussion with the business owner in order to decide the better process model. Thus, it can be concluded that PETRA can be applied in the multinational company.

7.2. Future work

We also noted several future works to improve our results that described in the following sub sections. This section is divided into four types of improvements: on discovering the control flow, on discovering the process model, on optimizing business process, and on PETRA as a tool itself.

7.2.1. Discovering the control flow

We already showed that our input files only contained a part of the complete log. This is due to the time limitation and algorithm limitation which resulted the control flow that is hard to read and analyse. During the last part of the study, we found the paper from Leemans, et al. that proposed a new process mining framework that extends the IMi miner which can handle big amounts of data in the log [26]. This new opportunity will become a great improvement in our study if we can use the complete log as the input file such that no hidden variants are left out of the control flow. We could compare the result of this new framework with current control flow and analyse the closeness between these two results.

7.2.2. Discovering the process model

In this study, we choose the property of each perspective such that it can be developed into a simulatable process model. Nevertheless, there are many other properties that can be discovered to make the process model closer to the real life situation. For example, we assume the resource always has the same working speed along her working hour. However, we also learn from [24] that the arousal level of the resource can turn up and down along the working hour. Therefore, it will be interesting to discover and analyse this property in future work because we deal with considerably large amount of resource.

7.2.3. Optimizing business process

In the last part of this study, we deliberately exclude the variants on resource due to the complexity that can be produced by configuring this variant. We assumed the resource is always available during the work schedule and every activity has a fixed amount of responsible resource. It will be interesting if we can configure the work schedule and amount of responsible resource such that the optimization result can provide a proposal on the optimal amount of responsible resource for every activity and optimal working schedule for every resource.

7.2.4. PETRA

According to [10], there are four dimensions of the qualitative evaluation of business process: time, quality, flexibility, and cost. From these four dimensions, only two can be easily measured quantitatively: cost and time. Currently, PETRA can handle configuration on resource property, time property, and control flow perspective. These configuration points yield the simulation result on time property. Even though we can derive the cost from the time property, it would become great improvement if PETRA also can include cost as an additional property of task or resource. Hence, the optimization can give more direct business value to the company.

Another improvement can be made is related to the search space. Currently, PETRA exhaustedly tries all possibilities of solution within the search space. Whereas, there might be a number of solutions similar to each other. If we can define the similarity formula for two process models, we can quickly eliminate the similar candidates of process model before PETRA runs the simulation such that we can reduce the memory needs and also the simulation run time.

Bibliography

[1] Schunselaar, D.M.M., et al. "Petra: A tool for analysing a process family."*International Workshop on Petri Nets and Software Engineering (PNSE'14)*. No. 1160.

[2] Van Der Aalst, W.M.P., et al. "Process mining manifesto." *Business process management workshops*. Springer Berlin Heidelberg, 2012.

[3] Buijs, Joos CAM, Boudewijn F. van Dongen, and W.M.P. van der Aalst. "On the role of fitness, precision, generalization and simplicity in process discovery." *On the Move to Meaningful Internet Systems: OTM 2012*. Springer Berlin Heidelberg, 2012. 305-322.

[4] Leemans, Sander JJ, Dirk Fahland, and W.M.P. van der Aalst. "Discovering block-structured process models from event logs-a constructive approach."*Application and Theory of Petri Nets and Concurrency*. Springer Berlin Heidelberg, 2013. 311-329.

[5] Leemans, Sander JJ, Dirk Fahland, and W.M.P. van der Aalst. "Discovering block-structured process models from event logs containing infrequent behaviour." *Business Process Management Workshops*. Springer International Publishing, 2014.

[6] Rozinat, Anne, and W.M.P. van der Aalst. "Conformance checking of processes based on monitoring real behavior." *Information Systems* 33.1 (2008): 64-95.

[7] Van der Aalst, W.M.P., Arya Adriansyah, and Boudewijn F. van Dongen. "Replaying history on process models for conformance checking and performance analysis." *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 2.2 (2012): 182-192.

[8] Davenport, Thomas H., and James E. Short. "Information technology and business process redesign." *Operations management: critical perspectives on business and management* 1 (2003): 97.

[9] Van Der Aalst, W.M.P., *Process mining: discovery, conformance and enhancement of business processes*. Springer Science & Business Media, 2011.

[10] Reijers, Hajo A., and S. Liman Mansar. "Best practices in business process redesign: an overview and qualitative evaluation of successful redesign heuristics." *Omega* 33.4 (2005): 283-306.

[11] Rebuge, Álvaro, and Diogo R. Ferreira. "Business process analysis in healthcare environments: A methodology based on process mining." *Information Systems* 37.2 (2012): 99-116.

[12] van der Aalst, W.M.P., et al. "Business process mining: An industrial application." *Information Systems* 32.5 (2007): 713-732.

[13] Hallerbach, Alena, Thomas Bauer, and Manfred Reichert. "Capturing variability in business process models: the Provop approach." *Journal of Software Maintenance and Evolution: Research and Practice* 22.6-7 (2010): 519-546.

[14] Taylor, Paul, Marcello Leida, and Basim Majeed. "Case study in process mining in a multinational enterprise." *Data-driven process discovery and analysis*. Springer Berlin Heidelberg, 2012. 134-153.

[15] Van Der Aalst, W.M.P., Hajo A. Reijers, and Minseok Song. "Discovering social networks from event logs." *Computer Supported Cooperative Work (CSCW)*14.6 (2005): 549-593.

[16] Van der Aalst, W.M.P., Ton Weijters, and Laura Maruster. "Workflow mining: Discovering process models from event logs." *Knowledge and Data Engineering, IEEE Transactions on* 16.9 (2004): 1128-1142.

[17] Weijters, A. J. M. M., W.M.P. van Der Aalst, and AK Alves De Medeiros. "Process mining with the heuristics miner-algorithm." *Technische Universiteit Eindhoven, Tech. Rep. WP* 166 (2006): 1-34.

[18] Günther, Christian W., and W.M.P. Van Der Aalst. "Fuzzy mining–adaptive process simplification based on multi-perspective metrics." *Business Process Management*. Springer Berlin Heidelberg, 2007. 328-343.

[19] La Rosa, Marcello, et al. "Questionnaire-driven configuration of reference process models." *Advanced Information Systems Engineering*. Springer Berlin Heidelberg, 2007.

[20] Rosemann, Michael, and W.M.P. van der Aalst. "A configurable reference modelling language." *Information Systems* 32.1 (2007): 1-23.

[21] La Rosa, Marcello, et al. "Configurable multi-perspective business process models." *Information Systems* 36.2 (2011): 313-340.

[22] Gottschalk, Florian. *Configurable process models*. Diss. Eindhoven University of Technology, 2009.

[23] Schunselaar, D. M. M., et al. "Creating sound and reversible configurable process models using CoSeNets." *Business Information Systems*. Springer Berlin Heidelberg, 2012.

[24] Schunselaar, D. M. M., et al. "Petra: Process model based Extensible Toolset for Redesign and Analysis." (2014).

[25] Gagne, D., Shapiro, R.: BPSim 1.0. http://bpsim.org/specifications/1.0/WFMCBPSWG-2012-01.pdf (Feb 2013)

[26] Leemans, Sander JJ, Dirk Fahland, and W.M.P. van der Aalst. "Scalable Process Discovery with Guarantees." *Enterprise, Business-Process and Information Systems Modeling*. Springer International Publishing, 2015. 85-101

[27] http://www.win.tue.nl/coselog/wiki/project accessed on: 09 July 2015

[28] Buijs, Joos CAM, Boudewijn F. van Dongen, and W.M.P. van der Aalst. "A genetic algorithm for discovering process trees." *Evolutionary Computation (CEC), 2012 IEEE Congress on*. IEEE, 2012.

[29] Dalinghaus, W. R. *An Introduction to Reference Process Modeling in Healthcare*. Diss. Master Thesis, Eindhoven University of Technology, 2013.

Appendices

Appendix A Trace Pattern Analysis
[Redacted due to confidenatiality]

Appendix B Experiment result of discovering process
[Redacted due to confidenatiality]

Appendix C Processing and Waiting Time
[Redacted due to confidenatiality]

Appendix D Work Schedule Pattern and Task – Resource relation
[Redacted due to confidenatiality]

Appendix E Data perspective
[Redacted due to confidenatiality]

Appendix F Impact Analysis
[Redacted due to confidenatiality]