Delft University of Technology

FACULTY MECHANICAL, MARITIME AND MATERIALS ENGINEERING
Department Maritime and Transport Technology

Mekelweg 2
2628 CD Delft
the Netherlands
Phone +31 (0)15-2782889
Fax +31 (0)15-2781397
www.mtt.tudelft.nl

Specialization: Transport Engineering and Logistics

Report number: 2019.TEL.8332

Title: **Integral Internal Transport Planning and Coordination of Multiple Assets and Resources for turnaround of Aircraft for KLM at Amsterdam Airport Schiphol**

Author: P.L. Vos

Title (in Dutch) Integrale Interne Transportplanning en Coördinatie van Meerdere Assets en Middelen voor de omdraai van Vliegtuigen voor KLM op Amsterdam Airport Schiphol

Assignment: Master Thesis

Confidential: Yes

Initiator (university) Prof.dr. R.R. Negenborn

Initiator (company) B. Tulleken, MSc (Unit Manager Flow Control, KLM)

Supervisor: Dr. W.W.A. Beelaerts van Blokland

Date: March 27, 2019

Integral Internal Transport Planning and Coordination of Multiple Assets and Resources for turnaround of Aircraft for KLM at Amsterdam Airport Schiphol

by
Peter L. Vos
4034481

Transport Engineering and Logistics
Delft University of Technology

Delft

2019

APPROVED BY:

Dr. W.W.A. Beelaerts van Blokland

B. Tulleken, MSc

Prof.dr. R.R. Negenborn
Chair of Advisory Committee

Abstract

Currently, KLM's apron processes work in separate silos and there is only minimal integration. The result is that processes tend to work to a local optimum instead of a global optimum. A Transport Logistic Control Tower is a centralized information hub from which integral decisions can be made which could benefit the overall turnaround process at Schiphol airport. There is little knowledge about the effects on the system performance when input parameters are varied. A uniform way of scheduling tasks and making integral decisions is missing.

The goal of this research is to determine the integral control scenario that should be used by a Transport Logistic Control Tower in order to optimize the overall performance of KLM's apron processes. The main research question is formulated as:

> *Which integral control scenario should be used by a Transport Logistic Control Tower in order to optimize the overall performance of KLM's apron processes?*

In the current state analysis the Delft Systems Approach is used in combination with swim lane analyses to analyze the current system. Relations between the different processes, how they interact and the decisions that are made to plan their tasks are analyzed to get a better understanding of the apron processes. To asses the performance of the apron processes, the On Time Start (OTS), On Time Performance (OTP) and the On Time Delivery (OTD) are introduced. The number of flight delays and the corresponding costs are used to combine the process KPIs in order to make a final decision on the best control scenario for KLM.

A thorough data analysis is performed by combining the multiple data sources of KLM to get insight in the performance of the system. This showed us that the data quality is not sufficient, and data does not correspond well with the real system. It is therefore recommended to use historical data in combination with the current calculation methods to improve the way the task duration is calculated. This also means that the way data is stored in the HHT should be automated to reduce the amount of measurement errors.

A simulation model is designed to assess different control scenarios and the effect they have on the KPIs. The one day run shows significant improvements when dynamic flight priorities are used in combination with a departure oriented control scenario that schedules based on departure time. Compared to the same control scenarios using static priorities, an average improvement of 31,1% or 1,7 million euros annually could be seen in the delay costs. This control scenario is therefore recommended and shows the necessity to use a uniform way of planning and the potential of the proposed dynamic flight priorities.

The recommendations are done based on the analyses and the found results, and used to advice KLM on their future control scenario for their apron processes and improvement levers from which they could benefit in the future.

Preface

This master thesis is written for the master Transport Engineering and Logistics at the Delft University of Technology. This thesis is performed during an internship at the department "Flow Control" at KLM Royal Dutch Airlines. The last couple of exciting months I learned many new people and experiences about the airline industry and what it's like to work at a big company as KLM.

I could never have conducted this research without the support of a number of important people. First, I would like to thank Bob Tulleken for the opportunity he gave me to do an internship at KLM and his willingness to supervise me from KLM's end. He guided me in the right direction and was always available if I needed his opinion about questions; also he told me which people I could best talk to for information or data. I always enjoyed our meetings in which he provided me with valuable and fresh input for my research problem. Also, I would like to thanks all colleagues at KLM, who were always available for questions on data and process execution.

This research was the first time I used simulation software, which was completely new for me. I would like to thank Mark Eager from Scheduling Solutions, who had multiple meetings with me during the last couple of months when he helped me built my simulation model in Simio. He lives in Australia and even with the big time difference he was always willing to help me with the problems I faced. Without his help I would never have been able to get the full potential out of my model.

Furthermore, I would like to express my gratitude to Dr. Wouter Beelaerts van Blokland and Prof. dr. Rudy Negenborn, my supervisors at Delft University of Technology. Their valuable input, sharp questions and reviews during several meetings have guided me several times in the right direction.

I would also like to take this opportunity to thank my girlfriend Sterre Burgers who was and is always there for me. She was always willing to discuss any issues I faced during the writing of this thesis and was of great importance during this time. Also I would like to thank my family and friends who made my time as a student in Delft amazing. My parents always encouraged me to pursue my interest and were always interested in what I was doing.

Peter L. Vos
Delft, 2019

Table of Contents

LIST OF TABLES

LIST OF FIGURES

Acronyms

A	Arrival
A-CDM	Airport Collaborative Decision Making
AAS	Amsterdam Airport Schiphol
ACGT	Actual Commence of Ground Handling Time
ADC	All Doors Closed
AIBT	Actual In-Block Time
ALDT	Actual Landing Time
AOBT	Actual Off-Block Time
APU	Auxiliary Power Unit
ARDT	Actual Ready Time (for Movement)
AS	Aircraft Services
ASAT	Actual Start Up Approval Time
ASRT	Actual Start-Up Request Time
ATA	Actual Time of Arrival
ATC	Air Traffic Control
ATD	Actual Time of Departure
ATF	Actual Time of Flight
ATM	Air Traffic Management
ATOT	Actual Take Off Time
BTS	Baggage Turnaround Services
CBA	Cost-Benefit Analysis
CDM	Collaborative Decision Making
CHIP	"Communicatie Hub IndelingsProgramma"
CISS	Central Information System Schiphol
CR	Critical Ratio
CTOT	Calculated Take Off Time
D	Departure
DAM	Duty Area Manager
DD	Due Date
DES	Discrete Event Simulation
DFM	Duty Flow Manager

DFM’s	Duty Flow Managers
DHM	Duty Hub Manager
DMA	Duty Manager Aircraft services
DSA	Delft Systems Approach
EBD	Estimated Block Time
EIBT	Estimated In-Block Time
EOBT	Estimated Off-Block Time
EOH	End Of Handling
FIRDA	Flight Information Royal Dutch Airlines
FPU	Fixed Power Unit
GOMS	Ground Operations Manual Schiphol
GPU	Ground Power Unit
GS	Ground Services
HCC	Hub Control Center
HHT	Hand Held Terminal
HPO	High Performance Organization
HubDB	Hub Data Base
ICA	Intercontinental
KCS	KLM Catering Services
KLC	KLM Cityhopper
KLM	Royal Dutch Airlines
KPI	Key Performance Indicator
KPIs	Key Performance Indicators
LST	Least Slack Time
LVNL	"Luchtverkeersleiding Nederland"
MGT	Minimum Ground Time
MTT	Minimum Turnaround Time
OC	Operations Control

OCC	Operations Control Center
OTD	On Time Delivery
OTP	On Time Performance
OTS	On Time Start
PROPER	PROcess-PERformance
RTD	Real Time of Departure
SD	System Dynamics
Simio	SImulation Modeling framework based on Intelligent Objects
SMORE	Simio Measure of Risk & Error
SOBT	Scheduled On-Block Time
SPT	Standard Process Time
STD	Standard Time of Departure
SVSM	Swim lane Value Stream Mapping
TA	Turnaround
TAT	Turnaround Time
TC	Team Captain
TGM	Team Member Large Material
TLCT	Transport Logistic Control Tower
TLO	"TeamLeider Omdraai"
TM	Team Member
TM's	Team Members
TOBT	Target Off-Block Time
Triple-A	Amsterdam Advanced Air Traffic Control
TSAT	Target Start-Up Approval Time
ULD's	Unit Load Devices
VOP	"VliegtuigOpstelPlaats"
VSM	Value Stream Mapping
WPI's	"Werkplekinstructies"

CHAPTER

1

INTRODUCTION

1.1 General Background

Amsterdam Airport Schiphol (AAS) is an important hub in Europe, and the Royal Dutch Airlines (KLM) is its main carrier. Many flights depart and arrive each day, and as a result there are a lot of ground operations at Schiphol. The dynamic nature of the airline industry makes the control of the ground operations vital for the operation to work according to the expectations.

In the current situation KLM is growing faster than Schiphol. The maximum number of flight movements imposed by the government was almost reached in 2017, and will likely reach its maximum in 2018 [Sedee, 2018]. The result is that airlines like KLM have to deal with shortages of gates and slots and need to use them more efficiently if they want to keep growing. This puts pressure on the control of the ground processes, which have to perform more efficient in a system that gets more complex. Not much is known about which variables are important for the performance and how they influence the system. The different ground processes currently operate individually in different silos, with only minimal integration between the different silos. KLM wants a more integral approach when controlling the different processes by sharing more information between the silos. This allows for better scheduling and a faster reaction time to last-minute changes in the schedule.

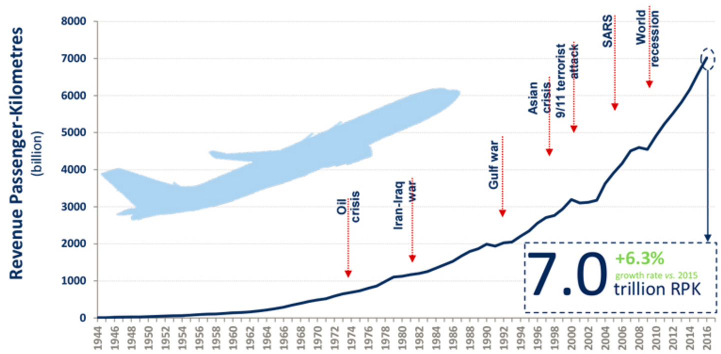

Figure 1.1 Revenue Passenger-Kilometers 1944 - 2016 [ICAO, 2017].

1.1.1 The Airline Industry

Figure 1.1 shows the worldwide growth of Revenue Passenger-Kilometers. Except for some small declines due to big events in history, a superlinear growth of the number of passengers can be seen during this period. This increase in air traffic makes major airports like Schiphol more and more congested: sensible delays may be caused by ground operations, and problems related to the efficiency and the safety of the apron area assume more and more importance [Andreatta et al., 2014].

Another key change in the industry is the expansion of low-cost carriers and the airlines from the Middle-East, ensuring that the game with competitors on all routes is played on product and price. In the current situation KLM is too complex, slow and expensive to stay competitive against these airlines. As the main carrier at Schiphol airport, KLM is continuously trying to improve its performance by a more efficient use of its resources.

Seven Wave Network

Since the foundation of KLM in October 1919, Amsterdam Airport Schiphol has been its main base. In 2017, KLM welcomed a record amount of 32.7 million passengers aboard its flights. The greatest rise was in Europe. The high passenger growth led to a record load factor of 88.4% over the year as a whole, which is 1.2% higher than in 2016 [KLM, 2018h]. Schiphol is a hub airport which serves as a gateway for airlines to transfer passengers to their final destination. The pattern of arriving and departing KLM aircraft at Schiphol is concentrated into seven peaks of an hour and a half to two hours, referred to as 'waves' in airline jargon [KLM, 2018f]. The waves are in sync with the arrival and departure patterns of passengers in Western Europe and are illustrated in figure 1.2. The ICA flights

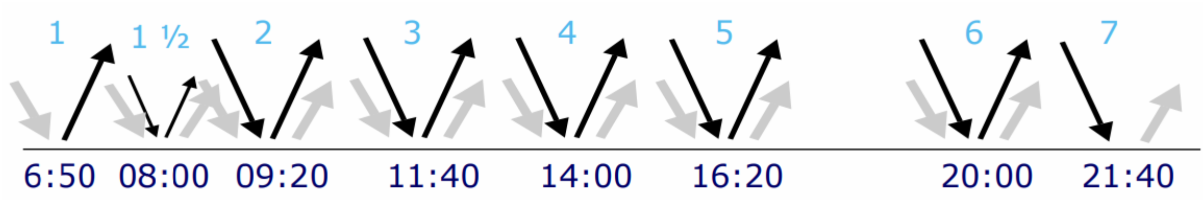

Figure 1.2 Seven Wave Network of KLM [KLM, 2018f].

are represented by the light gray arrows, and the European flight by black arrows. This carefully coordinated balance between arriving and departing flights during the day forms the basis (hub) of KLM's worldwide network [KLM, 2018f]. The seven wave system results in high peaks and low valleys of aircraft that need to be handled. In order to maintain an advantage against competitors, KLM decreased connection times at Schiphol to a minimum. This puts pressure on all Turnaround (TA) processes, making a seamless and fast integration between the different processes very important. The guaranteed connection time by KLM is crucial for competition between other airports and any delay can lead to extra costs and loss of goodwill.

1.1.2 KLM Operation Control

KLM distinguishes three layers of control, Operation Control, Hub Control and Department Control, each making decisions on a different level.

Operational Control Center

The Operations Control Center (OCC) can be seen as the nerve center of all KLM operations and houses Operations Control (OC). The controllers in the OCC have a global view of the different operations and hubs around the world, and it is their task to make the best integral decisions to increase profit. The OCC is divided into different KLM operations which each have their own controller. Together these controllers collaborate to make the best decisions for the KLM flight schedule. Decisions that are made by the OCC could be:

- The KLM network
- Priority on aircraft handling
- Monitoring situations around the world that can influence the network

Hub Control Center

The OCC takes decisions on a global level, whereas the Hub Control Center (HCC) tries to make the most optimal decisions for the operations at the hub Schiphol. The HCC is much smaller than the OCC, and consists of approximately 20 employees. The flight schedule which is produced by the OCC is used to make a planning for the hub. The Duty Hub Manager (DHM) is responsible for the operational performance of the HCC and is in direct contact with the Duty Manager Aircraft services (DMA) of the department control. Some examples of decisions that the HCC has to make are:

- Priority on aircraft handling (changes <10 min)
- Line maintenance
- Gate allocation

Department Control

Department control supervises the ground processes and is the closest to the operation. Currently each ground process has its own controller which tries to find the most optimal order to perform all tasks. The control departments use a program called "Communicatie Hub IndelingsProgramma" (CHIP), which will be described in section 3.4.1. Some processes can use the built-in optimizer of CHIP uses to schedule the tasks automatically, but the controllers can override these decisions when they know a better solution.

The control departments make decisions about which teams handle the different aircraft, and if there are shortages shift people between teams for example.

1.1.3 High Performance Organization

To be more competitive in the current market KLM started the HPO project in may 2016 to transform to a High Performance Organization (HPO). An HPO is a more customer-oriented, cheaper, faster and simpler organization compared to traditional organizations [KLM, 2016]. The HPO-project has four goals: contributing to the customer focus, improving the collaboration in the organization, increasing involvement and reducing costs [KLM, 2016]. To reach these goals]changes in the organization are necessary. One of the key changes in this process is to become a more horizontal organization with less management layers and more integration between the different silos.

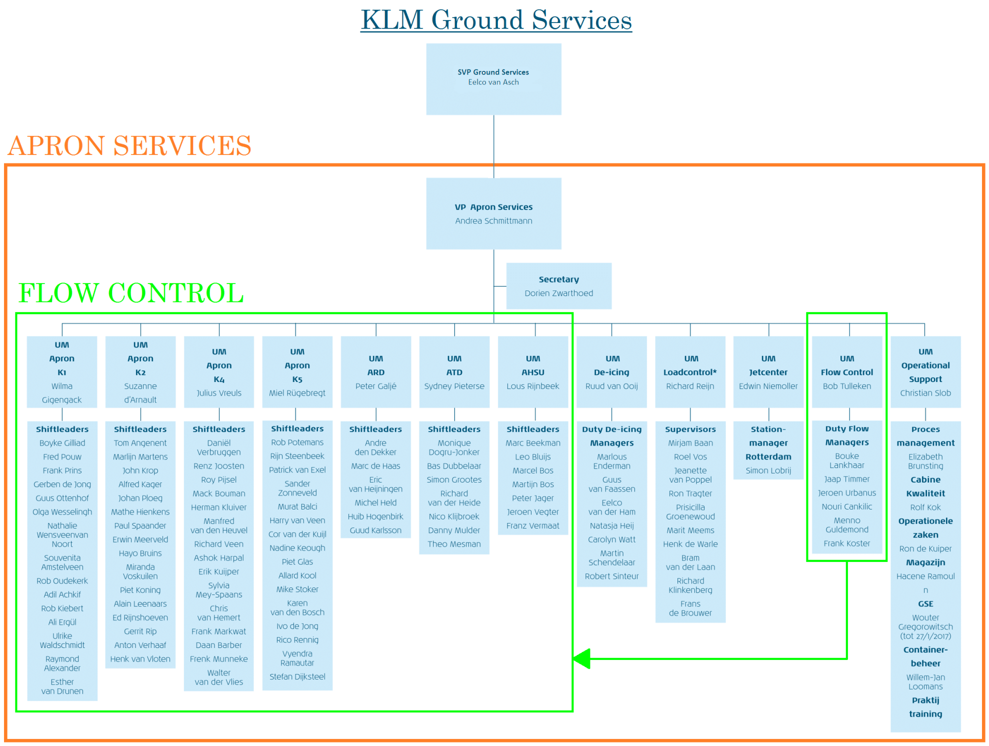

Figure 1.3 KLM Ground Services [Services, 2018] (modified by P.L. Vos).

1.1.4 KLM Ground Services & Apron Services

KLM Apron Services was introduced in January 2017 and is part of KLM Ground Services. This can be seen in figure 1.3. Apron is a merger between the former departments Aircraft Services (AS) and Baggage Turnaround Services (BTS), which is responsible for all processes "under the wing". The goal of this merge is to increase the integration and efficiency of the current resources.

Flow Control

Figure 1.3 also shows the flow department as part of Apron Services. The unit flow control was introduced with the objective to control the flows of equipment and people between the different gates. The flow department is highlighted with a green box on the right side of the figure and controls the departments which are highlighted by the green box on the left side of the figure. These departments contain the following processes:

- Aircraft Refueling
- Catering Services
- Aircraft Towing & Pushback
- Cleaning
- Water & Toilet Services
- Ramp Services

1.1.5 Airport Collaborative Decision Making

The Airport Collaborative Decision Making (A-CDM) project was started by EUROCONTROL back in 2006. The project aims at improving the overall efficiency of airport operations by optimizing the use of resources and improving the predictability of events. It focuses especially on aircraft turnaround and pre-departure sequencing processes [EUROCONTROL, 2018]. Increased predictability can be of significant benefit for all major airport and network operations; it raises both productivity and cost-efficiency [EUROCONTROL, 2018]. Some key players which could benefit from A-CDM are: airports, air traffic control, ground handlers, and the European flight network. Along with this increased predictability, Airport CDM brings myriad other benefits for airports, such as environmental impact reduction and enhanced planning of the turnaround, to name but two [EUROCONTROL, 2018].

EUROCONTROL defines three key elements in the foundation layers of Airport CDM, each bringing additional benefits [euro cdm, 2009]:

1. **Information sharing:** between all involved partners, provide the right information at the right time to the right people. Each partner has at some point a piece of information that is more up-to-date and more reliable than the estimates used by other partners; yet all too often this better information is not shared. CDM Information Sharing helps to create common situational awareness by making this information visible to those that are affected by it. Each partner has the same picture, more accurate than before, and acts on it.

2. **Driven by processes and procedures:** Airport CDM must be driven by procedures and processes, linked into a common platform. Only well-defined procedures (including vocabulary) can ensure that those who have the best information regarding the status of the flight are responsible for informing others, and doing that in time to allow others to act on the update.

3. **Milestones:** are the key to efficient operations. The whole process can be further improved by defining a set of so-called "milestones". These represent the significant events that occur during inbound flights and following turnaround. By monitoring these events and following the procedures and rules that are defined for each one of them, the partners can anticipate problems quickly when there is any deviation from the plan [euro cdm, 2009].

EUROCONTROL gives the following definition for the milestone approach in layer three: "The Milestone Approach element describes the progress of a flight from the initial planning to the take off by defining Milestones to enable close monitoring of significant events. The aim is to achieve a common situational awareness and to predict the forthcoming events for each flight with off-blocks and take off as the most critical events." Figure 1.4 shows the 16 basic milestones, as well as a list with some additional information. For a more detailed description of these milestones the reader is referred to the EUROCONTROL Airport CDM Implementation Manual [EUROCONTROL, 2012].

Number	Milestones	Time Reference	Mandatory / Optional for Airport CDM Implementation
1	ATC Flight Plan activation	3 hours before EOBT	Highly Recommended
2	EOBT - 2 hr	2 hours before EOBT	Highly Recommended
3	Take off from outstation	ATOT from outstation	Highly Recommended
4	Local radar update	Varies according to airport	Highly Recommended
5	Final approach	Varies according to airport	Highly Recommended
6	Landing	ALDT	Highly Recommended
7	In-block	AIBT	Highly Recommended
8	Ground handling starts	ACGT	Recommended
9	TOBT update prior to TSAT	Varies according to airport	Recommended
10	TSAT issue	Varies according to airport	Highly Recommended
11	Boarding starts	Varies according to airport	Recommended
12	Aircraft ready	ARDT	Recommended
13	Start up request	ASRT	Recommended
14	Start up approved	ASAT	Recommended
15	Off-block	AOBT	Highly Recommended
16	Take off	ATOT	Highly Recommended

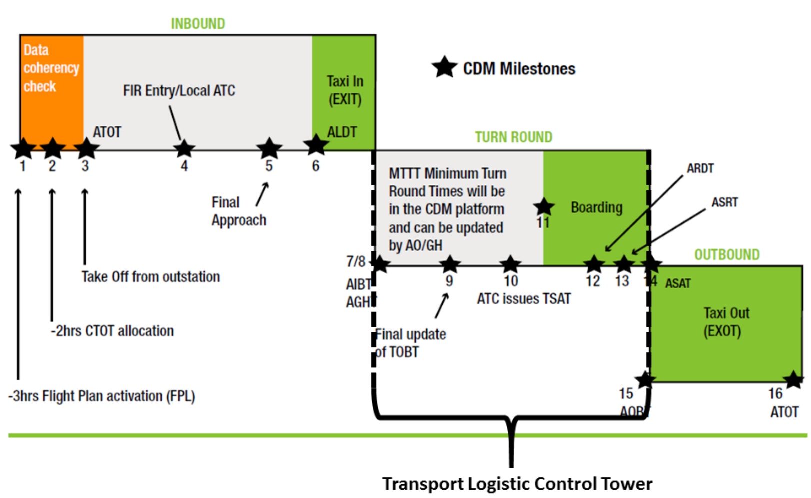

Figure 1.4 CDM Milestones [EUROCONTROL, 2012].

CDM@AMS Program

The CDM@AMS program at Schiphol Airport is a joint initiative between the airlines, handlers, "Luchtverkeersleiding Nederland" (LVNL) and Schiphol. The key aims of the program are to facilitate the sharing of operational processes and data to allow better informed decisions to be made [Schiphol, 2018a]. To assure the best possible coordination of resources the optimization of the turnaround process is an important factor. By providing the different stakeholders and partners with accurate and timely information, decisions which benefit the turnaround of a flight can be made.

The implementation of the A-CDM program at other major European Airports has shown improvements in stand and gate management, resource management, slot adherence leading to reduced costs for all parties and improved accuracy of passenger information [Schiphol, 2018a].

The Target Off-Block Time (TOBT) is based on the Estimated In-Block Time (EIBT) + Minimum Turnaround Time (MTT). The MTT is predefined by the KLM, and determines how much time each process needs to finish its task. The CDM project focuses on the whole timeline as seen in figure 1.4, while all ground processes should be performed between milestones 7 and 14. This area is marked in figure 1.4 by the accolades. In extent of the HPO and CDM@AMS programs, KLM wants to know what the best way is to set up the control departments and information flows for their ground processes to make the best use of the more accurate information provided by the CDM program. A possible solution is the use of a Transport Logistic Control Tower (TLCT), which will operate in the area marked by the accolades in figure 1.4. This will be further explained in section 1.1.6.

1.1.6 Transport Logistic Control Tower

The essence of the TLCT concept is to make the information needed to control the apron more transparent. The heart of the control tower is an information hub that is supported by a set of detailed rules for decision making. The big advantage of this central information hub is that it collects and integrates information that comes from a multitude of sources and then distributes it in one consistent form. With this integrated overview, the control tower operator can identify disturbances like equipment failure or delays and early arrivals at an early stage and communicate this with all parties involved.

When looking at the Anthony triangle [Anthony, 1965], the TLCT operates at the operational level and covers the part of the CDM project between milestones 7 and 14, see figure 1.4. The CDM project provides more accurate information about the aircraft which results in a better prediction of the Actual In-Block Time (AIBT) and TOBT. The control of the ground processes happens between these two points, and having more accurate time information allows for better scheduling of the ground processes. Where the CDM project focuses on the scheduling of the aircraft in the available slots, the TLCT will focus on a more integral control of the ground processes. Having all information at a central location (the TLCT) allows for an integral control and faster and better decision making. Already briefly mentioned in section 1.1, the different silos currently operate on their own with only minimal integration between them. The Control Tower provides a vertical dimension (in the silo) as well as a horizontal dimension (through the silos). Figure 1.5 shows a conceptual Transport Logistic Control Tower Model for the apron processes of an airport.

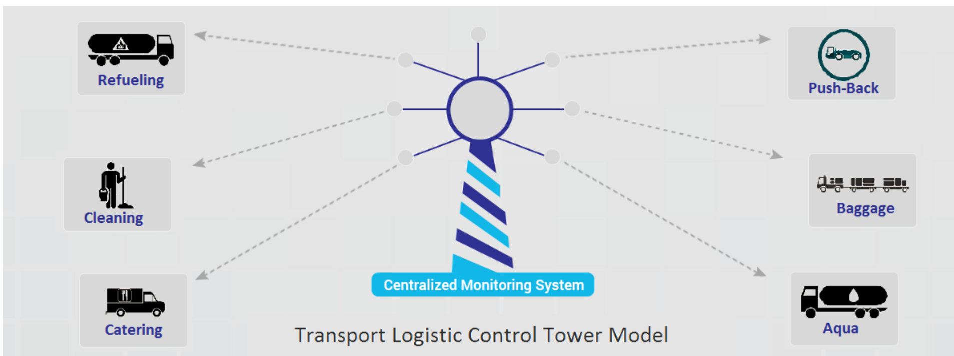

Figure 1.5 Transport Logistic Control Tower Model.

1.2 Research Outline

This section will describe the scope, objectives and the research questions in more detail.

1.2.1 Research Scope

As mentioned before the focus of the research will lie at the control of the apron processes. These processes are mentioned in section 1.1.4, and will be further explained in chapter 3. How the processes operate is out of this scope.

From a strategic perspective the control of the different apron processes can be seen as a black box. A black box could be a device, system, or object which is described in terms of its inputs and outputs, without any knowledge of its actual internal workings. For this research, the black box includes all apron processes which are used to handle arriving aircraft (input), and perform all necessary steps to make the aircraft ready again for departure (output). Figure 1.6 shows the black box. Everything outside these borders is not taken into account during this research.

Figure 1.6 Black box of the scope of this research.

This black box can be elaborated to a simplified and general version of the system that will be researched in this paper, this can be seen in figure 1.7. It shows a map of AAS, the location of the pitches for the aircraft and the routes they use to taxi. All ground processes happen at these pitches, and to get there they drive over the peripheral roads which in general lie at the edges of the black shape of the airport in the figure.

All aircraft that are being handled at the gates in the figure are part of the scope of this project. This means freighters are out of this scope.

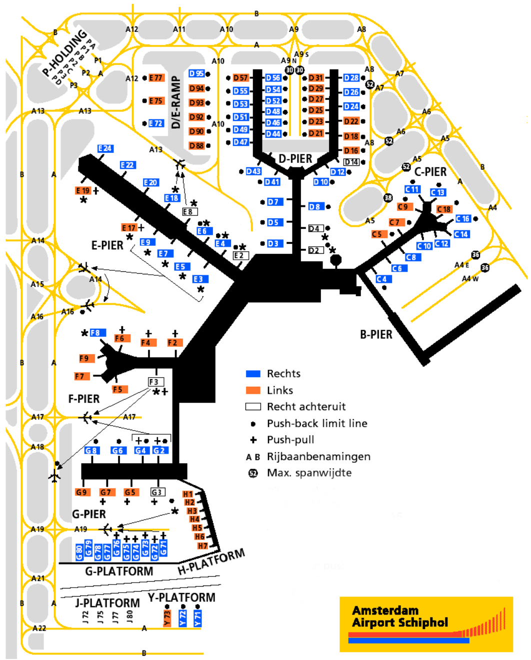

Figure 1.7 Map of Amsterdam Airport Schiphol [Schiphol, 2018b].

1.2.2 Research Objective

The objective of the research is to realize a more integrated control of the apron processes using a transport logistic control tower and compare this to the current situation.

A thorough current state analysis will be the basis for the development of the simulation model that will be used to analyze the performance of the control of the ground processes. This includes all aspects of the process: process steps, information flows, decision rules, operational control and execution. Also data availability and Key Performance Indicators (KPIs) will be reviewed and additionally several improvement possibilities will be defined.

The overall objective of the research can be summarized as follows:

> *Realize a more integrated control between the aprons processes using a Transport Logistic Control Tower and assess different control scenarios.*

1.2.3 Research Questions

Based on the research scope and objectives the following research question is formulated:

> *Which integral control scenario should be used by a Transport Logistic Control Tower in order to optimize the overall performance of KLM's apron processes?*

The main research question is driven by the following sub-questions:

1. What is the current state practice of KLM's apron processes?
    - What is the current operational practice: process steps, information flows, decision rules, operational control and execution?
2. What alternative scenarios can be used to control the apron processes?
3. How is the process currently monitored: data availability and KPIs?
4. How can alternative control scenarios be assessed using simulation?
5. How do the redesigned control scenarios improve the performance of KLM's apron processes?

CHAPTER

2

LITERATURE

This chapter is used to perform a literature study which is done to find prevailing literature about the topic. Mapping the information flows is a key step in this research and methodologies which can be used for this task will be described in this chapter.

2.1 Process Improvement Methods

There is a multitude of different ways to visualize the value stream of a system. The well-known Value Stream Mapping (VSM) is only one of many ways to structure it, but most of them have a structure according to either:

- Time
- Physical location
- Sequence

or any combination thereof [allaboutlean.com, 2015]. An overview of some of the different methods which can be used to visualize the value stream, and in which category they belong can be found in figure 2.1. Depending on the system and what you want to achieve, some may be better than others.

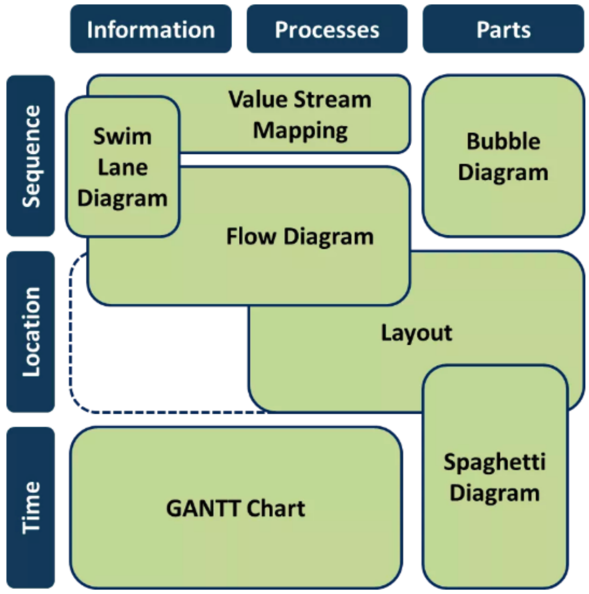

Figure 2.1 Relation of different methods to visualize systems [Roser, 2015a].

The focus of the research will be on the control of the apron processes which is a complex system with many different subsystems. An important step is analyzing the information flows between the different departments and processes in the system. The apron processes have to be performed in a certain sequence in a tight time window. Different processes have to work simultaneously and at the same location, the coordination of this requires the sharing of information between the different silos. Looking at figure 2.1 it can be seen that a swim lane analysis in combination with VSM or flow charts is a good method to analyze such a complex system.

2.2 The Delft Systems Approach

The Delft Systems Approach (DSA) makes use of these methods and is an instrument to decompose complex systems in a simplified overview [Veeke et al., 2008]. The PROcess-PERformance (PROPER) model is introduced as a tool for the description of a system and analyze all aspects of the system and its interrelations. The Proper model is based on the principle of "Zooming". A system is considered as a black box. From there, the system black box is opened and aggregation layers can be reached by selecting subsystems and functions of interest to zoom into. A system always consists of subsystems where, zoomed out, two subsystems can be distinguished: 1. a control system and 2. an operational system, see figure 2.2.

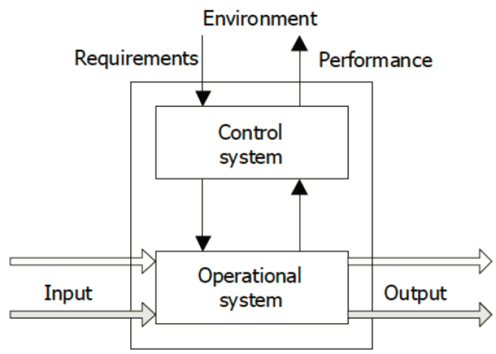

Figure 2.2 PROPER model of a system Veeke et al. [Veeke et al., 2008].

It is described by Veeke et al. as: "An industrial system is a subsystem of the organization as a whole; it contains a subset of the elements, but includes all of the relations. We now approach industrial systems from the viewpoint of the primary function, and at least three aspects are included in the conceptual model:

1. The "product" as a result of a transformation.
2. The "flow of orders"; without customer orders no products will flow. In this flow, orders are transformed into handled orders.
3. The "resources" (people and means) required to make the product. To make use of them, they must enter the system, and they will leave the system as used resources. The results of the transformations are delivered products, handled orders and used resources."

Figure 2.3 shows the conceptual model of the PROPER model as described by [Veeke et al., 2008]. The model can be used to define subsystems in a total system that needs further investigation. This is achieved by using aggregations layers, where each aggregation layer is based on the results of decision making in the preceding aggregation layers, thereby defining standards & efforts and dividing functions into a structure of sub functions [Veeke et al., 2008].

The control function coordinates all transformation functions by generating executable tasks from orders and by assigning resources. The task and assignment flows help to get insight in the factors that influence the performance of the operation. Bad performance can be the result of long task lead times, a shortage in assigned equipment, or misunderstandings between different components [Veeke et al., 2008]. By doing an in-depth analysis on these processes, a good understanding of the system is given. To achieve optimal use of resources, the information flows are essential to realize an integral control of the processes.

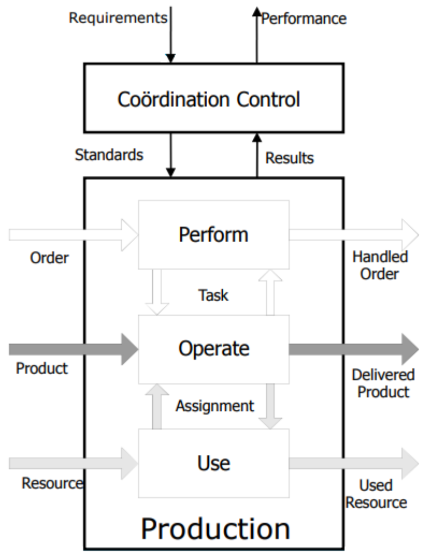

Figure 2.3 PROPER model as described by Veeke et al. [Veeke et al., 2008].

2.3 Swim Lane Analysis

A swim lane diagram is a helpful tool to map the steps or activities of a process flow or workflow. These activities are then grouped into swim lanes which are horizontal or vertical columns that contain all of the activities which fit into the category represented by that swim lane [modernanalyst, 2011]. Swim lane diagrams are regarded to be an important tool when validating business rules and procedures with stakeholders because they are believed to convey information about business process models effectively and efficiently [Jeyaraj and Sauter, 2014]. Swim lanes can represent many categories of information which makes them suitable to analyze systems with many actors and processes which have to work together. An example of a swim lane diagram can be seen in figure 2.4.

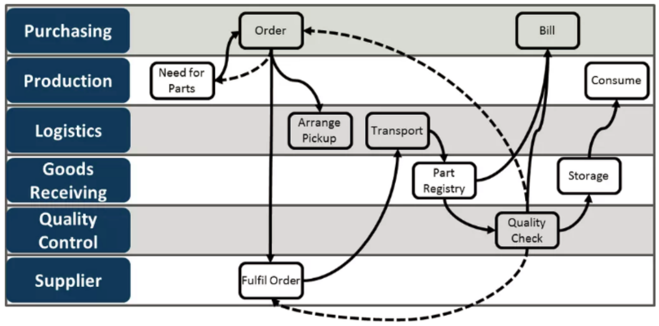

Figure 2.4 Example of a swim lane diagram [Roser, 2015b]

2.4 Push-Pull Strategy

The business terms push and pull originated in logistics and supply chain management [Yuvaraj and Zhang, 2013] and are an element of lean. Other elements are: reducing waste, standardization and managing processes [Jones et al., 1997].

Figure 2.5 shows the original meaning of push and pull as used in supply chain management and operations management:

- In **Push** systems; the information flow is in the same direction as the goods flow [Bonney et al., 1999]. Production begins based on demand (actual or forecasted demand).

- In **Pull** systems; every succeeding node makes an order request for the preceding node. The preceding node reacts by producing the order, which involves all internal operations, and

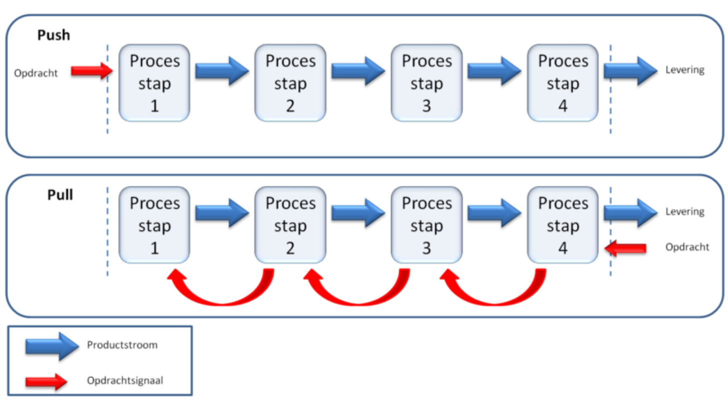

Figure 2.5 Push-Pull Strategy [Toledo, 2018].

replenishes when finished [Bonney et al., 1999]. Production orders begin upon inventory reaching a certain level.

- In **Hybrid push-pull** systems; a combination of both push and pull is used, where the interface between the push-based stages and the pull-based stages is sometimes known as the push-pull boundary or decoupling point [Harrison, 2005].

Harrison summarized when to use each one of the three strategies:

- A *push* based strategy is usually suggested for products with low demand uncertainty, and minimum customer input. The forecast will provide a good indication of the number of products to produce and keep in inventory.
- A **pull** based strategy is usually suggested for products with high demand uncertainty and high customer input. Therefore a system based on realized demand is more desirable.
- A **hybrid push-pull** strategy, usually suggested for products which uncertainty in demand is high, while economies of scale are important in reducing production and transport costs [Harrison, 2005].

2.5 KPI Selection

Key Performance Indicators (KPIs) are a type of performance measurement, that evaluate the success of an organization or of a particular activity in which it engages [Fitz-Gibbon, 1990]. There are many different KPIs possible and there is a need to understand what is important for an organization when choosing the KPIs. Various techniques to assess the present state of the business, and its key activities, are associated with the selection of performance indicators. These assessments often lead to the identification of potential improvements, so performance indicators are routinely associated with 'performance improvement' initiatives.

Two categories of KPI measurements can be distinguished [Chan and Chan, 2004]:

- **Qualitative** KPIs have a "descriptive" characteristic - a property, an opinion or taste which can be represented an interpretation of these elements as any numeric or textual value.
- **Quantitative** KPIs have a measurable characteristic - anything that involves numbers which are often measured against a standard. This is the most common type of KPI and is without distortion from prejudices or personal feelings. It covers many things for instance: number of delays, percentage up/down-time or average process times.

KPIs reflect strategic value drivers rather than just measuring non-critical business activities and processes [Bauer, 2004]. The right KPIs for improvement (what needs to be done?) can be found in these focus areas within the organization.

The Delft Systems Approach defines two concepts for assessment: 1. Effectiveness, 2. Efficiency:

$$\text{Effectiveness} = \frac{R_{Actual}}{R_{Standard}} \tag{2.1}$$

Where:

R_{Actual} : The actual result
R_{Standard} : The maximum attainable result with given means

$$\text{Efficiency} = \frac{S_{Standard}}{S_{Actual}} \tag{2.2}$$

Where:

S_{Actual} : The actual sacrifice of resources
S_{Standard} : The minimum attainable sacrifice with given means

These concepts will be used to assess the performance of alternative control standards.

2.6 Simulation Methods

There are several methods to evaluate the performance of a system, often divided in three categories; analytic methods, simulation or emulation and physical experiments [Thierry et al., 2008]. Most systems are too complex to use analytic methods, and physical experiments are too expensive to use on a large scale. "Simulation is the imitation of the operation of a real-world process or system" [Jerry, 2005]. It provides a cost-efficient method to analyze complex systems, and it is sometimes required that a model is developed which represents the key characteristics of the system, functions and behaviors of the selected system or process. The model represents the system itself, whereas the simulation represents the operation of the system over time [Ionica and Leba, 2015]. The ground operation system at Schiphol is subjected to multiple uncertainties in its processes that have a great impact on the performance of the system. To cope with these uncertainties and the dynamic behavior of the system, a simulation tool will be used as a tool for evaluating different ways to control the processes.

Figure 2.6 shows a high-level representation of the model and simulation development phases from real network environment to simulation model. The input of the simulation model consists of uncontrollable and controllable variables, which are converted into output variables that represent the response to the input variables of the modeled system. The feedback loop is an iterative process

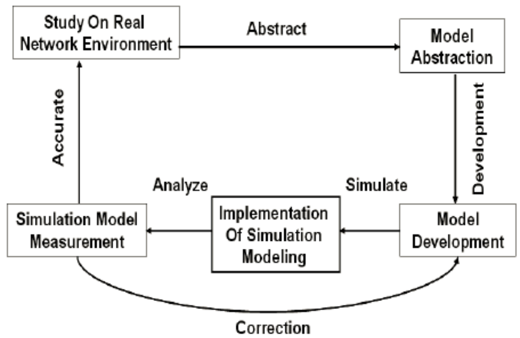

Figure 2.6 Model and Simulation Development Phases [Ismail and Zin, 2008].

where the input variables are varied in order to analyze different scenarios and get outputs which are closer to reality.

Continuous and Discrete Simulations

There are different simulation methods which can be categorized into two main categories: (1) Continuous Simulations and (2) Discrete Event Simulation (DES). The most common known continuous simulating method is System Dynamics (SD), which is an approach to understanding the nonlinear behavior of complex systems over time using stocks, flows, internal feedback loops, table functions and time delays [of Technology, 1997]. In this category the state of an object varies continuously resulting in many unnecessary calculations. In DES the state of an object varies at discrete events in time. Each event occurs at a particular instant in time and marks a change of state in the system [Robinson, 2004]. Because recalculations of the system are only performed when events occur, less computation power is needed, resulting in shorter simulation runs. The ground operations at Schiphol consists of many variables and different actors making it very suitable for DES.

Software

The selection of the right simulation tool is essential to achieve the objective of the research [Jerry, 2005]. For complex dynamic and stochastic systems there are many options for a simulation tool which can be categorized in:

1. **Simulation languages** - can be continuous and discrete and allow much detail and accuracy when modeling. Some discrete languages are Delphi Tomas, Matlab or Python, which all allow tailor made solutions but can take much time to develop from scratch.

2. **Simulator packages** - such as Simio, Arena or FlexSim can also be used when modeling stochastic and dynamic systems. These packages are object-oriented and have built in blocks which can be edited to the modelers preference. This allows for faster model development but less freedom compared to the simulation languages.

From the authors own experience building a model using a simulation language can be very time consuming. The result is that the modeler needs more time to develop the model, and less time is left to focus on the research questions itself. Therefore a simulator package is chosen.

Simio

SImulation Modeling framework based on Intelligent Objects (Simio) is an object-oriented simulation tool which is used in many different industries, such as: airports, manufacturing, supply chains or ports. The key advantage of Simio is that it provides much freedom by adding new blocks if necessary, but it can also use the built-in blocks and thus saving much time when building the model. It also supports 2D and 3D visualization, which allows easy model validation and verification and helps when finding bugs and errors in the model. It is an easy to use and effective DES package which has very useful built-in modules when plotting results and running multiple experiments.

2.7 Research Methodology

This research uses the generic approach proposed by Dr. Beelaerts van Blokland, called the Generic Research Process [van Blokland, 2018]. This approach consists of the following steps which can be seen in figure 2.7:

Figure 2.7 DMDMAT Methodology

- **Define:**
  The goal of the first phase of the research is to define the problem. A problem definition is given, as well as the scope, objective and the research questions. This is done in chapter 1.
- **Methodology:**
  The methodology phase is used to determine the approach that will be used to answer the

research questions described in the define phase. A literature study is done to find prevailing literature about the topic, and solving methods for similar problems are examined. This is covered in chapter 2.

- **Design:**
  A thorough current state analysis of the ground processes is performed using swim lane analysis. A theoretical preliminary model is made using these tools, which will be used to built a model in a discrete event simulation. The model will then be explained, verified and validated before experiments will be performed. This phase is described in chapters 3, 4 and 5.

- **Measure:**
  With the preliminary model introduced all necessary data, and where this is stored has to be analyzed. The different locations where data is stored will be mapped to provide a complete view on the data availability and which data is shared to which parties. This phase is intertwined with the design phase and will also be described in chapters 3 and 4.

- **Analyze:**
  The data from the measuring phase is used in the analyze phase to determine the current state performance and find possibilities for improvement. New improvement methods which could improve the current system will be proposed and designed. Future states of the system will be described here and can be found in chapter 4.

- **Test and Evaluate:** The last phase will evaluate the performance of the proposed improvement methods with the current state. The experimental plan is presented and simulations are performed. A conclusion of the experiments will be given as well as recommendations for future research will be given in chapters 6 and 7.

Research Overview

Figure 2.8 presents an overview of the research phase, chapters and corresponding research questions.

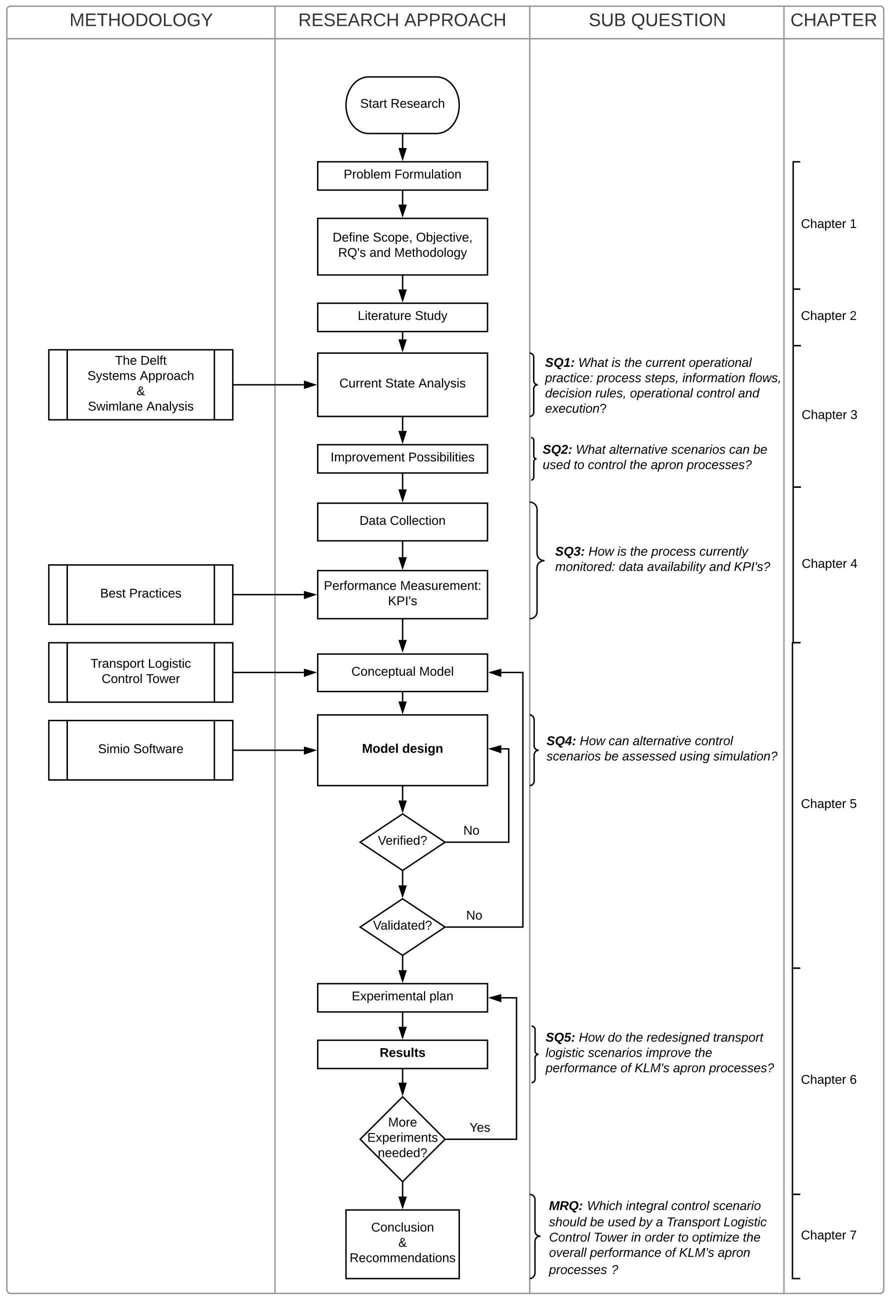

Figure 2.8 Research Methodology adopted from Jerry [Jerry, 2005].

CHAPTER

3

CURRENT STATE ANALYSIS

In this chapter a system analysis is performed. Herein, various aspects of the research with respect to the transportation of employees and equipment at the airport, as introduced in chapter 1, are treated. This chapter will answer the first sub questions:

1. *What is the current state practice of KLM's apron processes?*
   - *What is the current operational practice: process steps, information flows, decision rules, operational control and execution.*
2. *What alternative scenarios can be used to control the apron processes?*

3.1 The Delft Systems Approach

The general approach of the DSA [Veeke et al., 2008] as described in section 2.2 is used to further analyze the black box from figure 1.6. For this the PROPER model is used to get an understanding of the product, order and resource flows in the current system.

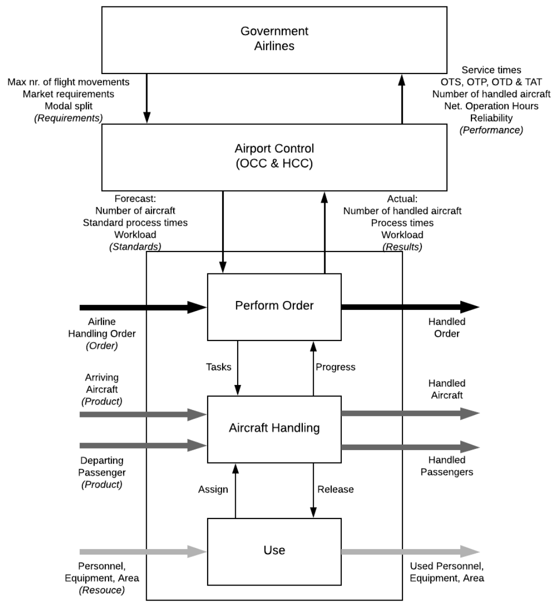

Figure 3.1 PROPER Model KLM Apron Services

Figure 3.1 shows the PROPER model for KLM's apron processes. Its core functions are abstracted to Perform the Airline Orders (order flow), Handle the Aircraft (product flow), and Use the Resources (resource flow):

- The *order flow* are the airline requests to handle the aircraft. The orders are given by the airlines which want their aircraft to be handled at Schiphol. Handled orders are the output.
- The *product flow* consists of the arriving aircraft at the gates and departing passengers. Aircraft arrive at Schiphol and need to be handled by performing the different apron processes as described in section 3.3. The output are handled aircraft ready for departure and passengers that arrived at their destination.
- The *resource flow* consists of the resources that are used to handle the aircraft. This includes all equipment and personnel used by the different ground processes as well as the area at the gates that is used for the handling.
- The *task flow* stands between the perform order function (order flow) and the aircraft handling function (product flow). When an airline has an order for the TA of an aircraft, the TA process is divided into multiple tasks: refueling, cleaning, pushback etc. based on the flight information.

The task lead time is a measure of the performance of the system and the responsiveness of the system to orders. This is sent back in the form of progress to the order flow.

- The *assignment flow* is between the aircraft handling function (product flow) and the use function (resource flow). To perform the tasks that have to be performed in order to handle the aircraft, the resources have to be assigned to the tasks determined by the order flow. Service time, operational time, defects and number of handled aircraft are valuable performance indicators for the system. After the resources have performed their tasks, they are released to be used again later.

The task and assignment flows help to get insight in the factors that influence the performance of the operation: bad performance can be the result of long task lead times, a shortage in assigned equipment, or misunderstandings between different components [Vos, 2017]. A good understanding of the system is obtained when zooming in on these processes. To achieve optimal use and control of the resources, the information streams will be visualized in the next sections using swim lane analysis.

3.2 Aircraft Handling Standards

There are different handling standards depending on the type of aircraft and the type of turnaround. To ensure an efficient handling procedure, standards are developed to categorize the different scenarios. For every type of aircraft a detailed documentation for each situation can be found at the KLM website [KLM, 2018d].

A short summary is given for the main aircraft types and handling standards that are used for the different turnaround scenarios.

Aircraft Types

Aircraft can be categorized in three main categories:

1. *Wide-bodies* - are aircraft with a fuselage wide enough (around 6m) to accommodate two passenger aisles. Flights executed with wide-bodies have intercontinental destinations and origins and are indicated as *ICA* flights. Unit Load Devices (ULD's) are mainly used to transport baggage and cargo, but there are also bulk holds if necessary.

2. *Narrow-bodies* - are single-aisle aircraft with a fuselage below 4 meters of width. Destinations are usually within Europe and are therefore indicated as *EUR* flights. Baggage and cargo is stored as bulk in the cargo holds.

3. *Commuters* - are smaller than narrow-body aircraft and often have short turnaround times with significantly less baggage. They operate as KLM Cityhopper (KLC), a daughter company 100% owned by KLM. They only have destinations and origins within Europe and are also indicated as *EUR* flights.

The complete fleet of KLM Royal Dutch Airlines as of 2018 can be seen in table 3.1.

Table 3.1 Fleet KLM (2018) [planespotters.net, 2018] [airfleets.net, 2018].

Aircraft	Category	In Service	Passengers
Boeing 747-400	Wide-body	15	408
Boeing 777-300ER	Wide-body	14	408
Boeing 777-200ER	Wide-body	15	316
Boeing 787-9	Wide-body	12	294
Airbus A330-300	Wide-body	5	292
Airbus A330-200	Wide-body	8	268
Boeing 737-900	Narrow-body	5	178
Boeing 737-800	Narrow-body	27	176
Boeing 737-700	Narrow-body	18	132
Embraer 190	Commuter	32	100
Embraer 175	Commuter	17	88
Total		**168**	

Ways of Handling

Each aircraft type has its own handling standard, which is provided by the manufacturer of the aircraft. The time each process takes is well documented in the Ground Operations Manual Schiphol (GOMS) (see section 3.2.2) provided by KLM and Schiphol and can be divided into three categories for each type of aircraft:

1. **Arrival:** The aircraft only arrives and does not have to depart anytime soon. It can stay parked at the gate, towed to one of the buffer location or towed to the hangar for maintenance.

2. **Turnaround:** The aircraft arrives to the gate and has to be prepared for departure again. This is the most critical type since there is little room for slack, especially if the turnaround time is the same or close to the MTT determined in the GOMS.

3. **Departure:** The aircraft is towed from another location (buffer or hangar) to the gate from which it has to depart.

3.2.1 Flight Priorities and Delay Costs

To determine which flight to delay in case of shortages in work capacity, priorities are given to flights. Controllers of each process have some general guidelines when scheduling their tasks.

1. "Star flights" have the highest priority.
2. *ICA* flights have a higher priority than *EUR* flights.
3. In case not all tasks can be scheduled, the controller asks priorities from the Duty Area Manager (DAM) and informs the Duty Flow Manager (DFM).

These set of rules on the priority of aircraft handling are static and controllers will always schedule their tasks in this order if not told otherwise. This means sometimes multiple EUR flights get delayed to let one ICA flight depart on time.

In general this approach works good, but on short delays (<15 minutes) the practice shows us that a delay of an EUR flight costs more than that of an ICA flight. The reason for this can be found in the fact that it is more difficult to make up time on short flights. Dynamically changing flight priorities could therefore benefit the controllers decisions on which flight to delay, and reduce the costs caused by delays.

Costs

Eurocontrol did research on the costs of delays for European airlines and distinguishes three types of passenger costs that can be considered [of Westminster, 2015]:

- "hard" costs: borne by the airline (measurable, bottom-line costs such as re-booking and compensation);

- "soft" costs: borne by the airline (such as loss of market share due to passenger dissatisfaction);

- "internalized" costs: borne by the passenger, not passed on to the airline (e.g. potential loss of business due too late arrival at meeting; partial loss of social activity.

A fuller discussion of these cost types can be found in the report adopted by EUROCONTROL [Eurocontrol, 2015]. Table 3.2 shows the costs that Eurocontrol gives to delays, which will be used in chapter 5 to calculate the delay costs for each flight in the model.

Total cost of passenger delay by delay duration and aircraft type (base cost scenario)

Delay (mins)	5	15	30	60	90	120	180	240	300
B733	40	250	910	3 320	6 800	11 110	22 180	36 370	53 520
B734	40	290	1 040	3 780	7 750	12 670	25 280	41 470	61 020
B735	30	230	810	2 950	6 040	9 860	19 690	32 290	47 510
B738	40	330	1 170	4 250	8 700	14 220	28 390	46 570	68 520
B752	50	400	1 420	5 180	10 610	17 340	34 610	56 770	83 520
B763	70	490	1 760	6 420	13 150	21 490	42 900	70 360	103 530
B744	110	790	2 850	10 360	21 220	34 670	69 220	113 530	167 050
A319	40	270	960	3 510	7 180	11 730	23 420	38 410	56 520
A320	40	310	1 110	4 030	8 260	13 500	26 940	44 190	65 020
A321	50	380	1 350	4 900	10 040	16 400	32 750	53 710	79 020
AT43	10	90	310	1 120	2 290	3 740	7 460	12 240	18 010
AT72	20	120	440	1 610	3 300	5 400	10 780	17 680	26 010
DH8D	20	140	490	1 770	3 620	5 920	11 810	19 380	28 510
E190	30	180	660	2 390	4 890	7 990	15 960	26 170	38 510
A332	80	550	1 980	7 200	14 740	24 080	48 080	78 860	116 030

Figure 3.2 Total cost of passenger delay by delay duration and aircraft type (base cost scenario) [of Westminster, 2015]

3.2.2 GOMS Standard Processes

There are two types of processes, the first is focused on Arrival (A) and the second on Departure (D). GOMS has defined the order of the processes based on the dependency on events during the TA process. The longest running time (platform or cabin process) determines the minimum turnaround time. This is being then rounded off at 5 minutes. Any space in turn over time in the cabin will benefit the boarding process to one maximum start time of D-30. The standard processes according to GOMS for the gate/cabin processes and the platform processes are described below:

Gate / Cabin Process

During deboarding may start and can be performed in parallel:

- Cabin cleaning (condition platform: rear staircase is connected).

After deboarding may start and can be performed in parallel:

- Catering (condition platform: truck is positioned).
- KLM Catering Services (KCS) may also handle the aircraft with one scissor car.
- Cabin security search (platform condition: bridge is connected).
- Technical handling.

Before boarding must be completed / present:

- Cabin cleaning.
- Catering.
- Cabin security search.
- Cockpit crew in aircraft.

During boarding may still be present:

- Technical service.

Cabin check:

- Must be ready before boarding.

Boarding:

- Finished on D-04 at the latest.

Platform Process

After placing blocks (A+01) may start:

- Connect Ground Power Unit (GPU).
- Connect bridge (or passenger stairway).
- Place pylons.

After placing pylons (A+02) may start:

- Connect stairs.
- Positioning catering truck.
- Unloading / loading baggage / freight / mail.
- Water and toilet service (in random order, but draining the toilet and filling water are not performed simultaneously).
- Refueling (when only departing), otherwise wait until the deboarding is finished. Refueling may continue during boarding.
- Technical handling.

  On D-04, previous handling processes must be completed (with the exception of the loading of the baggage / cargo / mail (= D-02)).

To finalize aircraft start the following tasks:

- Remove the stairs : D-06 (B737-900: D-07)
- Disconnect GPU : D-06
- Remove pylons : D-04
- Remove the bridge : D-02
- Remove blocks : D-01

Start technical departure service: D-04.

These standard processes can be drawn in a flow chart to show dependencies between the ground processes. The ground handling process can be seen in figure 3.3, and can be used for all types of aircraft. The type of aircraft, destination and load are different for each flight and determine the length of each task, and in result the critical path. Critical processes are dependent on other processes, and a delay automatically leads to a delay of the Turnaround Time (TAT) if no measures are taken. Noncritical processes have more freedom when scheduled due to less dependency on other processes and more built in slack in case of a delay.

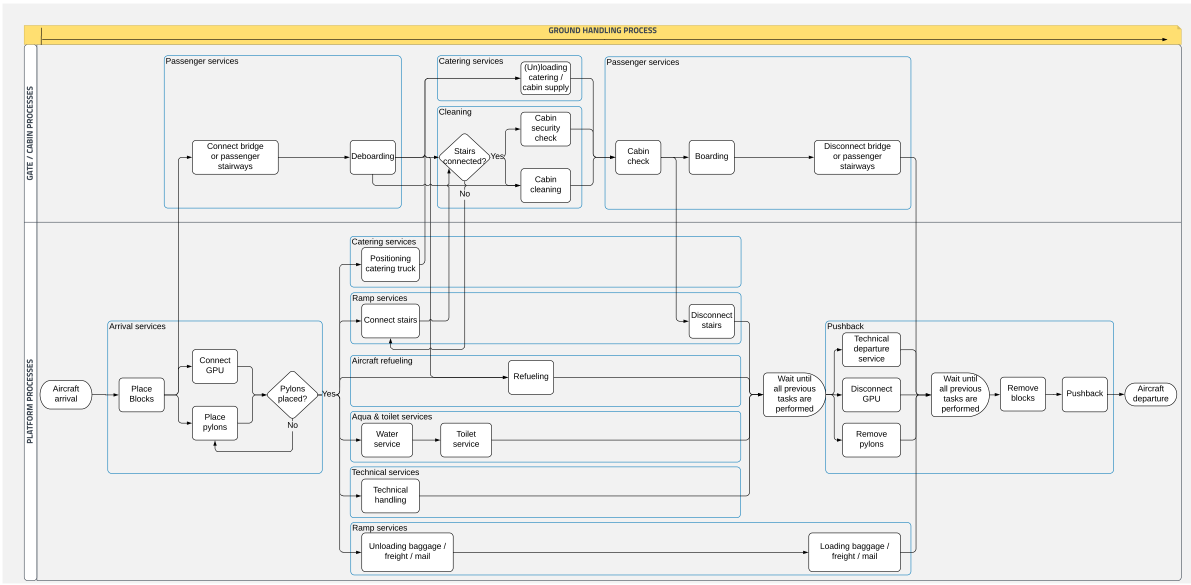

Figure 3.3 Flow chart ground handling process.

3.3 Aircraft Turnaround Processes

The aircraft turnaround processes which fall under flow control, the software that is used and the way the controllers schedule the tasks:

- Aircraft Refueling (CHIP → optimizer)
- Catering Services (CARE → manually)
- Aircraft Towing & Pushback (CHIP → optimizer)
- Cleaning (CHIP → manually)
- Water & Toilet Services (CHIP → optimizer)
- Ramp Services (CHIP → manually)

For each process the events that have to be performed by the operator are listed and how they enter this in their Hand Held Terminal (HHT), and how the controller sees this in CHIP. In case of disturbances the type of disturbance is mentioned for each process in the disturbance field of the HHT. Each section will end with possible methods that can be used in case of disturbances of the original schedule. Possible reasons that can cause disturbances are flights that arrive early or are delayed, equipment failure, or shortages of people and equipment.

The Duty Flow Managers are responsible for the daily operation of all processes and intervene if necessary and in case of major disturbances. For a detailed work instruction of the different processes the reader is referred to the "Werkplekinstructies" (WPI's) which can be found on the website of KLM [KLM, 2018c].

For each process the following events and the time at which they took place can be seen by the controllers in CHIP:

- Task assigned to operator by controller.
- Operator accepts task.
- Operator starts driving.
- Operator arrives at location.
- Operator enters the aircraft registration.
- Operator starts the task.
- Operator processes data in the HHT.
- Operator completed the task and is ready for next assignment.

Swim Lane Analysis

Swim lane analyses are used to get a better insight in the different processes. A swim lane analysis is a powerful tool to analyze how different stakeholders are involved in different parts of the process [Damelio, 2011]. At the end of each process the swim lane analysis for that process will be given.

3.3.1 Apron

Figure 3.4 shows the swim lane analysis for the apron. The swim lanes show the many different departments that are involved to schedule the different process tasks. The two bottom swim lanes show the apron processes (BTS and AS) that will be analyzed in this research. The orange boxes can be opened to see what happens inside of them, and will be further discussed in sections 3.3.2 to 3.3.7. It can be seen that the apron processes each have their own box, and there is only minimal integration between these boxes. Also, it can be seen that the decision to delay flights is not taken by the controllers of the processes, but by the HCC and OCC. Controllers do not have enough information about the flight network, and the corresponding costs for a delay to make such decisions.

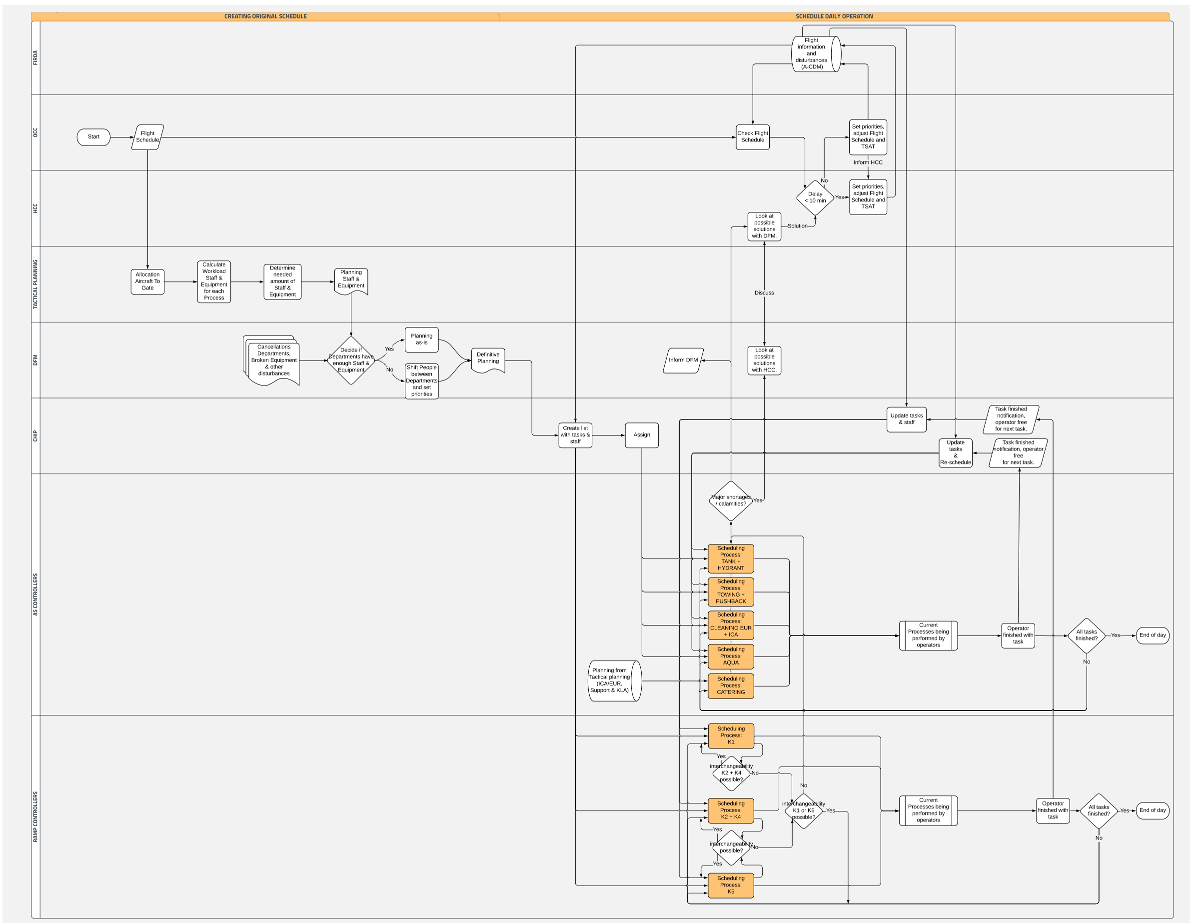

Figure 3.4 Swim lane analysis for the apron.

3.3.2 Aircraft Refueling

The fueling department has the task to refuel the aircraft with a specific volume which is based on: flight destination, weight and expected weather conditions. The department consists of 100+ staff, and pump no less than 2.5 million m^3 of Jet A1 into more than 120.000 aircraft each year [KLM, 2018a]. Refueling can be done using dispensers or bowsers. Most of the gates at AAS are connected to the underground hydrant fueling system which makes it possible to only use a dispenser at those gates. KLM has 21 dispensers [KLM, 2018a] that pump fuel from the airport fuel system into the waiting aircraft.

The refueling department can start refueling when all passengers are deboarded from the aircraft. Both the bowser and dispensers are operated by a single operator which gives updates on his HHT at certain events in time. Hydrants and non-hydrants (bowsers) are controlled by two different controllers, who can interchange equipment and staff if necessary. Figure 3.5 shows the swim lane analysis with all decisions that a controller can make to prevent tasks from not being able to get scheduled or ways to make up time if necessary.

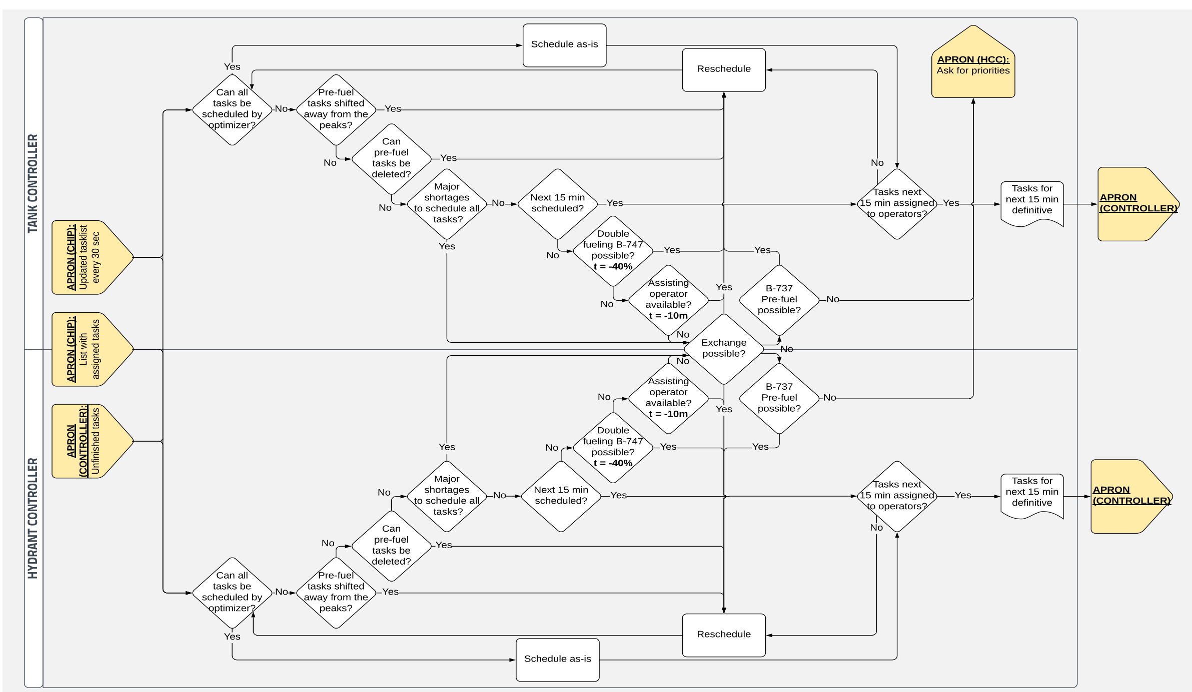

Figure 3.5 Swim lane analysis for the controllers of the refueling process.

3.3.3 Catering Services

Catering of the aircraft is outsourced to the catering company KCS, a daughter company of KLM. Therefore KCS works fully autonomous from other GS processes and is responsible for restocking all catering products and taking the waste from cleaning. There are 82 catering trucks that serve 55.000 meals for 350 intercontinental, European, and cargo flights each day [KCS, 2018]. Depending on the aircraft, one truck can fill one (large *ICA* flight) or a maximum of four (small *EUR* flight) aircraft.

Catering can start restocking the aircraft when everyone is deboarded from the aircraft or under certain circumstances it can start during deboarding and with permission of the crew. The catering truck is connected directly to one of the doors of the aircraft, leaving only small margins for the truck to maneuver to the right location. Most delays are the result of blockage from wrongly placed dollies, tank dispensers, GPUs, and other equipment which should not have been there in the first place. A supervisor on the platform is continuously checking if there are any blockages and tries to resolve them before the catering truck arrives at the platform. Figure 3.6 shows the swim lane analysis with all decisions that a controller can make to prevent tasks from not being able to get scheduled or ways to make up time if necessary.

If one catering truck has to fill multiple aircraft, the "positioning truck" and "start task" events get repeated for each catering task. Because one truck can fill multiple aircraft in a row, and each aircraft has its own specific board supply, disturbances in the flight schedule can have a big impact on future catering tasks. Also, each truck which has been loaded has to go through the security check before it can enter the platform which results in extra driving time, making a good planning essential. Note that KCS uses their own tool (CARE) instead of CHIP for assigning the tasks. This means less data availability and less integration with the other processes.

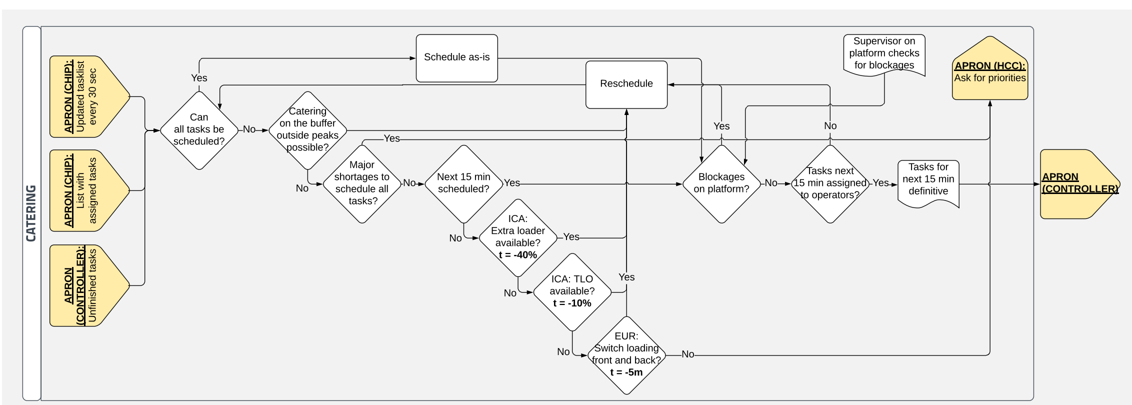

Figure 3.6 Swim lane analysis for the controllers of the catering process.

3.3.4 Aircraft Towing & Pushback

This department is responsible for the pushback of departing aircraft and the moving of aircraft between gates, hangars and buffers. KLM Aircraft Towing & Pushback Services handles 260 pushbacks a day, which adds up to 96.000 a year [KLM, 2018b]. To boost efficiency and save on manpower, the conventional tow bar tugs are replaced by tow barless tugs which are more flexible, economical and can be operated by a single person. KLM has 39 one-man tugs and eight conventional tugs and all KLM drivers are fully qualified to perform part of the pre-flight inspection, saving valuable time and money [KLM, 2018b].

The pushback of the aircraft is the last step in the TA processes and starts when all other processes are finished. The pushback task is scheduled based on the TSAT time, which can change at the last moment. This makes it difficult to schedule tasks ahead, and assigning tasks to operators is therefore often done by the built in optimizer of CHIP at the last possible moment. The scheduling of towing tasks can be improved, but towing tasks have to take all other processes into account when assigned to an operator. The two departments work closely together and interchangeability between the two happens a lot to divide the workload between the two departments. Figure 3.7 shows the swim lane analysis with all decisions that a controller can make to prevent tasks from not being able to get scheduled or ways to make up time if necessary.

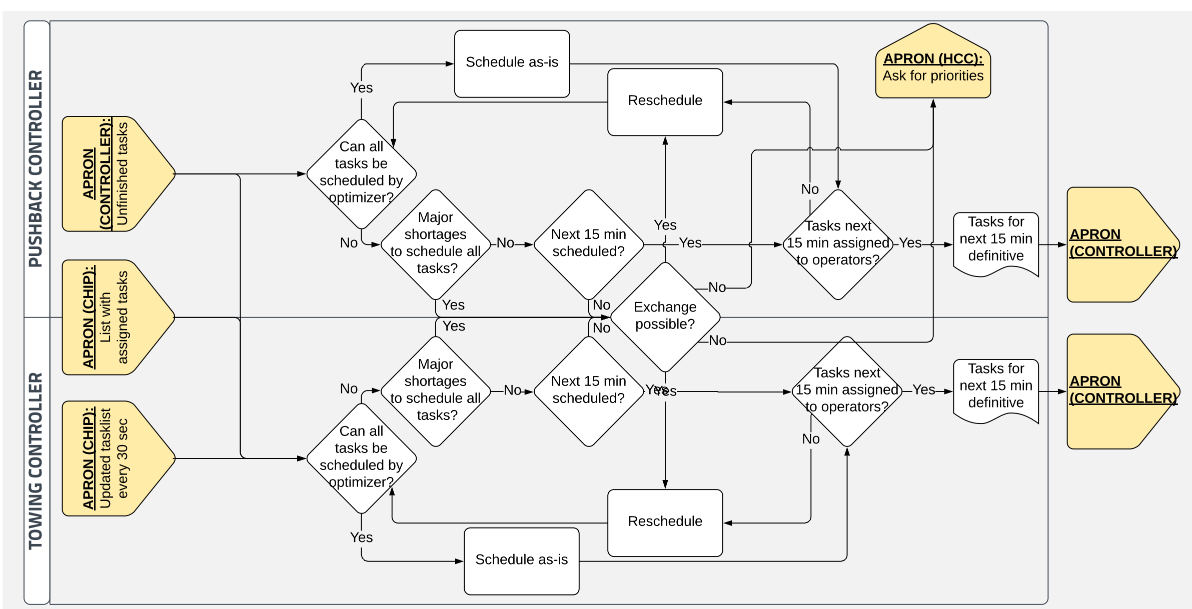

Figure 3.7 Swim lane analysis for the controllers of the pushback and towing process.

3.3.5 Cleaning

Cleaning is outsourced to two external companies Klüh (*EUR flights*) and Asito (*ICA flights*), which are responsible for the cleaning of the cabin and toilets. This means the removing of traces of food, packaging waste, used newspapers, and forgotten belongings from the aircraft [Klüh, 2018]. They are also responsible for the changing of the seat covers and clean baggage bins and aisles [Klüh, 2018]. For both Klüh and Asito, the team sizes are between 3 and 4 people. Because Klüh is responsible for *EUR* flights and Asito for *ICA* flights, interchangeability between the two is not possible. Both companies work with CHIP but because not only KLM flights are scheduled, the optimizer of CHIP does not work well and tasks have to be scheduled manually.

The cleaning department can start its activities immediately after all passengers have deboarded the aircraft. Often there is a gap between these two moments because the cleaning crew reacts only after the last person deboarded the aircraft, thus losing valuable time.

Note that Klüh and Asito are external companies which means there is less data availability, and less information about the standard cleaning procedures making integration with other processes more difficult. Figure 3.8 shows the swim lane analysis with all decisions that a controller can make to prevent tasks from not being able to get scheduled or ways to make up time if necessary.

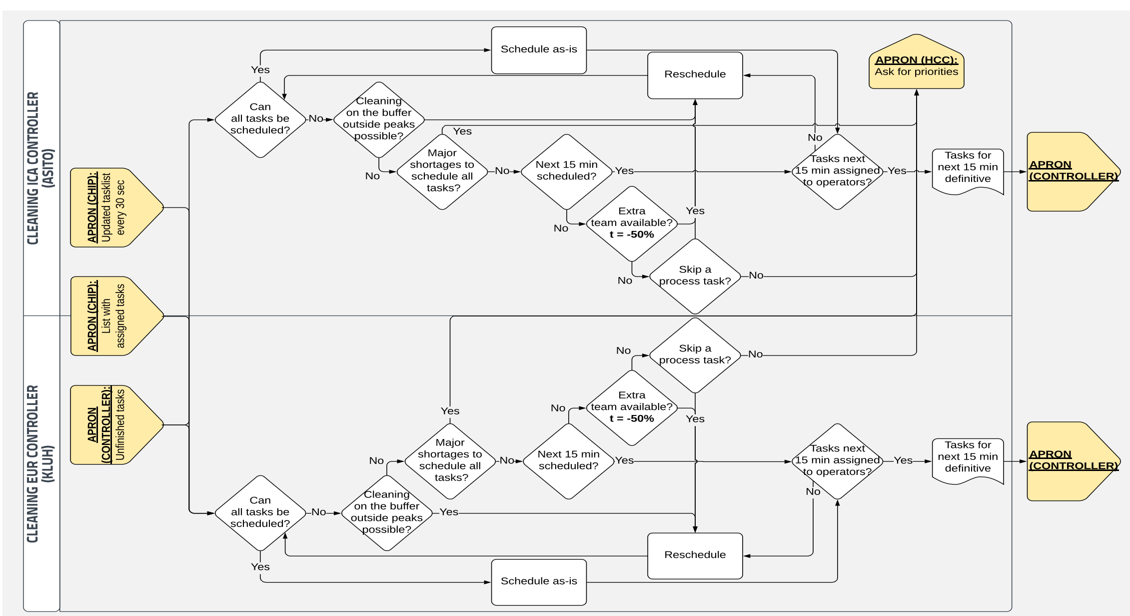

Figure 3.8 Swim lane analysis for the controllers of the cleaning process.

3.3.6 Water & Toilet Services

Water & toilet services is responsible for the filling and draining of fresh water supplies and toilet services. Also, it is responsible for jet-starter equipment for aircraft without serviceable Auxiliary Power Unit (APU) and the delivery of air-conditioning units that are used to cool and heat aircraft on the ramp [KLM, 2018g]. Still, the core activities are the supply of potable water to the aircraft and the emptying of the toilets. water and toilet services do not depend on other aircraft TA processes, allowing for a wider time-window in which the tasks can be performed. The two processes can be performed in random order but are always performed separately for hygienic reasons. The trucks work separately, and interchangeability is possible, although some hygienic measures have to be taken when an operator from toilet services wants to switch to water services. Figure 3.9 shows the swim lane analysis with all decisions that a controller can make to prevent tasks from not being able to get scheduled or ways to make up time if necessary.

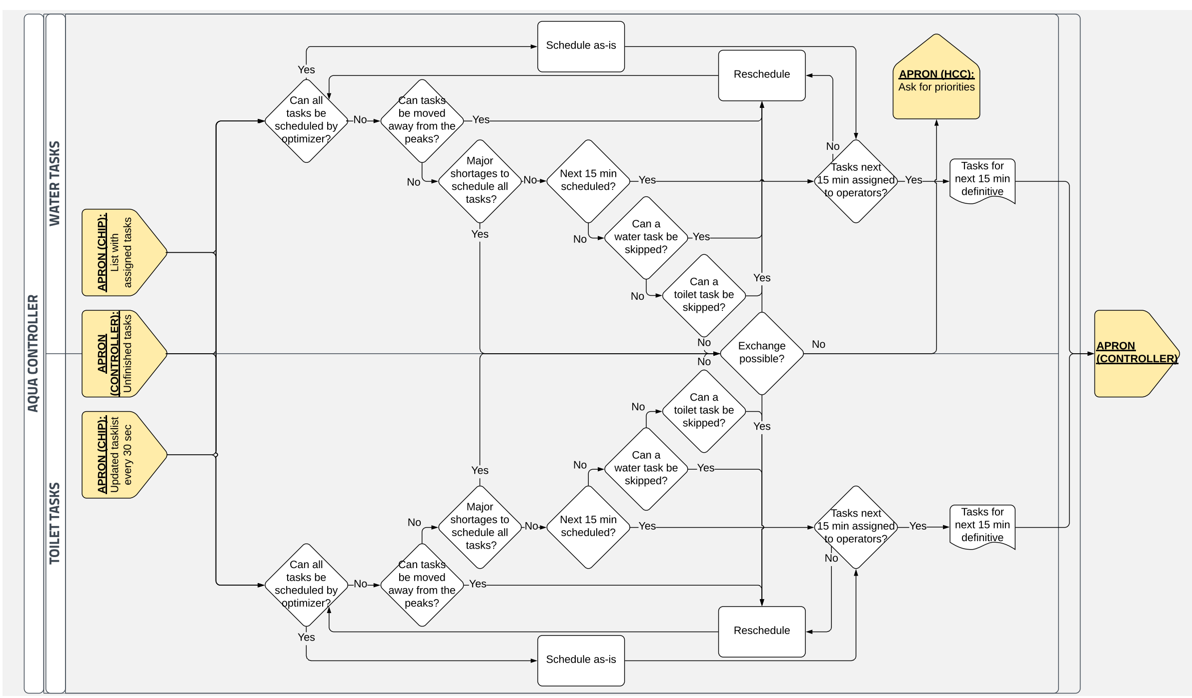

Figure 3.9 Swim lane analysis for the controllers of the water and toilet process.

3.3.7 Ramp Services

Ramp Services is responsible for the loading and unloading of baggage & cargo and used to be a separate department at KLM. The department is split into the following units, which all have their own responsibilities and way of working:

- **K1 -** Responsible for KLC, which is located at the A- and B-pier and B-platform. K1 works in fixed teams with a Team Captain (TC) that is part of the team. The team captain uses the Apron App which shows information about which tasks to perform, the load sheet, and all other necessary information that is used to handle the aircraft. The team captain is responsible for the communication of this information to the Team Members (TM's) and it's the only unit which is responsible for its own equipment.
- **K2 -** Responsible for all Europe and Transavia flights which are located at the C-pier. Teams consist of one team captain and one or two couples of two people, depending on the load of the aircraft. The teams are flexible, and team captains and couples can circulate between different teams. Different than for K1, K2 is not responsible for its own equipment, but there are multiple "top drivers" who are responsible for the equipment at the apron.
- **K4 -** Responsible for all Europe flights at the D-pier. The same team structure as for K2 is applicable here, and just like K2 equipment and staff are also separated.
- **K5 -** Responsible for all Intercontinental (ICA) flights which are spread over the D-, E-, F-, and G-pier. Teams consist of five operators and one team captain. The teams captains are not bounded to a team so they can switch between teams if necessary. Just like K2, equipment and staff are separated.

The equipment that can be used by the team members at the ramp depends on the type of cargo the aircraft contains. The different types of equipment are [KLM, 2018e]:

- Loaders; are used for the loading and unloading of pallets and containers.
- Transporters; KLM has three types of transporters from: ERMA, MULAG and TLD. A transporter is a vehicle that transports containers and pallets between the dollies and pallet vehicles or the main and lower deck loader.
- Towing tractors; are used to move carts, containers, dollies, GPUs (ICA, EUR, K1) and pallet trucks. KLM uses MULAG towing tractors.
- Conveyor belts; are used to for the loading and unloading of baggage, freight and mail. There are two types used by KLM: TLD (electric) & TUG (diesel), and the front and rear of the conveyor belts is height-adjustable.

The loader and transporter are only needed when containers or pallets have to be loaded or unloaded from the aircraft. This is the case at K5, where large ICA aircraft have to be loaded with a lot of cargo. The towing tractors and conveyor belts are always present at each unit to load loose bags in the belly of the aircraft. The equipment is used by the team members, and each team consists of a combination of people with different skill sets:

- Team Member Large Material (TGM); can operate all equipment: Mulag, conveyor belt, main deck loader, transporter.
- 1^{st} loader; can perform the same tasks as a TGM, except operating the loader.
- Temporary worker; can perform the same tasks as the 1^{st} loader except the towing tractor.
- B-skills; can only operate the conveyor belts.

Each team consists of at least one TGM and 1^{st} loader to operate the loader and tractor. Depending on the aircraft, the size of the teams can vary between two and five people.

The team members are called via a speaker in the general area. No feedback is given if the assigned tasks are accepted by the team members, and the transceiver they are supposed to use is often not registered to the corresponding team member. The team captains use the apron app which in turn gives status updates of the different processes. The team captain communicates this information with his team members and coordinates the processes on the "VliegtuigOpstelPlaats" (VOP). Figure 3.10 shows the swim lane analysis with all decisions that a controller can make to prevent tasks from not being able to get scheduled or ways to make up time if necessary.

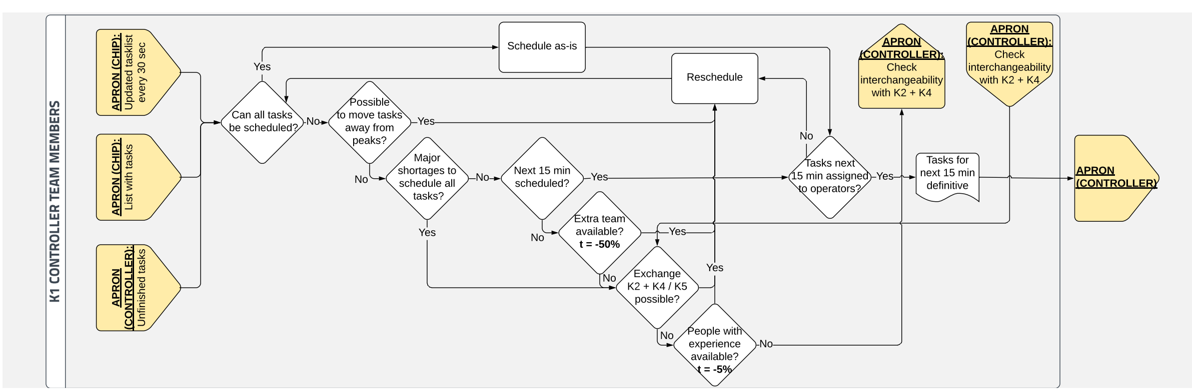

Figure 3.10 Swim lane analysis for the controllers of the ramp K1 process. The swim lanes for K2, K4 & K5 controllers can be found in appendix A.

3.4 Information Systems

The main partners of the CDM@AMS program are KLM, Schiphol and LVNL, each with their own information system:

- **KLM** - *Flight Information Royal Dutch Airlines (FIRDA)*
- **Schiphol** - *Central Information System Schiphol (CISS)*
- **LVNL** - *Amsterdam Advanced Air Traffic Control (Triple-A)*

These can also be seen in figure 3.11. The CDM project interconnects between these systems to get the most complete, accurate and up-to-date information when making decisions. This information can then be used to schedule the ground processes more accurate and efficient. Flight information of all aircraft and their movements are provided by the information systems, and for each partner the main information systems will function as the data source for all other applications.

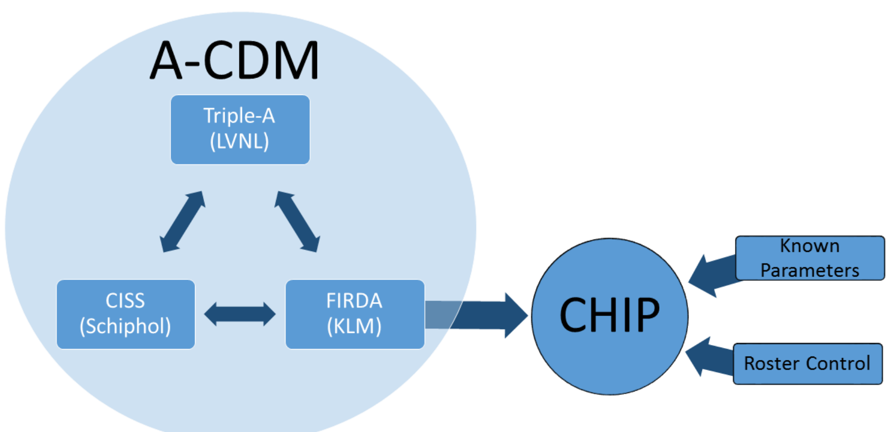

Figure 3.11 Information Sources A-CDM and CHIP.

3.4.1 CHIP

KLM uses the program CHIP as their scheduling tool for their ground operations. CHIP is built by INFORM[1] and is a decision support tool which helps the controllers in finding the most optimal sequence of tasks. It automatically assigns tasks to the employees, but the controller can overrule these decisions if necessary.

CHIP's interface shows the operators on the y axis and the time line on the x axis. The tasks are represented by blocks which have to be assigned to the operators. There are a lot of restrictions when dispatching jobs to operators like: flight schedules, time windows, breaks, shift roster limitations, and used equipment for example. There can be many reasons why not all tasks can be scheduled by the optimizer due to the strict constraints. The controller then has to make concessions and make a decision on what to do.

Information Sources CHIP

Figure 3.11 shows the input sources of CHIP:

- *Known Parameters* - such as: distances between gates, equipment, employees and their qualifications, aircraft types, and airlines are used to define the operational environment. These are known parameters which have a static behavior and do not change during normal operation.
- *FIRDA* - gives actual flight information such as: arrival and departure times, delays and the turnaround time. FIRDA updates continuously, and the data provided by it therefore has a dynamic behavior.
- *Roster Control* - provides information such as: the number of available employees and the shifts, restrictions and breaks they have.

The Optimizer

Some of the processes (see section 3.3) already have a built-in optimizer which schedules most of the tasks. CHIP's optimizer can schedule the tasks using these information sources. Tasks are continuously updated, created and deleted based on real time information from the input sources. The data that is used by the optimizer to assign tasks to the operators:

[1]More information can be found on: https://www.inform-software.com/

- Duration
- Travel time
- Time windows
- Task requirements
- Priority
- Task type
- Teaming
- Workload

The optimizer from CHIP schedules all tasks in a four hour time window. The optimizer looks for combinations of tasks and resources, taking all constraints intro consideration. Based on the data that is used by the optimizer, each task gets multiplied by a weight factor in the cost function. The optimizer tries to minimize a cost function by assigning all tasks to the different resources. The system owner uses predefined parameters which are used to configure the cost function.

Push-Pull Strategy

A known flight schedule and the resulting tasks are predetermined allowing for employees and equipment to be scheduled long before the day of operation. Together with minimum input from the customers this shows similar characteristics with push strategy when looking at the literature.

On the other hand the day of operation is extremely dynamic. The airline industry operates on a global level and disturbances all over the world can influence the schedule here in Amsterdam. This results in a very uncertain schedule on the day of operation and continuous shifts in the schedule. These are characteristics which belong to a pull strategy.

This combination shows a hybrid push-pull strategy with characteristics of both strategies. This strategy is currently used by each individual process, where tasks for each aircraft are planned by that process. This means different planning strategies can be used by different controllers when planning their tasks for the same aircraft. This can result in unnecessary waiting for other processes to finish, which can even result in the delay of multiple aircraft when different processes choose different aircraft tasks to delay. A more integrated approach is desirable from the perspective of the aircraft in order to pull all resources that are necessary.

3.5 Alternative Control Scenario

An alternative control scenario needs to realize a more integral control between the different silos. To do this, the processes will communicate with an umbrella body, the TLCT, instead of with each other. Figure 3.12 shows how the TLCT would operate and communicate with the other processes. The arrows and processes with a green color show the changes that are made compared to the current state as seen in figure 3.4.

The TLCT is used as an umbrella body that is responsible for all the communication and control between the different processes. It uses the data from different data sources and combines this information into a structured way that is understandable by all processes. If one of the processes has difficulties scheduling its tasks it will communicate this with the TLCT and look for possible solutions. The found solution will then be communicated to all other processes at the same time,

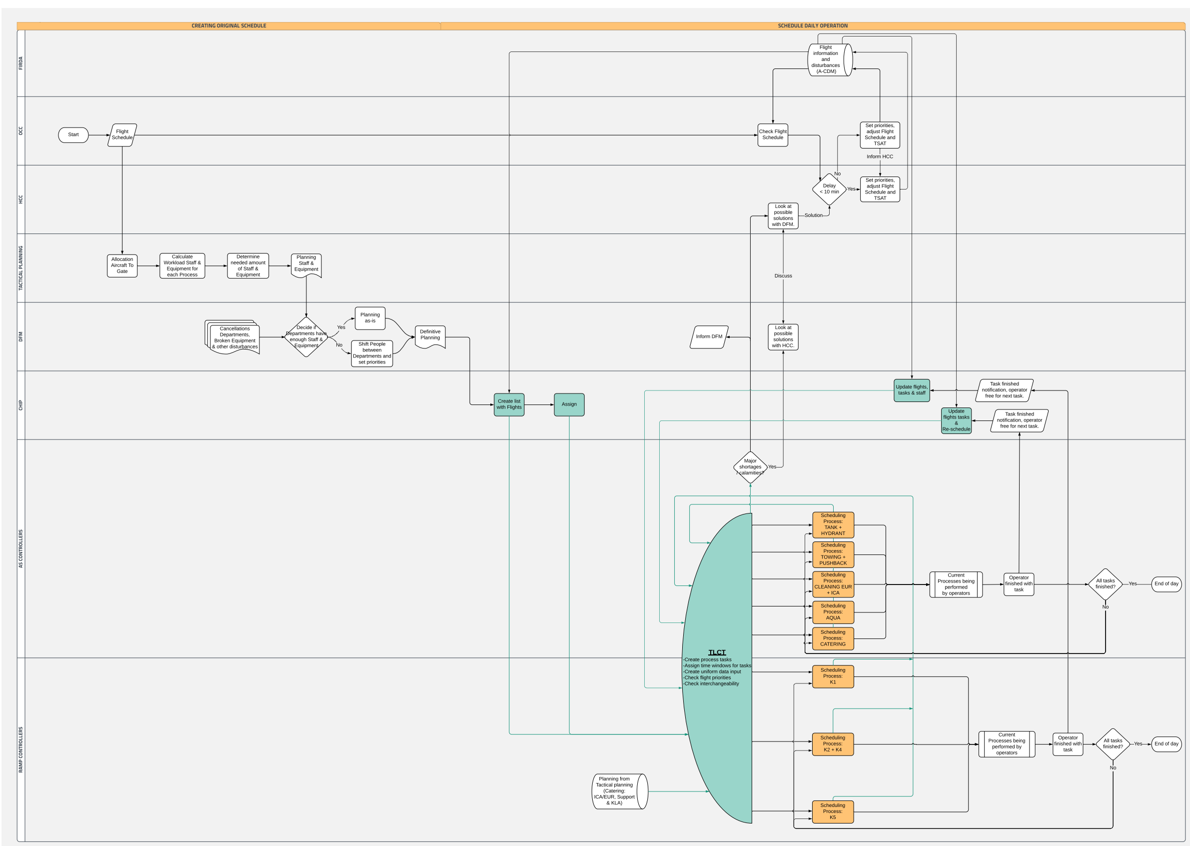

Figure 3.12 Swim lane analysis for the alternative control scenario for the apron.

allowing all processes to have the same information at the same time. The TLCT will use the arrival and departure time of the aircraft as input and will determine the time windows for the corresponding processes based on these times. Delays or disturbances in the turnaround process will be communicated to the TLCT, which will automatically update this information for all other processes as fast as possible. A pull strategy will be used for all arrival oriented processes to schedule tasks as early as possible to maximize the slack time for each processes.

Within the orange boxes not much changes, processes still operate as they used to do and make the same decisions. The only difference is their communication with the other processes and departments. All communication will go via the TLCT, which will spread this information fast and clean to all the other departments. This way no information is lost due to bad communication between the departments.

The TLCT will also be responsible for the exchange of workers and equipment between the different processes. Controllers tend to schedule their own tasks as good as possible, building in some slack between tasks for any unforeseen disturbances. Controllers get assessed based on how they schedule the tasks of their own process, and not on the number of flight delays. If other processes need extra resources, a controller will only provide these resources if they are absolutely sure they will be able to schedule their own tasks in case of any disturbances. The results is that the process performs good, but the overall turnaround performs not optimal. A TLCT has a better overview on the integral turnaround picture, and can intervene in an earlier stage forcing certain processes to exchange resources if necessary.

3.6 Conclusion on Current State Analysis

This chapter gave an answer to the first two sub-research questions:

1. *What is the current state practice of KLM's apron processes?*
    - *What is the current operational practice: process steps, information flows, decision rules, operational control and execution.*
2. *What alternative scenarios can be used to control the apron processes?*

The first sub-question is answered by the current state analysis and showed us that KLM has very limited insight in the integral picture of their control of the apron processes. The swim lane analysis showed us that the system is highly complex, non-transparent and has many input variables.

Given the current state of the process, the second sub-question can be answered and the following alternative scenarios need further research:

- The use of an umbrella body in the form of the introduced TLCT, that will be used for the communication between the processes.
- A pull strategy where aircraft pull the necessary resources from a global pool of resources.
- Varying the time before workers move to their destination.
- Different control scenarios for workers to choose their next tasks.
- Use dynamic flight priorities instead of the current static method.

CHAPTER

4

DATA ANALYSIS AND KEY PERFORMANCE INDICATORS

This chapter will discuss the availability of data in the different databases of KLM, and the data quality. An in depth analysis of the available data generated by the apron IT systems has been performed. This data is used to track the performance of the current system, and assess different control scenarios. KPIs are used to track efficiency and effectiveness of alternative scenarios. This is done for the overall performance of the processes as well as the individual sub-processes. The third sub question is answered in this chapter:

3. *How is the process currently monitored: data availability, KPIs?*

4.1 Data Sources

Various data sources are analyzed and merged to create input that resembles the actual process. Data is currently stored and managed at different, independent departments and locations within KLM. An in depth analysis of the available data generated by the apron IT systems has been performed. Overall, data is gathered in the Hub Data Base (HubDB). Figure 4.1 gives an impression of the data

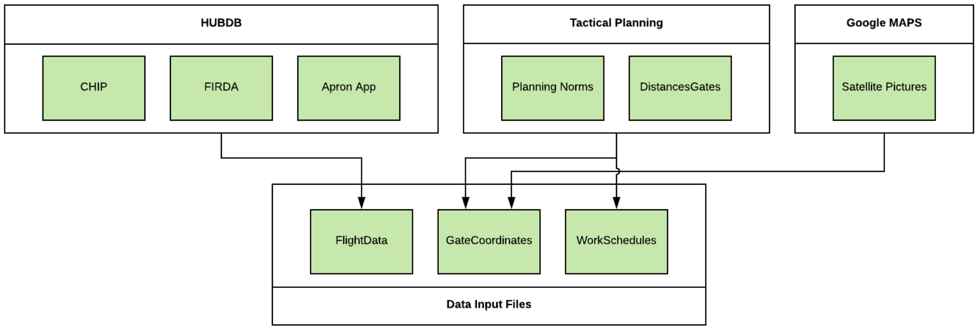

Figure 4.1 Data & IT landscape KLM.

& IT landscape of KLM.

As described in the introduction, the department flow control is a combination of AS and BTS. Together with passage, their main task is the TA of all aircraft. Before this merger, each department used to work using their own methods and standards, resulting in big differences in data availability and data quality. For example, AS works with terminals and operators get forced to press a button for the next step before they get certain information which is needed to perform their task. Also, most AS processes can use a built-in optimizer which automatically logs all process steps within CHIP. For BTS this is completely different. Only the team leaders use a terminal, passing information to their team members using verbal communication. Almost all communication with the controllers is done using transceivers, making it hard to track personnel. Their control department does not use the optimizer, and often events in the terminal are skipped by the operator. This is the reason why essential information is not always logged, resulting in negative time duration or "NULL" values. Because of these reasons, big differences in the quality of the necessary data can be found in the database. The data from the databases is reviewed, together with data analysts from KLM, GroundStar and CHIP. The percentages below are used as a rule of thumb and represent the amount of data within the database which is reliable and correspondents with the real system.

- Aircraft Services: 80%
- Passage: 60%
- Baggage Turnaround Services: 40%

Section 4.3 will describe how the raw data is converted to workable data that can be used for the analysis and as input for the simulation model.

4.2 Current Performance of the Actual System - KPIs

This section discusses the KPIs to track the performance of the apron process. The KPI criteria discussed in chapter 2 will be used to evaluate current and future key performance indicators.

4.2.1 Delays and Costs

The two most important KPIs are the number of delays and their corresponding costs. A more efficient and effective system results in lower operational costs which will eventually determine whether another scenario will be implemented. As determined in section 3.2.1, delay costs are different for each flight and strongly correlated with the length of the delay. This means it is very important to take these costs into account when choosing which tasks to perform first and which flights to delay. By adding costs to each delay, different delays can be compared, and it can be determined how important certain flights are. The costs are therefore used to combine other KPIs into a single KPI that is used by the objective function of the simulation optimizer.

Definition OTS, OTP & OTD

To keep track of why certain flights delay, and which processes are causing these delays, the On Time Start (OTS), On Time Delivery (OTD), and On Time Performance (OTP) as described by Beelaerts are introduced. They are used to zoom in on the processes and see which processes cause the most turbulence in the TA process. Figure 4.2 graphically represents the OTS, OTD & OTP.

- **On Time Start (OTS)**
  The OTS is defined as the earliest time a task can start. This is represented in CHIP by the earliest start time. If a process starts later, the OTS is not achieved.

- **On Time Delivery (OTD)**
  The OTD is the latest time a task can be finished before it will cause delays, making it an important KPI.

- **On Time Performance (OTP)**
  The OTP is the time that is necessary to perform a task. This time is calculated by CHIP keeping in mind all parameters which can influence the length of a task. If a task takes longer than the predetermined time in CHIP, the OTP is not achieved.

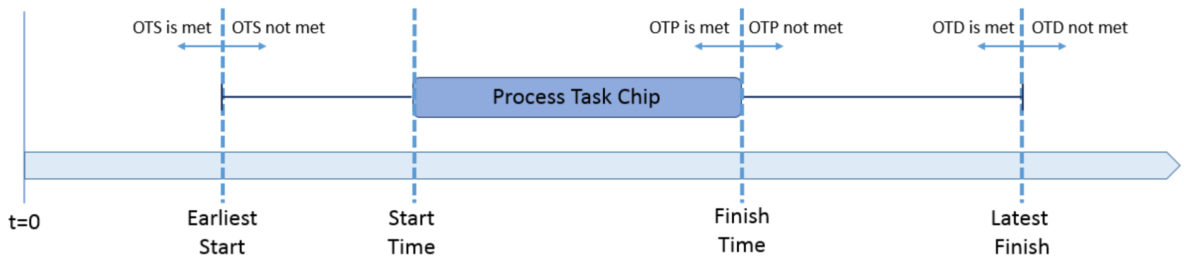

Figure 4.2 The OTS, OTP and OTD graphically illustrated in a timeline.

4.2.2 Effectiveness

Effectiveness is the capability of producing a desired result or the ability to produce desired output. When something is deemed effective, it means it has an intended or expected outcome, or produces a deep, vivid impression [dictionary.com, 2018].

To track overall effectiveness, the ratio between the number of times the OTS, OTD, and OTP are achieved are compared to the total number of tasks. This can be done for the overall TA process (the Standard Process Time (SPT)), as well as each individual TA process and is calculated using the formula:

$$OTS_{ratio}/OTD_{ratio}/OTP_{ratio} = \frac{OTS_{NotAchieved}/OTD_{NotAchieved}/OTP_{NotAchieved}}{NumberofTasks} \tag{4.1}$$

The lower the ratio the better the effectiveness of the process. The rate can be improved by minimizing the number of times the OTS, OTD, and OTP are not achieved. This means that improving the control of the apron processes has a positive effect on the OTS_{ratio}, OTD_{ratio} and OTP_{ratio}. Also increasing the number of flights has a positive impact on the ratios, but in this research the number of flights is constant. There can be multiple reasons why the OTS, OTP or OTD is not achieved, and it is therefore important to log this and find which processes cause the most turbulence in the TA process.

4.2.3 Efficiency

Efficiency is the ability to do things well, successfully, and without waste [Longman, 2018]. In more mathematical or scientific terms, it is a measure of the extent to which input is well used for an intended task or function (output) [wikipedia.org, 2018].

To measure the efficiency of the TA processes, the time a resource is used is compared to the time the resources is available. The following formula is used to measure the efficiency of personnel and equipment:

$$\eta_{resource} = \frac{t_{ResourceUsed}}{t_{RescourceAvailable}} \tag{4.2}$$

An increase in efficiency does not automatically result in a decrease in the amount of necessary personnel and equipment due to under utilization during rush hours caused by the 7-wave network at Schiphol which is described in section 1.1.1. However it does save capacity of resources in rush hours, making it possible to do more recovery handling and thus increase effectiveness.

4.3 Data Analysis

To measure the performance of KLM data from *July and August 2018* is chosen. These are the summer months which are the busiest months of the year, laying the emphasis on the efficient and effective use of available resources. This makes it suitable as a reference to benchmark the efficiency and effectiveness when trying other scenarios.

4.3.1 Overall Performance

When the aircraft arrives, an Estimated Off-Block Time (EOBT) is calculated to the aircraft based on the Minimum Ground Time (MGT) and SPT, resulting in the available ground time for the aircraft.

The definition of the SPT is given by KLM as the Ground Throughput Time Amsterdam. KLM gives the following description of this: If an incoming flight is delayed, or the aircraft availability is not on time (after a towing delay or registration change, for example), as a result of which the Current Ground-time < MGT, then the ground time is stretched to the MGT (SPT standard for turnaround or SPT standard for departure handling). The planned doors closed moment is corrected for the turnaround time calculation. If the current All Doors Closed (ADC) <= corrected planned ADC, then the SPT is realized [KLM, 2018g].

A target is set for the SPT by KLM based on historical data and the GOMS, which is reviewed every day to check the performance. Table 4.1 shows the results for the months July and August 2018 as calculated by KLM:

The percentages show how often the SPT is accomplished compared to the total number of flights. The percentage are quite low, and not achieved in 33,3% of the time leaving a lot of improvement here.

Looking at the CHIP data, an available ground time is given and an actual ground time based on the arrival and departure time. The available ground time has 5 min slack due to the fact that the

Table 4.1 The SPT results from the HPM monitor for July and August 2018.

Week	Week 26	Week 27	Week 28	Week 29	Week 30	Week 31	Week 32	Week 33	Week 34	Week 35	Total	Targets
Days	1	7	7	7	7	7	7	7	7	5	62	N/A
ICA	33.9%	46.6%	55.3%	64.9%	35.4%	50.9%	44.1%	44.9%	50.3%	45.6%	48.5%	45.0%
EUR	51.5%	62.8%	63.7%	60.0%	52.3%	67.9%	55.7%	64.9%	63.0%	59.9%	61.0%	60.0%
Commuters	35.0%	64.4%	61.3%	63.4%	54.4%	70.7%	58.9%	65.9%	68.6%	68.5%	63.4%	60.0%

TSAT time has a time window of ±5 min. Table 4.2 shows the SPT results by comparing the available time with the actual time on the ground:

Table 4.2 The SPT results based on the available and actual ground time for July and August 2018.

Week	Week 26	Week 27	Week 28	Week 29	Week 30	Week 31	Week 32	Week 33	Week 34	Week 35	Total	Targets
Days	1	7	7	7	7	7	7	7	7	5	62	N/A
ICA	45.3%	53.9%	57.5%	43.8%	37.4%	46.3%	44.8%	47.6%	50.5%	48.1%	47.7%	45.0%
EUR	46.4%	60.6%	55.3%	51.8%	53.5%	64.6%	53.3%	61.9%	60.2%	59.3%	57.6%	60.0%
Commuters	46.5%	65.2%	65.6%	64.8%	57.2%	64.9%	59.2%	70.2%	66.5%	68.1%	64.2%	60.0%

In general the found percentages have similar results except for some small differences. Data from the database is unfiltered and applying the calculations as defined by the definition of KLM, similar results should be found. KLM uses its own filters and corrections which take more external disturbances into account compared to the calculations performed on the data from the HubDB. This could be a possible reason for the small differences with the data from the database. Another reason is the fact that aircraft sometimes have to wait for an available time slot to depart when all TA processes are already finished. The EOH time is then achieved, but the SPT not because the aircraft has to wait at the gate before it can depart.

To determine the delays of the aircraft in the model, a calculated departure time will be used based on the arrival time + the available ground time. The data for the available and actual ground time is available in the HubDB for each flight and uses ground times which are determined before the aircraft arrives. This is very similar to the real system, where it is often not known which external influences will disturb the schedule beforehand.

Simulation Day

For the simulation day the first of August is used as input data. The SPT results in table 4.1 are used as reference to calculate the number of times the SPT was not achieved on the simulation day. These results can then be used to compare different simulation scenarios with the real system.

There are many variables that influence the delay costs, and complicated algorithms are used by the OCC to calculate these costs. This info is not directly available in the HubDB, and it is therefore difficult to compare results from the simulation model with results from the real system. Costs will therefore only be used to compare different scenarios in the simulation model, and not to compare them with the real system.

Table 4.3 Number of delays on the simulation day.

	Number of Flights	*SPT Percentage*	Number of times SPT not achieved
Total	476	59%	280
ICA	63	48%	30
EUR	234	58%	135
Commuters	179	64%	115

4.3.2 Filtering Data from the Database

The percentages as described in section 4.1 show the necessity of filtering the raw data from the different data sources before it can be used to do the data analysis or use it as data input for the model. In consultation with data experts from KLM and the corresponding departments real start and finish times will be taken since these are always available compared to the high percentage of NULL values and negative duration for the CHIP tasks. These also represent the real duration of the processes under the conditions of that day, making them better suitable to assess different scenarios.

Table 4.4 shows an overview of the applied filters and data source for each process. Table 4.5 then gives the average values for each process sorted alphabetically. In general the values match values from the real system, but some values do look quite high. For example: all boarding tasks and ramp load and turnaround tasks for commuters look higher than expected. Many reasons

Table 4.4 Data assumptions and filters used to analyze the data and create the data input for the model.

Process	*Department*	*Data Source*	*Data filter(s)*
All Processes	All	HubDB	All tasks
Blocks	Ramp	Data Experts	Assumption
Boarding	Passenger Services	HubDB	All PH & ZK Boarding Tasks
Cabin Check	Crew	Data Experts	Assumption
Cabin Security Check	Cleaning	Data Experts	Assumption
Catering	KCS Catering	KCS Database	All KCS tasks
Cleaning	Cleaning	HubDB	All Klüh & Asito tasks (security tasks excluded)
Connecting Stairs	Ramp	Data Experts	Assumption
Deboarding	Passenger Services	Calculated	(1/2)*Boarding Time
Disconnect Bridge	Passenger Services	Data Experts	Assumption
Disconnect Stairs	Ramp	Data Experts	Assumption
Load	Ramp	HubDB	All load tasks
Passenger Bridge	Passenger Services	Data Experts	Assumption
Positioning Catering	KCS Catering	Data Experts	Assumption
Pushback	Pushback	HubDB	All pushback tasks
PylonsGPU	Ramp	Data Experts	Assumption
Refueling	Refuel	HubDB	All "Fin fuel" & "Pre-fuel" Tasks
RemoveBlocks	Pushback	Data Experts	Assumption
TechDepSerPylonsGPU	Pushback	Data Experts	Assumption
Technical Handling	Technical Department	Data Experts	Assumption
Toilet	Toilet	HubDB	All "Toilet Service" tasks, for the B-747 "ex-hangaar" tasks as well.
Towing	Towing	HubDB	Same as Pushback tasks, Towing is excluded from the model.
Turnaround	Ramp	HubDB	All turnaround tasks
Unloading	Ramp	HubDB	All unload tasks
Water	Water	HubDB	Water tasks (Drain tasks excluded)

could cause these high values, and it is most likely that these are caused by measurement errors. If possible, departing oriented processes often get started early to reduce the risk of delaying the flight when disturbances occur during the process. As a result, higher process times can be seen in the data than the time that is actually used. A good example of this is the pushback truck which is often waiting under the aircraft for quite a while before the real pushback can start, resulting in average values of 16 minutes while the real pushback takes a lot less. In some cases tasks even take longer than the total time between the arrival and departure of the aircraft.

Looking at figure 3.3 in the previous chapter, processes are always in series or parallel and do not overlap like often happens in the real system. Also some process of the flow chart like placing blocks and pylons are included in the process times for ramp services for example.

To use this data as input for the model some adjustments have to be made before it can be used as input for the model. This is done in consultation with data experts from the corresponding departments and will be explained in chapter 5.

OTS, OTP & OTD Performance

Some general information about the data set is given in table 4.6. The OTS, OTP, OTD or a combination of them is given for all processes. This section will discuss the key numbers of the data analysis of the different processes.

Table 4.5 Average duration for each process.

Process	*Overall*	*A-330*	*B-737*	*B-747*	*B-777*	*B-787*	*E-175*	*E-190*	*F-100*
All Processes	29	38	30	43	40	38	17	19	19
Blocks	1	1	1	1	1	1	1	1	1
Boarding	54	74	57	80	76	78	31	32	31
Cabin Check	7	9	7	9	9	9	5	5	1
Cabin Security Check	11	14	11	14	14	14	11	11	2
Catering	16	25	10	34	28	32	9	11	6
Cleaning	17	36	12	48	46	45	10	10	9
ConnectingStairs	1	1	1	1	1	1	1	1	1
Deboarding	27	37	28	40	38	39	15	16	16
DisconnectBridge	2	2	2	2	2	2	1	1	1
DisconnectStairs	1	1	1	1	1	1	1	1	1
Load (Ramp)	45	57	32	66	62	53	38	40	40
PassengerBridge	2	2	2	2	2	2	1	1	1
PositioningCatering	1	1	1	1	1	1	1	1	1
Pushback	16	25	18	28	26	24	8	9	12
PylonsGPU	1	1	1	1	1	1	1	1	1
Refueling	24	46	20	56	49	49	16	19	15
RemoveBlocks	1	1	1	1	1	1	1	1	1
TechDepSerPylonsGPU	3	5	4	3	3	3	2	2	2
TechnicalHandling	*23*	*35*	*25*	*35*	*35*	*30*	*15*	*15*	*15*
Toilet	5	13	4	14	10	10	5	5	11
Towing	16	25	18	28	26	24	8	9	12
Turnaround (Ramp)	59	-	61	-	-	-	55	56	58
Unloading (Ramp)	32	32	32	41	39	36	20	22	19
Water	7	27	3	26	31	17	2	2	2

KCS note:
Catering task are performed by the daughter company KCS, which does not use CHIP and has other measurement points compared to the standard measurement points used by CHIP. Because different software is used to log resources and different business rules are applied some inconsistencies on the data could be expected. The following steps are taken to convert these measurement points to the same measurements CHIP uses:

- *OTS;* catering tasks include a driving time. Therefore the driving time should be subtracted from the task before comparing the earliest start and real start.
- *OTP;* for the OTP the norm and real time for first door open and last door closed are compared.
- *OTD;* for the OTD the norm and real time for the last door closed are compared.

A total of 21954 flights can be found with a total number of tasks of 386784 tasks. The tasks take on average 29 minutes with a standard deviation of 24 minutes. The CHIP duration is the time that is calculated by CHIP to fulfill a task and determines how long an operator can take to finish his task. The real duration is based on the finish and start time of the task and is how long the operator really needs to finish a task. The OTS, OTD & OTP are based on respectively the earliest start time, latest finish time and the real duration compared to the duration calculated by CHIP.

Table 4.6 Overview of some general information, OTS, OTP & OTD for all tasks for July and August 2018.

Duration:	**CHIP**	**Real**	**Unit**
Number of Flights:	21954	21954	*[-]*
Tasks	386784	386784	*[-]*
Average	28	29	*[min]*
SD	25	24	*[min]*
KPI	**Nr. of Times**	**Percentage**	
OTS not achieved	307239	79,4%	*(based on earliest start)*
OTP not achieved	248934	64,4%	*(CHIP duration vs real duration)*
OTD not achieved	178276	46,1%	*(based on latest finish)*
OTD not achieved and OTP achieved	36560	20,5%	*(percentage of OTD)*

OTS

The OTS is a measurement to see if the processes starts in time. Ideally all processes start as soon as they can start, which gives them the most built in slack that can be used when issues during the process occur.

Figure 4.3 shows the performance of the OTS for each of the apron processes and compares these for each type of aircraft. High percentages can be seen when looking at the times the OTS is

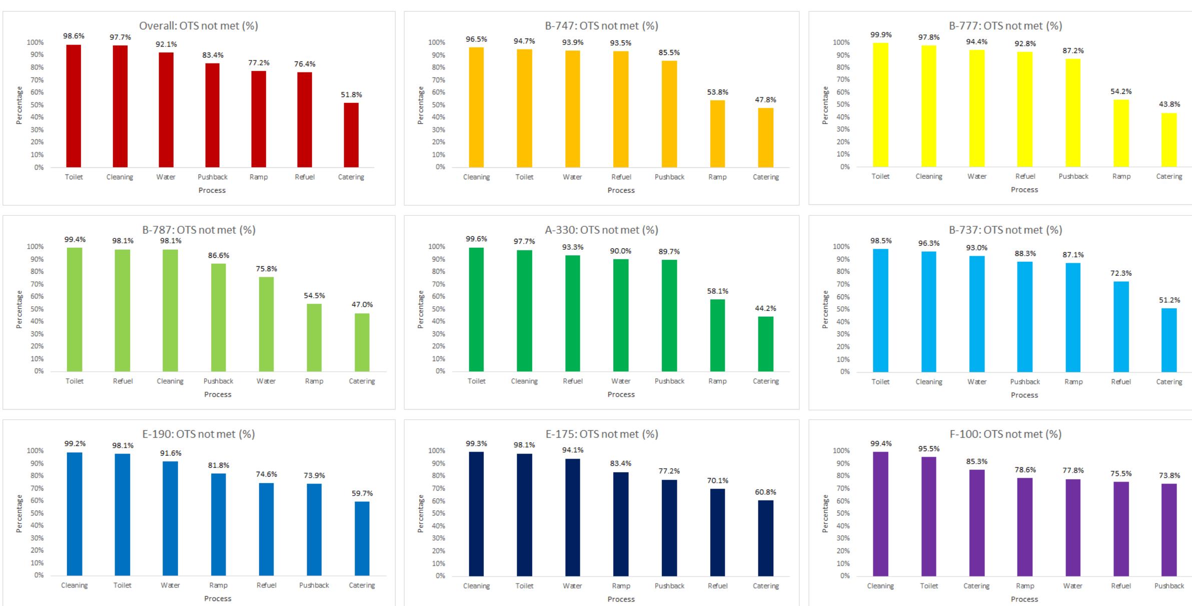

Figure 4.3 Overview of the OTS performance sorted by aircraft type.

not achieved for the processes, meaning processes tend to start later than is possible. Valuable slack time is consumed which could be used by the process in case of extra work, equipment failure or other unforeseen disturbances. Toilet and water services have a lot of built in slack time, giving them more freedom when planning their task. Because of this freedom tasks often do not get planned at the earliest start to prevent unnecessary waiting at the VOP. The result is the bad performance on this KPI. Cleaning on the other hand is a more critical process, but looking at the graphs it can be seen that the OTS is not achieved in almost 97% of the time. Cleaning is dependent on its predecessors before they can start, this could be a possible cause for the high percentages. Another reason could be that CHIP has set the earliest start of cleaning too early, resulting in a bad OTS performance, but a better performance is expected here.

Overall, catering performs the best when looking at the OTS performance. A note should be made that catering does not use CHIP, and therefore it is difficult to compare the catering results with that of the other processes. Catering gets basic task lengths which they have to adjust and schedule manually, allowing for changes in the measurement points. CHIP does all this automatically, using strict rules for the measurement points with no room for adjustments or corrections which could be a possible reason for their lower performance. Looking at the CHIP tasks, ramp services and refueling perform the best, especially for ICA flights. But with the OTS which is not achieved in more then 75% of the times, the performance is still not good. Ramp services have to place the blocks when an aircraft arrives making all other processes dependent on them. It is therefore a critical process, and it is of great importance that they are on time at the right location. Also the baggage of passengers has to get off the aircraft as quick as possible so that the bags get as fast as possible to their owners. This and the fact that ramp services do not have any successors could be a possible explanation for the lower OTS not achieved percentages.

OTP

The OTP is measurement to assess the performance of the process. Therefore the real duration is compared to the length of the task as calculated by CHIP. Figure 4.4 shows the performance of the OTP for each of the apron processes and compares these for each type of aircraft.

The results for the OTP are better compared to the OTS. Overall, ramp services score the worst with 84% of not achieved OTP times, while the water tasks score the best on this KPI. For wide-body and narrow-body aircraft ramp services score the worst, while for the commuters cleaning (Klüh) performs the worst. There could be many reasons processes take longer than expected like equipment failure, wrong information or last minute changes which result in a higher work load. The high percentages for some processes mean that the real duration does not match the duration

Figure 4.4 Overview of the OTP performance sorted by aircraft type.

as determined by CHIP. Wrong assumptions could be made during the calculation of the task times, but a better performance is needed to get a better estimation of the turnaround time.

OTD

If the OTS or OTP is not achieved, other processes are not necessarily influenced by this and flights can still depart at their scheduled departure time. If the OTD is not achieved, other processes will be influenced, and delays in the flight schedule can occur if no time is made up by successive processes. This makes the OTD an important KPI to track. Figure 4.5 shows the performance of the OTD for each of the apron processes and compares these for each type of aircraft.

The graphs show that overall the ramp services score again the worst on this KPI. Looking at the results of the OTS and OTP it is not strange that ramp services also performs bad on the OTD since processes take almost always longer than originally planned. Especially for commuters and narrow-bodies, the ramp services perform significantly worse compared to the other processes. With commuters having many short turnarounds with small error margins, a better OTD is needed for this process to improve the overall TAT.

Looking at wide-body aircraft, pushback performs the worst. Pushback is dependent on all its predecessors as well as the air traffic control tower for an open slot. These dependencies on other processes and external factors resulting in delays decrease the performance of the OTD.

Overall water and toilet services perform the best for almost all aircraft categories. These services

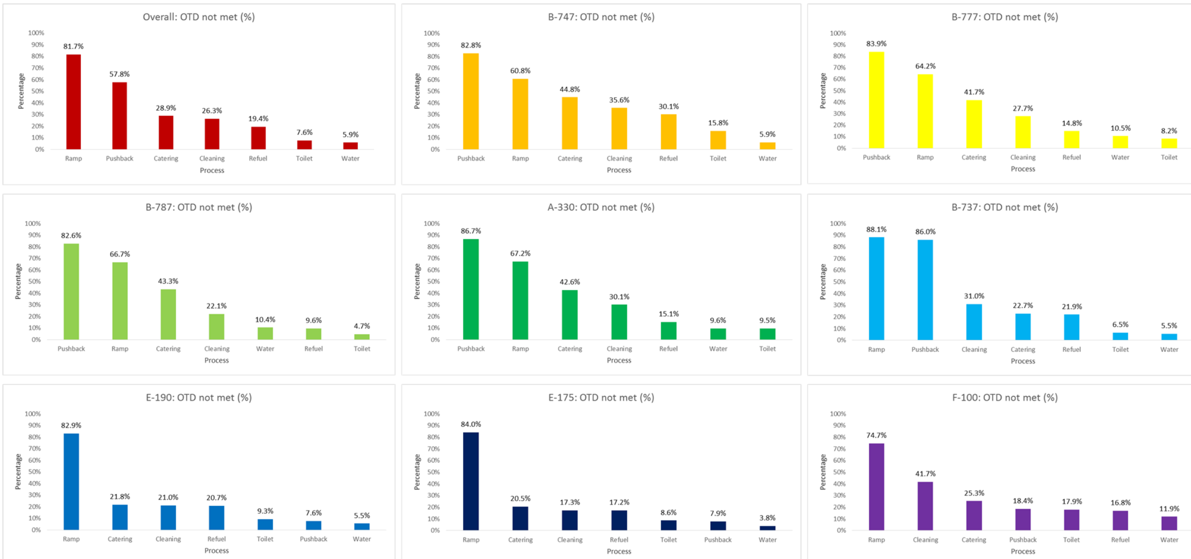

Figure 4.5 Overview of the OTD performance sorted by aircraft type.

have the most freedom when planning their task and are only dependent on each other because they can not operate at the same time. This can be seen in the low percentages for not achieving the OTD.

A closer look at the OTD

There can be multiple reasons the OTD is not achieved like a shortage of people, broken equipment or delays of other processes. Not achieving the OTD does not necessarily mean a bad performance of the process. The process can start too late for example, automatically resulting in a later delivery. Therefore a closer look at the OTD is needed by looking how often the OTP is achieved while the OTD is not achieved. If the OTP is achieved, but the OTD is not achieved, this is because the process started too late. The cases where this happens are the focus areas for improvement in this research. A better control should prevent processes from starting too late, having the biggest impact on the overall performance of the system. Figure 4.6 shows the percentages where the OTP is achieved when the OTD is not achieved, and compares these for each type of aircraft.

For ramp services it can be seen that almost all the time the OTD is not achieved, the OTP is not achieved as well. This means that most times the OTD is not achieved due to a bad performance of the process, and probably not by bad control. This can be the case when personnel is not working fast enough, but also wrong assumptions could be made when calculating the duration of tasks

Figure 4.6 Percentage of OTP achieved when the OTD is not achieved, sorted by aircraft type.

in CHIP. To improve the OTD for the ramp services, a better estimation of the process length will therefore probably have the highest impact on the performance.

Overall, water, toilet, catering and refuel tasks all have a percentage of more than 50%, meaning a better control of these processes could result in much better performances. The full results in table format for each process can be found in appendix B.

Summary of the OTS, OTP & OTD

Table 4.7 Summary of the OTS, OTP & OTD Performance. The percentages how often the KPI is not achieved, and for each KPI the best and worst performing process.

	Average	*Best*	*Worst*
OTS	79,4%	Refuel (76,4%)	Toilet (98,6%)
OTP	64,4%	Water (34,4%)	Ramp (84,0%)
OTD	46,1%	Water (5,9%)	Ramp (81,7%)
OTD while OTP is achieved	20,5%	Ramp (5,8%)	Water (73,5%)

4.3.3 Efficiency

For the analysis of the efficiency of the TA processes, the time a resource is used is compared to the time the resources is available. Using the formula as described in section 4.2.3, the time a resource is used is defined as the time a resource is at the location performing the task it should perform. Travel times or lunch breaks are excluded from this time. The results for July and August 2018 can be seen in table 4.8.

Table 4.8 Worker efficiency for each process for July and August 2018.

Process	*Number of workers*	*Average number of tasks per worker*	*Average duration worker shift*	*Average time effectively worked*	*Percentage of the time working effectively*
Total	34822	6.0	456	164	35.5%
RampK1	2034	8.5	580	307	52.9%
RampK5	8784	4.0	479	193	40.3%
Cleaning (Asito)	2554	4.3	421	169	40.1%
RampK4	5708	4.5	415	160	38.5%
Water	834	22.7	451	157	34.7%
RampK2	4559	4.0	478	148	30.9%
Refuel (Hydrant)	2730	4.7	455	137	30.2%
Refuel (Tank)	1211	7.2	445	128	28.8%
Toilet	781	21.5	459	123	26.8%
Pushback	2505	7.1	456	118	25.9%
Cleaning (Kluh)	3043	8.3	389	97	25.0%
Catering	79	10.1	NA	NA	NA

Looking at the total of all workers, an average worker shift takes around 456 minutes or 7,5 hours. The longest shifts are from Ramp K1, which on average take 580 minutes or 9,5 hours. Klüh cleaners have on average the shortest working shifts compared to the other processes with only 389 minutes, or 6,5 hours. These times include a lunch break of 30 minutes and are realistic values when comparing them with normal working shifts.

For each process some KPIs are listed to assess the data set. Overall an efficiency of 35.5% can be seen for each worker, which is not strange if you look at the distances workers sometimes have to travel to get to their destination and the time needed for their lunch or coffee breaks. Also, the 7-wave network requires more workers during a day due to the peaks in the flight schedule. Since it is not possible to let people work only during these peaks, the workers have to do less tasks during the off peak hours.

In general, the ramp workers score high when looking at the time they are working efficiently, with workers at K1 having the highest percentage with almost 53% of the time used effectively. Ramp workers do not need to arrange their own equipment, and top drivers place most of the necessary equipment at the right location. Since ramp workers have their own area they work in, traveling distances are also much shorter compared to processes which have to travel all over Schiphol. Therefore ramp workers score high on efficiency but still, higher percentages should be possible.

Cleaning (Klüh), pushback and toilet have the lowest percentages, and workers are not used very effectively with percentages in the mid twenties. Toilet is expected to be quite low because tasks are relatively short, and more driving is required between the tasks. Also, when a toilet truck is full, it has to drive quite far to empty the truck resulting in a lot of time wasted on transportation. The same can be said for cleaning (Klüh), where cleaning trucks have to travel back to their home location when they have cleaned the aircraft, resulting in a lot of time wasted on transportation as well. Pushback on the other hand has very short tasks for which it has to travel around Schiphol, resulting in a lot of time that is used for traveling to the right location.

KCS works with truck IDs which are unique for that truck but do not change every day. Different workers drive in the same truck each day, hence it is difficult to say something about the efficiency of the workers. Currently only this data is available, but for future research a more in-depth analysis should be done on the KCS data.

Worker shifts

Workers in the real system as well as in the model work in shifts. Figure 4.7 shows a Gantt chart for all workers that handled an aircraft with a visitID on the 1^{st} of August 2018. Worker shifts vary from minute to minute since workers have to clock in and out when starting and ending their shift. The real times (instead of scheduled) are taken because they give the most realistic values of how long and when workers worked during the day that is being simulated. This data is then used to determine the starting times for each working shift that can be used by the workers in the simulation. The shifts can be divided into four categories:

1. **Morning shifts:** contain shifts 1 till 5, each starting 1 hour apart.
2. **Afternoon shifts:** contain shifts 6 till 8, each starting 1 hour apart.
3. **Evening shift:** contains shift 9.
4. **Night shift:** contains shift 10. Not all process work at night, hence the smaller amount of workers during these times.

Each shift takes the average of 7,5 hour as determined in the previous section and will have three different versions with an early, normal or late 30 minutes lunch break. After the tenth shift the first shift starts again for the next day. For each individual process this analysis is done to see how many workers fall in each category.

Note that the morning shift for the 2^{nd} of August is not as busy as for the 1^{st} of August due to the fact that only the visitIDs of the 1^{st} of August are taken. All workers working on the 2^{nd} of August are from aircraft which arrived on the day before.

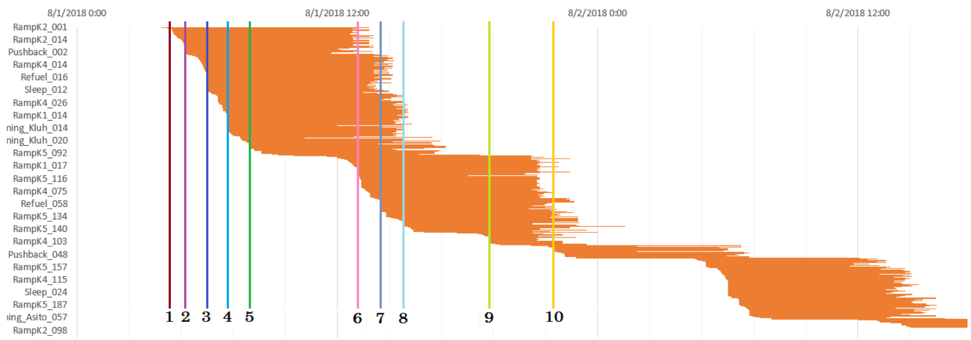

Figure 4.7 Gantt chart for all workers on the 1^{st} of August. The vertical lines represent the starting time of the worker shifts used for the model.

4.4 Conclusion on Data Analysis and KPIs

After this chapter the answer to the third research question can be given:

- *3. How is the process currently monitored: data availability, KPIs?*

There is a lot of data available, but often this data contains a lot of measurement errors and is incomplete. The result is that data from different data sources has to be merged and filtered to get data that can be used for the data analysis and as input for the simulation model.

Table 4.7 shows that the OTS, OTP and OTD are not achieved most of the time. Especially the high percentage for the OTD that is not achieved shows us that many processes finish their task too late, directly influencing the TA process. It can also be concluded that task duration as calculated by CHIP do not correspond well with task duration of the real system and take longer in 64,4% of the time. It is therefore important to first update the input data as used by CHIP to get data input that resembles the real system before different control scenarios should be applied.

Given the analysis on the data and KPIs, the following KPIs have been identified for tracking and monitoring performance of the turnaround as well as each individual apron process:

1. Effectiveness:
    - OTS_{ratio}
    - OTD_{ratio}
    - OTP_{ratio}
2. Efficiency:
    - $\eta_{resource}$
3. Costs
4. Simulation KPIs
    - Time in system
    - Time processing at gate
    - Time idle at gate
    - Traveling time worker
    - Distance traveled worker

CHAPTER

5

SIMULATION MODEL

This chapter will describe how the model which will be used to assess alternative control scenarios is developed, and which assumptions are made. Simio software is used to build the simulation model. A description on the different components in the model is given and how they interact with each other. The model will be validated and verified at the end of this chapter. This chapter answers the following sub research question:

> 4. *How can alternative control scenarios be assessed using simulation?*

5.1 Overview of the Model

The object-oriented simulation tool Simio is used to make a computerized simulation model. The simulation model is used to assess different scenarios on effectiveness and efficiency. The KPIs defined in section 4.2 are used to analyze the output results of the model. The simulation model is able to measure and track lead times in the subprocesses on process level. In figure 5.1 a high level overview of the architecture of the model is provided. A brief description of object functions, data input, key output and assumptions are given afterwards.

Figure 5.1 High level overview of the architecture of the model.

Push-Pull Strategy

The system analysis showed the current hybrid push-pull strategy is already used by each individual process. The current push-pull strategy will be implemented on a global level, allowing a more integrated control for the apron. A global queue is used where each flight entity arrives and requests the resources they need at the time needed. The flight entity can be seen as the parent object with all the corresponding TA processes linked to this parent object using their VisitID. In case an entity delays, all corresponding processes will automatically delay their request for resources with the same amount of time. When a worker is available, it scans this global queue to see if there are unfinished tasks available which can be performed by that particular worker. The worker then chooses which task it is going to perform based on a dispatching rule and reserve a time slot for this task. Because all processes are linked to the corresponding entity, a more integral control of the apron is accomplished where each process has a link to the other processes via the entity.

5.1.1 Overview

Schiphol airport consists of many different components which have to be modeled in Simio. Each of this components is thoroughly tested during the building of the model and will be validated at the end of this chapter. The main components are represented in the model using different objects:

- **Gates:** are represented using servers. Each gate uses the flow chart as described in chapter 3 to process the entities entering the server. The gates are placed using their geographical location from satellite images from Google Maps.
- **Roads:** are modeled using paths on which the different apron processes can travel from gate to gate. KLM uses distances and average driving speeds to calculate the driving times, so the same method is used for the model. Using average speeds from the KLM norms and the shortest path between the nodes, the driving times can be calculated.
- **Flights:** are modeled using entities, which are part of an object model and can have their own intelligent behavior. The arriving flights are created by the source, which is an object that allows the creation of entities by a specified arrival pattern. The flight entities are then processed by a server, before they depart and are destroyed by the sink.
- **Apron processes:** are modeled using workers, which are used as a movable resource that is seized and released for tasks by model process logic. For each process a type of worker is used which has a traveling speed based on the norms determined by KLM. Work schedules are then given to each worker to represent the different shifts. The processes at the different servers can only be performed when the right workers are at the right server.

A more detailed background on each object is given in sections 5.1.3 to 5.1.5.

Data Input

The analyzed data from the previous chapter is used to create input that resembles the actual process as good as possible. Different scenarios are simulated using input data of a normal busy day of operations in the summer flight schedule of 2018. The selected case day has to meet two requirements:

- No major technical disturbances for the different apron
- No major disturbances in arrival punctuality due to weather conditions or other external factors processes.

The 1^{st} of august 2018 has been selected as case day since its meets the required criteria. All flights from KLM and Transavia that arrive or depart on this day are used which are 476 flights in total. This includes flights that arrive the days before and depart on the 1^{st} of August. Flights to the hangar which do not depart again on the 1^{st} of August leave the system through the sink, and flights returning from the hangar departing on the 1^{st} of August are placed on one of the available buffer spots. This is done because the hangars are not part of the model.

The filters as described in section 4.3.2 are used to filter the merged data sources as seen in figure 4.1. The real duration for each task is used to determine the task lengths in the model for each specific aircraft. As described earlier in section 4.3.2, departure oriented processes tend to have longer task times than necessary because tasks start earlier but do not work full time during the whole length of the task. Especially boarding and loading tasks start much earlier than necessary, sometimes resulting in extremely long task times of multiple hours and overlap with other processes. An example of this can be found in appendix B figure G.1. The following extra adjustments to the processing times are necessary to use this data as input for the model:

- Real values of CHIP are chosen
- Multiple CHIP tasks with the same visitID will be grouped if they are performed by the same team or person. Depending if the processes are done in parallel or series the highest value for the duration is chosen or these are added together.
- Turnaround tasks are split in an unload and load task.
- All unloading and loading tasks are deducted by two minutes to compensate for the blocks and pylon steps at the beginning of the flowchart.
- Average values for boarding and deboarding are taken.
- Pushback tasks are reduced because they often overlap with other tasks while they are waiting to perform the pushback. Therefore only the time spent on the real pushback is taken.
- NULL values, outliers or negative process times will be replaced by average process times for each aircraft category which can be seen in table 5.2.

Table 5.2 shows the average values as found in chapter 4.3. The values that are adjusted are highlighted in bold and used as input for the model. For short turnarounds of commuters and the B-737, some extra adjustments are made as seen in table 5.1.

Table 5.1 Extra adjustments for short turnaround times.

Aircraft	*Turnaround time*	*Deboarding*	*Boarding*	*Unloading*	*Loading*
F-100	<50 minutes	9	17	17	19
E-175	<50 minutes	9	18	18	20
E-190	<50 minutes	10	20	21	22
B-737	<60 minutes	12	24	21	21
B-737	<70 minutes	14	29	25	26

Table 5.2 Adjusted process times used for the input of the simulation model. In bolt are the values which are corrected.

Process	*Orientation*	*A330*	*B737*	*B747*	*B74C*	*B777*	*B787*	*E175*	*E190*	*F100*
Blocks	ARR	1	1	1	1	1	1	1	1	1
Boarding	ARR	***30***	***25***	***35***	***35***	***35***	***30***	***18***	***20***	***17***
CabinCheck	ARR	9	7	9	9	9	9	5	5	1
CabinSecurityCheck	ARR	14	11	14	14	14	14	11	11	2
Catering	ARR	25	10	34	34	28	32	9	11	6
Cleaning	ARR	36	12	48	48	46	45	10	10	9
ConnectingStairs	ARR	1	1	1	1	1	1	1	1	1
Deboarding	DEP	***15***	***13***	***18***	***18***	***18***	***15***	***9***	***10***	***9***
DisconnectBridge	DEP	2	2	2	2	2	2	1	1	1
DisconnectStairs	DEP	1	1	1	1	1	1	1	1	1
Load (Ramp)	DEP	***33***	***33***	***43***	***43***	***41***	***37***	***20***	***22***	***19***
PassengerBridge	ARR	2	2	2	2	2	2	1	1	1
PositioningCatering	ARR	1	1	1	1	1	1	1	1	1
Pushback	DEP	***5***	***4***	***5***	***5***	***5***	***4***	***3***	***3***	***3***
PylonsGPU	ARR	1	1	1	1	1	1	1	1	1
Refueling	ARR	46	20	56	56	49	49	16	19	15
RemoveBlocks	DEP	1	1	1	1	1	1	1	1	1
TechDepSerPylonsGPU	DEP	5	4	3	3	3	3	2	2	2
TechnicalHandling	ARR	35	25	35	35	35	30	15	15	15
Toilet	ARR	13	4	14	14	10	10	5	5	11
Towing	DEP	25	18	28	28	26	24	8	9	12
Unloading (Ramp)	ARR	***30***	***30***	***39***	***39***	***37***	***34***	***18***	***20***	***17***
Water	ARR	27	***10***	26	31	17	***10***	***10***	***10***	***10***

With the adjustments to the data the following files are used as input for the model:

1. **tbl_Gates -** contains data about the coordinates and information of all gate objects used for the model. Using the AutoCreate function of Simio, all servers (obj_Gate) are created and placed at the right location. This way not all objects have to be created manually, and it's easy to adjust settings for multiple objects at the same time.
2. **tbl_Nodes -** does the same as tbl_gates, but for the TransferNodes in the model that are used to route the workers. For the tbl_Gates and tbl_Nodes files the data sources are:
    (a) Planning norms
    (b) DistancesGates
    (c) Satellite images Google MAPS
3. **tbl_FlightData -** contains all the necessary flight data used to run the model. The flight data is used to create the flight entities and contains information such as arrival and departure times, gate location, aircraft type and the orientation of the aircraft for example.
4. **tbl_Processes -** contains all data about the processes and their length.
5. **tbl_Routing -** is used to determine the routing of each aircraft. The routing table also determines the type of tasks that have to be performed for each aircraft at each server.
6. **tbl_Activities -** lists the choices for activities that can be performed at the servers such as all turnaround processes, or only the arrival or departure oriented processes.
7. **tbl_ActivityTasks -** contains info about the tasks, in which order they have to be performed, and which secondary resources are required at each server and for each flight. For the tbl_FlightData, tbl_Processes, tbl_Routing, tbl_Activities and tbl_Activitytasks the data sources are:
    (a) CHIP
    (b) FIRDA
    (c) Apron App
8. **tbl_WorkSchedules -** is used to determine the amount of workers and the time they are available during the simulation run. Data sources:
    (a) CHIP
    (b) Rosters from Tactical Planning

9. **Manual input -** from the modeler is required using the controls in the experiments. Required input is the call time for each type of worker, the dispatching rule and the tie breaker rule. These options are added to the model by adding properties to the model which then can be adjusted in the experimental runs for the different scenarios that will be assessed.

All input files are attached with the model.

Key Output

Output of the model are handled flight entities and all corresponding data of the entities itself, the used resources, waiting times and utilization. This data will be discussed in chapter 6.

Assumptions

The simulation model represents the key characteristics, behaviors and functions of Schiphol Airport. Some assumptions are made for correct simulation of the overall process. From the process analysis it became apparent that many variables and uncertainties affect the process. To approximate the actual process, these variables and uncertainties are implemented in the model. Not all uncertainties can be modeled due to data availability and are assumed to be fixed values. Uncertainties and key assumptions on variables and their way of modeling:

1. The **apron processes -** (described in chapter 3) are the main focus of this research. Workers that are needed as a secondary resource for other processes such as security checks, (dis)connecting passenger bridge and (de)boarding for example are considered to be always available and to have unlimited capacity.
2. A **Push-Pull strategy** is implemented, where all processes which are not departure oriented will be pulled at the earliest possible start based on the arrival time of the aircraft. The departure oriented processes will be requested as late as possible, based on the departure time.
3. **Distances -** between the gates are modeled using the geographical coordinates of the gates provided by satellite images of Schiphol airport in combination with the DistanceGates file from tactical planning. KLM currently only uses distances between "zones", and not between every individual gate. The method used in this research is more precise and therefore chosen to determine the distances between gates.
4. **Roads -** are simulated using paths and do not have capacity constraints. This means there will never be traffic jams of any kind or other obstructions that can cause increased driving times.

5. **Human errors -** like arriving too late, sickness, or mistakes are not taken into account.
6. **Disturbances -** like mechanical failures, bad weather, and delayed flights will not be taken into account.
7. **Equipment availability -** is considered to be unlimited and always available for ramp services.
8. **Number of runs -** the model is stochastic. Therefore, a minimal number or runs needs to be performed to reduce the chance of making a recommendation based on results of the model. In appendix D, an explanation of the calculation of number of runs is given.

5.1.2 Arriving and Departing Flights

A source and sink are used to create and destroy flight entities entering and leaving the system. The arrival mode of the source is set to an arrival table which uses the arrival time in table tbl_FlightData to create entities at that particular time. The sink is used to destroy the created entities when they are departing the real system.

Data Input

- *tbl_FlightData* - is used to create the entities at the right time and as arrival table for the source.

Key Output

Output of the source and sink gives information about how many entities are in the system, and the time they have been in the system.

Assumptions

- All flights arrive 30 minutes earlier than in the real system. This time is used to call the workers to the right location before the flight arrives there. A delay step of 30 minutes minus the call time used for that particular ramp worker is used to compensate for the early arrival. At the gate, another delay step is used for the remainder of the 30 minutes (the call time) so the flight entities can not be processed before the real arrival time. This will be further explained in section 5.1.3.

5.1.3 Gates

Servers are used to model the gates. The gate handling procedures as described in the analysis (see figure 3.3) are implemented into the server logic. This is done by using the task sequence option and task sequence number scheme from Simio, which is described in the reference guide as follows; The task sequence numbering scheme supported by the Task Sequence element has the following format [Simio, 2017]:

XX.YY.YY.YY...

XX is an integer referred to as the root sequence number. This number is the primary key that determines the serial precedence requirements of a task. YY are optional integer suffixes separated by dots (periods) that can be appended to the root sequence number in order to define parallel subsequences in the task workflow.

The precedence rules used to determine task order are as follows:

- If the task sequence number consists of only a root number (only XX), then all tasks with lower root numbers (ignoring any suffixes) must precede that task. Note: It is possible to have one or more tasks with the same root sequence number, with or without suffixes.
- If the task sequence number consists of a root number and one or more suffixes, then all tasks with lower root numbers must precede that task, as long as those tasks either have no suffixes or all shared suffixes match.
- If the task sequence number consists of a root number and one or more suffixes, then all tasks whose sequence number is the same root number without any suffixes must precede that task.

This is graphically represented in figure 5.2. Different processes have to be performed at different locations, depending on if the aircraft has a normal turnaround or has to be towed to another gate or buffer. Tows to the hangar which do not depart on the simulated day are modeled by entities leaving the model after the arrival oriented processes are performed. Arrival oriented processes have a higher priority compared to departure oriented processes because passengers want to leave the airport as quick as possible with their baggage. All unnecessary processes will therefore be moved and performed at the buffer or other gate if possible. Except for the refuel task which is performed at the departing gate, because the buffers are not connected to the underground fueling system of Schiphol. All departing oriented processes of the flight are performed at the departing gate and are

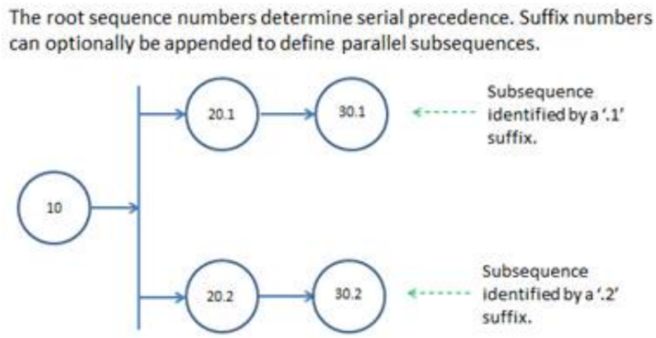

Figure 5.2 The task sequence number scheme of Simio [Simio, 2017].

scheduled based on the departure time. An overview of which processes are scheduled at which location and in which task sequence is shown in table 5.3.

When an entity arrives at the gate all processes which are arrival oriented are fired and workers are reserved at the time they are necessary. The entity stays in the input buffer of the server for the amount of time which is predetermined by the user of the model using the seize time ramp option in the controls of the model or experiments. Because of this waiting time at the gate, entities are sent earlier to their gate by the same amount of time to compensate for this waiting time. This way workers can schedule their tasks before the flight entity arrives at the gate and workers can start moving on time to the right gate. Delay steps are used in case the starting time of the task is further ahead in time than the time that is chosen to send the workers to their tasks. Each apron process is

Table 5.3 Task activities for a turnaround, buffer tow or change gate tow.

			Turnaround	**Buffer Tow**			**Gate Change**	
Task Name	*Orientation*	*Task Sequence Number*	*Turnaround Gate*	*Arrival Gate*	*Buffer Gate*	*Departure Gate*	*Arrival Gate*	*Departure Gate*
Blocks	ARR	10.1	*x*	*x*	*x*	*x*	*x*	*x*
PassengerBridge	ARR	20.1.2	x	x		x	x	x
Deboarding	ARR	30.1.2.1	x	x			x	
Catering	ARR	30.1.2.2	x		x			x
CabinSecurityCheck	ARR	30.1.2.3	x		x			x
Cleaning	ARR	30.1.2.4	x		x			x
CabinCheck	ARR	40.1.2	x			x		x
Boarding	DEP	50.1.2.1	x			x		x
DisconnectBridge	DEP	60.1.2.1	x	x		x	x	x
PylonsGPU	ARR	20.1.1	x	x	x	x	x	x
PositioningCatering	ARR	30.1.2.5	x		x			x
ConnectingStairs	ARR	30.1.2.6	x		x	x		x
Refueling	ARR	30.1.1.3	x			x		x
Water	ARR	10.2	x		x			x
Toilet	ARR	10.3	x		x			x
TechnicalHandling	ARR	30.1.1.1	x		x			x
Unloading	ARR	30.1.1.2	x	x			x	
DisconnectStairs	DEP	50.1.2.2	x		x	x		x
TechDepService	DEP	40.1.1.2.1	x		x	x		x
Load	DEP	40.1.1.2.2	x			x		x
RemoveBlocks	DEP	40.1.1.2.3	x	x	x	x	x	x
Pushback / Tow	DEP	70	x	x	x	x	x	x

represented by a worker which is used as a secondary resource by the task. The process can start its task only when the right worker is at the server, and when all its predecessors finished their task.

Each TA processes is modeled by a task, and each task has three add-on processes:

1. **Task Ready:**
   Occurs when all of the task's predecessor dependencies have been satisfied and the entity is about to try seizing the resources needed to perform the task. Two delay steps are used here, the first one is the standard delay due to the processing time of its predecessors. The second delay is a variable delay which depends on the time workers of predecessors arrive too late, and an extra delay is necessary for the worker of the corresponding task.
2. **Starting Task:**
   Occurs when all the task's resource requirements have been satisfied and the processing of the task is about to begin. Decisions are used to determine whether all previous processes are finished, and if this is not the case a wait step is used which waits until a fire event of the previous process is fired.
3. **Finished Task:**
   Occurs when the processing of the task has been finished. A fire step is used to fire an event and an assign step is used to change the entity state for that process to show the task has been finished.

Data Input

1. *tbl_gates* gives each gate an associated node to which the secondary resources have to go to perform their tasks and its coordinates in the model. It also contains all information about which resources can be used at the specific location such as the corresponding ramp department or if a hydrant truck of tank truck can be used.
2. *tbl_Activity_Tasks* - is used to determine the processes that have to be performed at the gate. The gate uses this table to determine all gate specific tasks and makes references to lists and properties that contain information about each specific gate.
3. *Flight entities* - are used as data input. The entities contain all data that is necessary for the TA processes to be performed at the gate.

Key Output

Key output are the handled flight entities and their corresponding data about all processes. This data will be used to assess the different control scenarios in chapter 6.

Assumptions

1. *Choosing workers* - if multiple workers are available as a secondary resource, the server chooses the worker which is closest to the server.
2. *Blockages and distances* - are not taken into account on the platform at the gates.

5.1.4 Flights

Flights are modeled by using entities. These entities are created at the source simulating arriving aircraft at Schiphol before they move to a TransferNode. The entities are created using the tbl_-FlightData, which contains information about the aircraft, its orientation, priority and arrival and departure time. The departure time that is used it the actual arrival time of the flight + the available ground time. If a flight departs later than this time, it is delayed. Each flight has a VisitID which is used as a column key to refer to the right data in the other tables. tbl_Routing is used for the routing of the entities and gives information about what processes have to be performed at which location. tbl_Processes contains all information about the processes for each flight and how long they take. If processes have a time of zero they are automatically skipped at the server.

Data Input

1. *tbl_FlightData* - is used to create the entities and contains information about the flights that are created.
2. *tbl_Routing* - contains the routing of the entities and information about which tasks have to be performed at which gate.
3. *tbl_Processes* - contains all information of each process that has to be performed for each flight entity and the process time.
4. Table 5.4 is based on the document from Eurocontrol and shows an overview of the costs that are given to a delay in this research. For delays of less than 15 minutes, the extra costs for fuel are considered negligible as calculated in appendix E.
5. The dynamic flight priorities are implemented setting the initial priorities to the new dynamic priorities. In the process logic the departure time is compared with the simulation time at the end of each task. If the simulation time exceeds the 15 minutes barrier, the flight priorities will switch back to the standard priority.

Table 5.4 Delay costs (€). For star flights there is an additional penalty of € 100000 and for ICA flights the costs for a delays with the new priorities are set to zero [of Westminster, 2015].

Delay (Minutes)	*5-15*	*>15-30*	*30-60*	*>60*
A-330	790 (0)	2850	10360	70000
B-737	275	980	3575	25000
B-747	790 (0)	2850	10360	70000
B-777	790 (0)	2850	10360	70000
B-787	790 (0)	2850	10360	70000
E-175	180	660	2390	15000
E-190	180	660	2390	15000
F-100	180	660	2390	15000

Key Output

The flight entities do not have output that is used by other objects in the simulation model. Data from the entities such as idle times, constraining resources, and all other relevant data is logged by the model and can be analyzed in more detail after the simulation run.
This will be done in chapter 6.

Assumptions

- The flight entities arrive earlier at the gate, depending on the call time for the workers, to call all necessary resources to perform their tasks. It waits at the input buffer until its real arrival time.
- The flight entities do not block each other.
- The process times are pre-defined for each process upon creation of the flight entities and multiplied by a normal distribution with a mean of 1 and SD of 0,05.
- Because all flight entities have to travel from the source to the server, and from the server to the sink and therefore have a speed of 1.000.000 m/s to make the traveling time from source to gate and from gate to the sink almost zero.
- The flight entities use TransferNodes to determine their routing and move the entities to the right gates.

5.1.5 Apron Processes

The apron processes are modeled using workers which are used as secondary resource by the flight entities when being processed at a server. Processes within the server can only be processed when the right worker is at the associated node of the server.

Each worker has a home node which represents the worker depot and is used as starting point when the simulation starts. Each worker works at the times which are determined by the working schedule and can only move via the network of paths to their destination. A standard dispatching rule is used to dynamically select the next task for each worker. When a worker becomes available to perform a task, it looks at the global queue with tasks as described in section 5.1 to see if it can schedule a next task to perform. There are multiple methods for a worker to choose the next task based on a dispatching rule e.g.: first in queue, priority, departure time, least slack time or distance. A secondary criterion is used to break ties if they occur. In chapter 6 different methods will be chosen to see which method works best in different scenarios. If no tasks are waiting in the queue, the worker travels back to its home node and waits for a new task entering the queue.

Workers work according to a work schedule which is based on the working shifts as found in the HubDB and they have one lunch break of 30 minutes during their shift. The amount of workers for the ramp, catering and cleaning services is adjusted for the fact that workers represent a team in the model and not individuals.

Data Input

1. *Traveling velocity* - is given to each worker. An overview of the velocities for each worker can be seen in table 5.5.

2. *tbl_Workschedules* - are used to determine the amount of workers which need to be created as well as the times they work. The shifts are used as determined in section 4.3.

Table 5.5 Worker velocities as determined by KLM's Tactical Planning department.

Worker	Velocity (km/h)
Ramp	14
Toilet	15
Water	15
Catering	15
Cleaning	15
Pushback	15
Refuel	18

Key Output

All information about the worker such as performed tasks, idle time, traveling distance and the times it is constrained can be found in the results. The results can be seen using the pivot table or graphically illustrated using Gantt charts. The Gantt chart displays each resource (worker) along with time bars that show each entity (e.g. a flight) that utilizes the resource. The changing resource state is shown on each Gantt row behind the time bars. Each row can be expanded to get a more detailed look on each worker, the tasks it performs, or why the worker is constrained. An example of how this looks can be found in appendix C in figures C.1, C.2 and C.3. The responses used in the experiments show the most important KPIs that are used to assess a scenario to quickly see and compare different scenarios.

Assumptions

Chapter 3 showed big differences between the different processes, the way they operate and the team sizes. For aircraft services all teams consist of a single operator with its own terminal. For ramp services different team sizes can be found which have a big influence on the amount of workers that will be used in the simulation model. In the simulation model workers represent a team, wherein the real system each worker with a HHT has a unique workerID consisting of one or more workers. Multiple tasks are created in CHIP for each workerID if multiple workers are used to handle the same aircraft. Therefore the workerIDs should be corrected depending on the team size and the amount of HHT used by a team. Table 5.6 shows the compensation for all workers. All other assumptions which are made for each type of worker are listed on the next page:

Table 5.6 Correction that is used to compensate the amount of workers to the team sizes.

Worker	*Multiplier*	*Reason*
Refueling	*1	Team size is 1
Catering	*1	Team size is 1
Pushback	*1	Team size is 1
Cleaning	*(2/3)	50% of the tasks is performed by two teams
Water	*1	Team size is 1
Toilet	*1	Team size is 1
RampK1	*1	Team size is 1
RampK2	*(1/3)	Team size is 2 and 50% of the time two teams are necessary for one task
RampK4	*(1/3)	Team size is 2 and 50% of the time two teams are necessary for one task
RampK5	*(1/5)	Team size is 5

- *All Workers:*

  1. Start immediately after they arrive at location.
  2. Travel with an average speed that can be seen in table 5.5.
  3. Are never sick or have to deal with broken equipment.
  4. Have the same level of experience and skills.
  5. Go back to their home node if no tasks are available.
  6. Tank, catering, toilet and water workers have a maximum number of tasks before they have to refill, restock or empty their load at their home node. Back at the home node the amount of tasks a worker has performed is reset to zero. This takes five minutes for each task until a value of zero is reached. The maximum number of tasks is given under the assumptions for each type of worker.

- *Ramp workers:*

  1. Only TMs are taken into account when compensating for team sizes because the TCs have their own separate task.
  2. Are modeled using one worker which represents the whole team. There is no difference between TCs and TMs, and they perform their tasks as a team.
  3. Are assigned to tasks in their own area (K1-K5), and only switch areas if a task cannot use workers from their own area. This is done according to a preference list based on the swim lane analysis in chapter 3.

- *Toilet and Water workers:*

  1. Cannot interchange between each other.
  2. Do not take jet start tasks into account.
  3. Have a maximum number of five tasks which they can handle before they have to return to their home node.

- *Catering workers:*

  1. Do not take special requests into account and therefore the order is not fixed.
  2. Have a maximum number of three tasks before they have to restock at their home node.

- *Cleaning workers:*

  1. Are divided in two groups: (1) ICA flights (Asito), (2) EUR flights (Klüh), which cannot interchange between each other.
  2. Are modeled by a single worker which represents the whole cleaning team.

- *Pushback workers:*

  1. Can pushback all types of aircraft.
  2. Only do pushback tasks. Towing tasks do not need a secondary pushback worker since towing is out of this scope.

- *Refuel workers:*

  1. Are divided into two groups: hydrants and bowsers.
  2. Depending on the gate a preference list is used where a hydrant truck is always chosen over a tank truck if possible.
  3. If a gate does not have a hydrant point, refuel tasks can only be performed by bowsers.
  4. Pre-fuel tasks are excluded and added to the fin-fuel task as one long fuel task.
  5. Hydrant trucks have unlimited capacity while bowsers can refill a maximum of five aircraft before having to refill at their home node.

5.1.6 Roads

Roads are modeled as paths with the same distance as they have in the real system. TransferNodes are used to connect paths together to create the network that can be used by the workers. Each road is modeled using two unidirectional paths with unlimited capacity and no speed constraints to prevent deadlocking. Workers can pass each other on paths if necessary. The TransferNodes are placed using their geographical coordinates, and an add-in is used to automatically create all paths between the TransferNodes.

Data Input

1. *tbl_Nodes* - are used to auto create all the TransferNodes in the model using geographical coordinates the same way the gates are placed.

Key Output

The paths are mostly used to route the workers to their location and constrain them in their movement just like the roads do in the real system. Data about the amount of workers that traveled specific roads can be obtained after running the simulation, if requested.

Assumptions

- All roads are modeled as unidirectional paths, using two if the road is bidirectional in the real system.
- The capacity is assumed to be infinite, resulting in no traffic jams or waiting at intersections.
- No waiting for passing aircraft is taken into account at the intersection with the airstrip at the D and E buffer.
- To turnaround, the workers have to exit and re-enter the network at a TransferNode.

5.2 Verification

The purpose of model verification is to assure that the simulation model reflects the operational process [Jerry, 2005]. A close and thorough examination of the different objects and their outputs is needed to assure the simulation model is accurate and credible [Jerry, 2005]. During the development of the model the objects as described in section 5.1.1 are individually verified with expected and realistic output. This way the results of each object is tracked and verified against expected results resulting in a reliable and stable simulation model. Results of these experiments can be found in appendix C.

5.3 Validation

Naylor and Finger formulated a three-step approach to model validation that has been widely followed [Jerry, 2005]:

1. Build a model that has high face validity.
2. Validate model assumptions.
3. Compare the model input-output transformations to corresponding input-output transformations for the real system [Naylor and Finger, 1967].

Complete validation results and discussion are presented in appendix C.

CHAPTER

6

EXPERIMENTAL RESULTS

In this chapter the model performance for different control scenarios will be assessed using the KPIs as determined in chapter 4. The first three experiments will be used as input for the final experiment using the OptQuest optimizer. This experiment will compare all results and will give a closer look at the values for the KPIs. The fifth sub research question will be answered in this chapter:

5. How do the redesigned control scenarios improve the performance of KLM's apron processes?

6.1 Experimental Plan

Tables 6.1, 6.2 and 6.3 show an overview of the experiments that will be performed to assess different control scenarios for the workers. To assess the different scenarios input data of the 1^{st} of August is used as described in section 5.1.1. This was a normal busy day of operations in the summer flight schedule of 2018.

The experiments will be assessed using the KPIs as described in chapter 4:

1. Costs
2. Effectiveness
3. Efficiency
4. Simulation KPIs

Table 6.1 Experiment 1: Call Time workers get sent to their task

Worker Name	**Call Time [min]**	**Increment Value [min]**
Catering	3-24	1
Cleaning	3-24	1
Pushback	3-24	1
RampK1	3-24	1
RampK2	3-24	1
RampK4	3-24	1
RampK5	3-24	1
Refuel	3-24	1
Toilet	3-24	1
Water	3-24	1

Table 6.2 Experiment 2: Control scenarios used by workers to determine their next task.

Name Control Scenario	**Dispatching Rule**	**Tie Breaker Rule**
PRIO-FIFO	LargestPriorityValue	FirstInQueue
PRIO-DD	LargestPriorityValue	EarliestDueDate
PRIO-CR	LargestPriorityValue	CriticalRatio
PRIO-LEASTSLACK	LargestPriorityValue	LeastSlackTime
FIFO-DD	FirstInQueue	EarliestDueDate
DD-FIFO	EarliestDueDate	FirstInQueue
CR-FIFO	CriticalRatio	FirstInQueue
LEASTSLACK-FIFO	LeastSlackTime	FirstInQueue
WAITINGTIME-DD	LongestTimeWaiting	EaliestDueDate

Table 6.3 Experiment 3: Dynamic flight priorities. Smaller priority value means a higher priority

	Standard Situation	First 15 minutes
STAR	1	1
ICA	5	10
EUR	10	5

Experiment Properties

The following properties are used for the experiments:

- Warm-up Period - is set to 24 hours. This is done to include the flights of the previous day that depart on the simulation day.
- Default Replications - the number of replications to be run by each scenario is set to 10 as determined in appendix D.
- Confidence Level - is used for statistics and is set to 95%.
- Upper and lower percentile - are set to 75% and 25% and represent the upper and lower bound of the SMORE plots.
- Primary Response - is the Objective Function which will either be set to minimize the delays or the costs.

Controls

The input controls for each experiment are used to control the different input variables when running multiple experiments. The seize times for each worker, their dispatching and tie breaker rule, and the type of priority will be used as input controls for the different experiments.

Responses

Responses are designed to display the output from the model and can either act as a purely informative display of a certain statistic of interest or an objective function that you are trying to optimize [Simio, 2019]. The KPIs as described in section 4.2 are used as the responses for the model.

Objective Function

The objective function is a response that defines the goal of the experiment and is used to summarize the performance of the system. This will either be the number of delays or the delays costs.

Results and the Optimal Solution

Simio Measure of Risk & Error (SMORE) plots, as seen in figure 6.1, will be used to graph the objective function. The SMORE plot shows the variability behind the values of the objective function. The objective function can be sorted to easily spot the optimal solution.

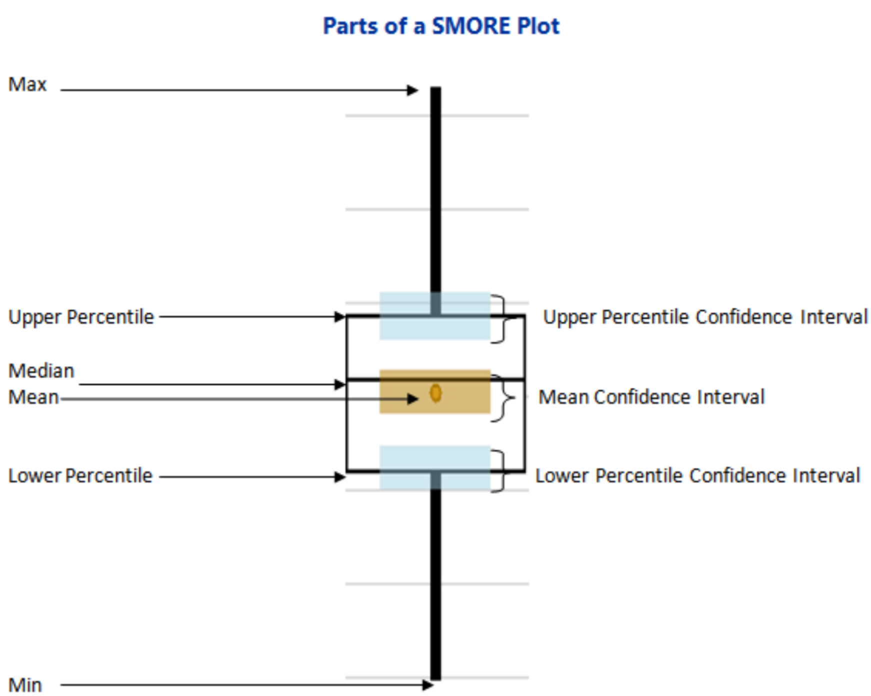

Figure 6.1 SIMIO Measure of Risk & Error [Simio, 2019].

OptQuest Add-In

Simio has the ability to create experiments allowing users to enter input values and run multiple replications to return estimated value of the system performance. OptQuest helps remove some of the complexity by automatically searching for the optimal solution. The simulation problem is described in the experiments and OptQuest searches for input controls to minimize the costs and the number of delays. OptQuest makes use of intelligent search methods and incorporates its special optimization algorithms alongside Simio's modeling power. Instead of using the algorithms to optimize a set of mathematical equations, it uses them to optimize a set of stochastic process interactions [Simio, 2017].

For additional information on the OptQuest engine, see the OptTek documentation on optimization [OpTek, 2015].

6.2 Performance of Modeled System

The performance of the modeled system will be assessed using the described experiments. The results will be used to determine the input variables for the OptQuest optimization algorithm.

The full results for the experiments can be found in appendix F.

6.2.1 Experiment 1: Call Time Workers get send to their Task

For this set of experiments the static flight priorities are used, the objective is set to minimize the number of delays and the PRIO-DD control scenario is used, which is the control scenario that will be closest to how controllers schedule their tasks.

Figure 6.2 shows the costs and number of delays for the different call times. It can be seen that the number of delays declines until around 8 minutes. After 8 minutes a steady state can be see in which the number of delays does not decline further if workers get called even earlier. The costs show the same decline, and between minutes 8 and 11 the lowest costs can be found. An increase in the costs can be seen if the call time is further increased. A longer call time results in longer tasks because workers have to move to the gate earlier and wait there if the flight entity did not arrive yet or the previous process did not finish yet. Flight entities cannot choose the best suitable worker, but only the worker that comes available, which will eventually result in a shortage of workers. Because no priorities can be set, flight entities have to be delayed longer, resulting in the higher costs. The number of delays does not increase when the call time is further increased, because the objective for this experiment was set to minimize the number of delays. Therefore the number of delays stays the same, but delays are longer and more expensive. If the call time will further increase, the number of delays will eventually increase as well.

The small decline around minute 20 looks quite strange because a further increase is expected. A possible cause could be the same fixed call time for all workers, resulting in call times that are good for some workers, but not for others. A closer look is needed to understand what causes these declines, which will be done in section 6.2.4.

Again, this experiment looks at the number of delays, which stays stable around minute 20. The

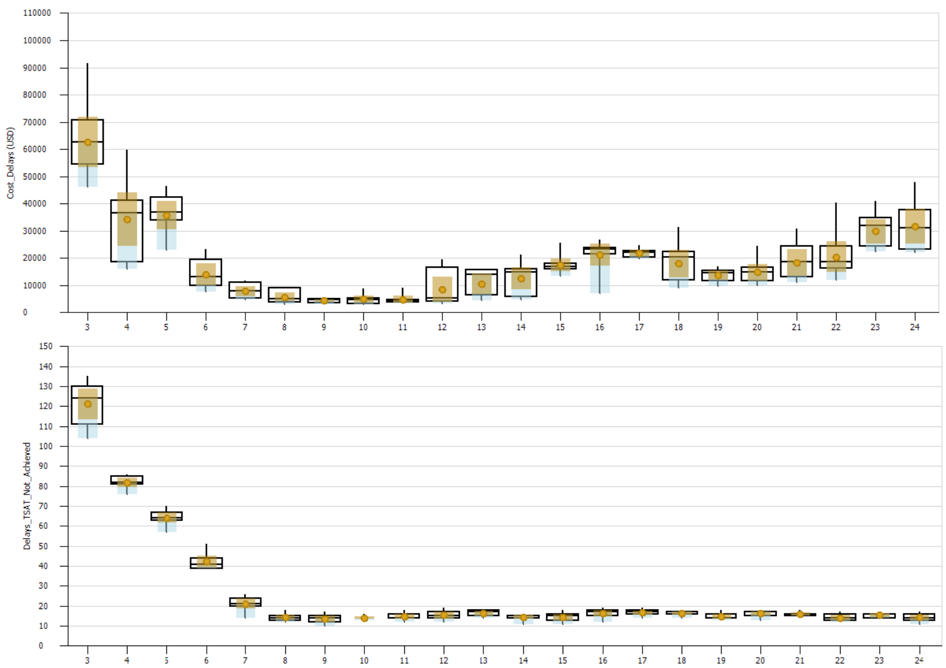

Figure 6.2 The costs and number of delays for different call times for workers. Priority: static, Objective: delays, Control Scenario: PRIO-DD

experiment shows us that minimizing the number of delays is not always the best method when looking from a cost perspective.

Table 6.4 shows the number of times the drive time of workers exceeds the time they have to travel for different call times. It can be seen that ramp workers need the least time, with ramp workers at K1 only needing 5 minutes and at K5 a drive time of 10 minutes max can be seen. Ramp workers work in smaller areas, therefore travel times are in general shorter than for workers that have to travel all around Schiphol. Looking at how this experiment is set up, the same process time for each process clearly is not the way to go. It results in workers that are waiting at the gate until the flight entity arrives while they could perform other tasks if the call time is set too high, or workers arriving too late at their location if it is set too low.

Toilet and water need the most time, and even with 24 minutes they still exceed their maximum travel time in some occasions. There is only a small pool of these type of workers to handle all aircraft. It is therefore possible that a worker has to accept a task on the other side of Schiphol because no other worker is available.

Table 6.4 The number of times workers have a longer travel time than available for different call times.

Minutes:	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**
RampK1	107.6	8.6	0.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RampK2	52.6	23.6	11.7	3.8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RampK4	150.3	60.9	6.7	1.6	2.3	1.3	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RampK5	74.3	51.7	10.5	4.2	2.5	2.7	1.9	0.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pushback	310.8	268.7	231.9	191.2	153.2	104.5	74.2	51.4	25.9	4	0.1	0	0	0	0	0	0	0	0	0	0	0
Cleaning	239	142.8	129.6	113	97.3	63.2	19.9	2.5	1.6	1.1	0.7	0.1	0	0	0	0	0	0	0	0	0	0
Catering	265.5	163.5	98.6	54	41.5	30.9	21.6	14.1	5	2	2.2	1.5	0.9	0.5	0.3	0.1	0.2	0.2	0	0	0	0
Refueling	147.9	58.6	17.9	11.8	7.9	5.2	5.6	5	4.4	2.5	0.9	1.8	1.2	0.2	0.4	0.5	0.1	0	0.1	0	0	0
Toilet	284.3	232.1	195.1	145.2	109.4	64.9	38.9	30.7	22.5	19.7	17.6	15.6	13.3	9.3	8.8	6.3	5.2	6.1	3.5	3.1	2.6	1.6
Water	303.2	241.5	205.1	155.4	113.5	71.1	48.5	38.5	30.9	28.7	24.1	24.9	19.1	19	14.1	12.4	13.7	11.7	8.5	7.3	5.8	4.8

The results of this experiment are used to give an indication for the upper and lower bound of the worker call time that is used by the optimizer. Longer drive times to reduce the amount of late arrivals of workers does not always means a better performance of the objective function. The optimizer takes costs into consideration and can make a better decision about when to increase the call time for workers or to accept a single delay that will benefit the other flights.

6.2.2 Experiment 2: Different Control Scenarios

The control scenarios as seen in table 6.2 are simulated to see the impact they have on the costs and number of delays. For each scenario the static flight priorities are used and different call times of 5, 10, 15 and 20 minutes are used to see which control scenario performs best under certain circumstances. The results for each control scenario can be seen in figure 6.3.

The 10 minute scenario performs the best with the lowest overall costs of around 5000 EUR. The same decline can be seen as in the previous experiment around minute 20, which performs better than the 15 minute scenario, which was expected when looking at the first experiment. The 5 minute scenario performs worst because workers arrive too often too late at their destination.

Control scenarios that use the Due Date (DD), Critical Ratio (CR) or Least Slack Time (LST) show the best performance for both number of delays as well as the delay costs. The FIFO and WAIT scenarios perform worst in all cases and should not be used as control strategy.

The control scenarios that do not use the PRIO dispatching rule seem to perform slightly better than the ones that do use priorities. The flight priorities are supposed to give a first indication of the importance, and therefore costs, that are associated with a flight delay. Looking at these results it can be concluded that the current method that is used to prioritize flights does not work as good as it should.

Because often the same priorities will occur, the tie breaker rule is important as well. This can be seen when looking at the PRIO-FIFO scenario which performs almost as bad as the other FIFO control scenarios. All three control scenarios (DD, CR & LST) perform almost the same and no conclusion can be made about which control scenario to choose based on these experiments. From a practical perspective, controllers need priorities to make decisions about which tasks to schedule first. They cannot calculate the costs for each flight or the impact a delay will have on the flight schedule. Therefore the three most promising control scenarios that do use priorities will be assessed in the next experiment using the proposed dynamic flight priorities. These will be compared with the current static method.

6.2.3 Experiment 3: Dynamic Flight Priorities

The dynamic flight priorities are used to see if switching ICA and EUR priorities for short delays result in a reduction in the costs. The CR, DD and LST control scenarios are the best performing control scenarios and will be used as input to compare the proposed dynamic priorities with the current static priorities. The different control scenarios will be simulated for a call time of 5, 8, 11, 14, 17 and 20 minutes to see how they perform when different call times are used. The results can be seen in figure 6.4. The solid blue lines are used to separate the experiments that use a different call time while the blue dotted lines separates the dynamic priorities on the left from the static priorities on the right.

Both experiments use exactly the same resources and amount of entities for each scenario, but differences can be seen in the output. The dynamic priorities outperform the static priorities for almost every control scenario, and for the best scenarios costs of around €5000 can be seen.

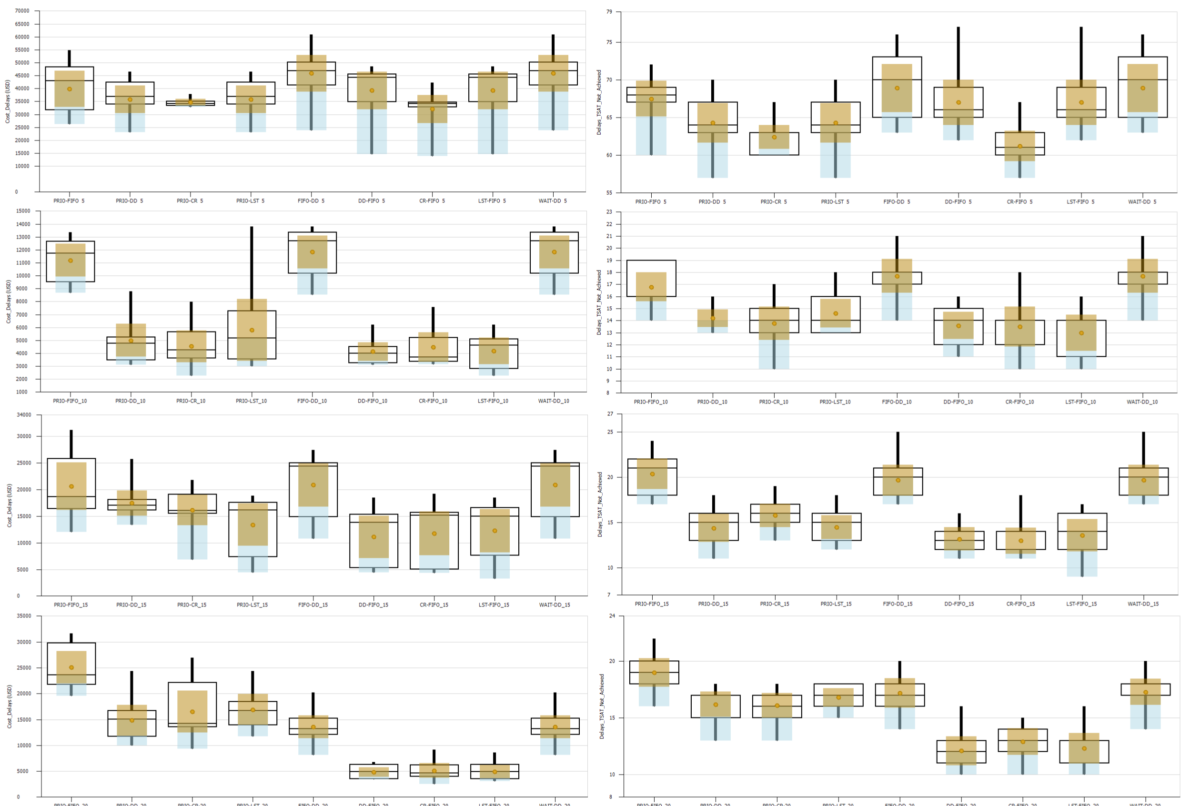

Figure 6.3 The costs (left) and number of delays (right) for different control scenarios at different call times. Priority: static, Objective: costs, Call time: 5 min (top) till 20 min (bottom)

Especially when the delay costs increase due to shortages, and decision making is more important, the dynamic priorities show their potential. No big differences can be seen between the different control scenarios, just like in the previous experiment. Also the same increase in costs around minute 17 and decrease in costs at minute 20 can be seen as in experiments one and two due to the fixed call times.

This experiment shows us the potential of using dynamic priorities instead of the current static scenarios. Again, no conclusion can be made about the best performing control scenario because they perform similar under the same circumstances. The next experiment will use the OptQuest optimizer to see if further improvements in the costs can be made using different call times for the different processes.

6.2.4 Experiment 4: The OptQuest optimizer

The previous experiment showed the importance of using flight priorities. This experiment will be used to further explore these findings using the OptQuest optimizer. The optimizer can make better decisions based on the different driving times of workers, and the influence a process delay has on the delay costs.

The objective function will be set to minimize costs and the control scenarios: PRIO-CR, PRIO-DD and PRIO-LST will be simulated using both the static and dynamic flight priorities. Table 6.5 gives an overview of the experiments that will be simulated.

Table 6.5 Experiment 4: Control scenarios used by the OptQuest optimizer.

Experiment Info				
Name	*Objective*	*Scenario*	*Priorities*	*Replications*
Costs_Under15_PRIO-LST	Costs	PRIO-LST	Under15	10
Costs_Under15_PRIO-DD	Costs	PRIO-DD	Under15	10
Costs_Under15_PRIO-CR	Costs	PRIO-CR	Under15	10
Costs_Normal_PRIO-LST	Costs	PRIO-LST	Normal	10
Costs_Normal_PRIO-DD	Costs	PRIO-DD	Normal	10
Costs_Normal_PRIO-CR	Costs	PRIO-CR	Normal	10

The OptQuest algorithm can quickly deplete the available computer resources when there are many scenarios with a high number of replications. The maximum number of scenarios is therefore set to 60 and the number of replications between 4 and 10. After running the OptQuest optimizer, the best performing scenarios will be ran again with the minimum number of replications set to 10 as calculated in appendix D and compared with the other best performing scenarios.

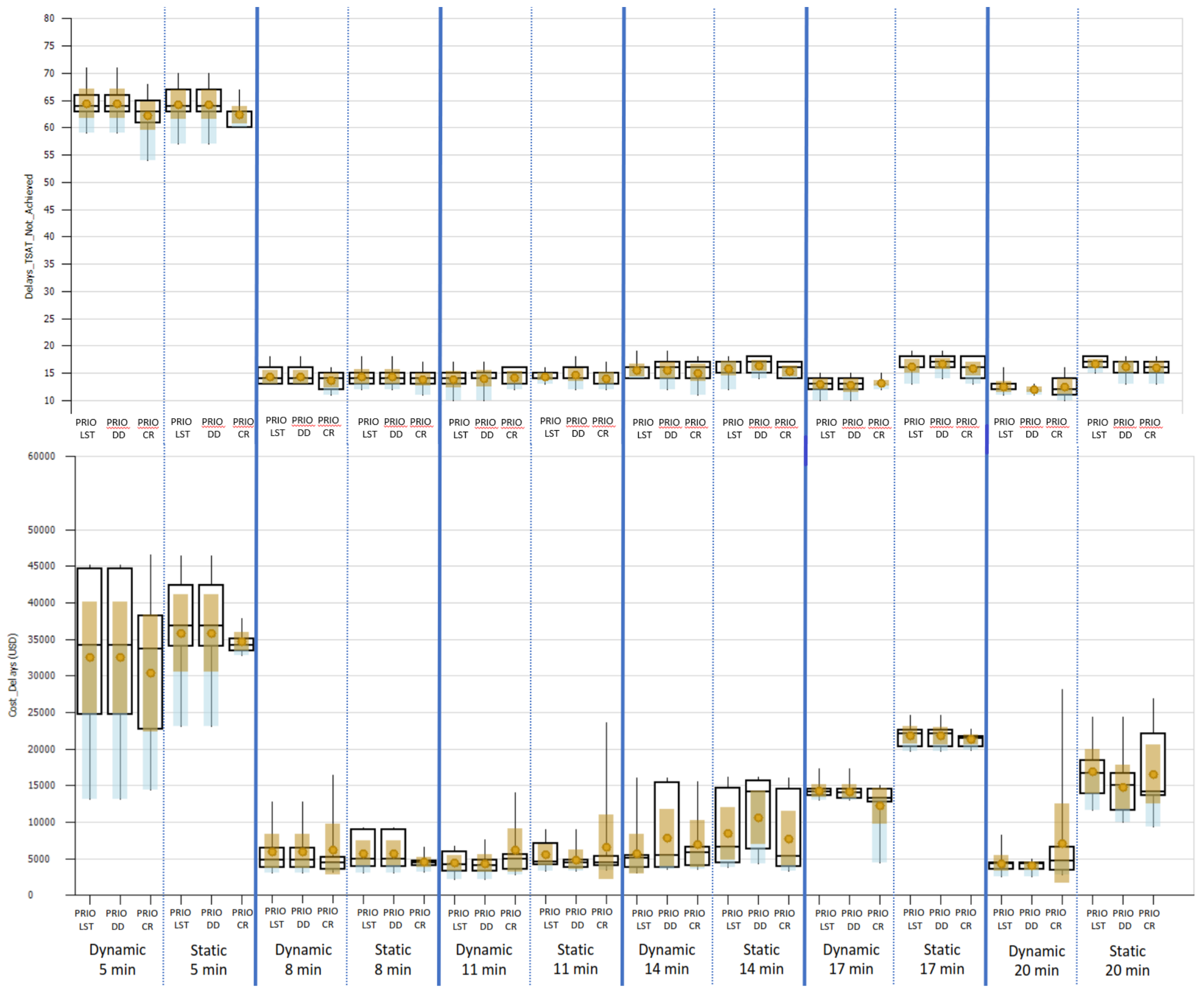

Figure 6.4 For each control scenario the proposed dynamic priorities (left of the dotted line) are compared to the current static (right of the dotted line) priorities for different call times (separated by the blue lines). On top are the delays, below the corresponding costs.

Table 6.6 The lower and upper bound for each control variable in the optimization based on the results of experiment 1.

Seize Time Workers [min]	
Refuel Hydrant	*5-20*
Pushback	*9-19*
Catering	*8-20*
Cleaning Asito	*8-20*
Toilet	*7-25*
Water	*7-25*
RampK1	*4-10*
RampK2	*5-12*
RampK4	*5-15*
RampK5	*5-15*
Cleaning Kluh	*8-20*
Refuel Tank	*5-20*

Figure 6.5 shows the results for running the experiments while table 6.7 shows the input that generated this outcome. For the delays the TSAT time is used (departure time + 5 minutes), and for the SPT time the departure time +10 seconds to compensate for the traveling time of the entities. The SPT time will be used to compare the model results with the SPT results of the current state as seen in table 4.1. The complete results for each OptQuest scenario are attached to the model, and a summary of the results can be found in appendix F.

Table 6.7 The input parameters as determined by the OptQuest optimizer and the corresponding output.

Experiment Info					**Seize Time Workers**												**Flights**		
Name	*Objective*	*Scenario*	*Priorities*	*Replications*	*Refuel Hydrant*	*Pushback*	*Catering*	*Cleaning Asito*	*Toilet*	*Water*	*RampK1*	*RampK2*	*RampK4*	*RampK5*	*Cleaning Kluh*	*Refuel Tank*	*Costs*	*Delays (TSAT)*	SPT
Costs_Under15_PRIO_CR	Costs	PRIO_CR	Under15	10	14	19	20	16	9	7	17	6	6	14	19	7	2250	8	41
Costs_Under15_PRIO-DD	Costs	PRIO-DD	Under15	10	5	14	5	11	6	23	16	5	9	7	11	6	2270	10	49
Costs_Normal_PRIO-DD	Costs	PRIO-DD	Normal	10	6	9	15	9	8	10	18	8	9	8	19	9	3335	13	65
Costs_Under15_PRIO-LST	Costs	PRIO-LST	Under15	10	6	13	8	10	7	24	17	5	13	7	13	6	3382	11	56
Costs_Normal_PRIO_CR	Costs	PRIO_CR	Normal	10	11	12	11	13	13	15	11	11	12	12	12	13	3885	14	70
Costs_Normal_PRIO-LST	Costs	PRIO-LST	Normal	10	10	12	12	13	12	14	12	10	12	12	14	14	4254	14	68

With exactly the same resources and amount of entities for each scenario, some differences can be seen in the output. The key findings will now be discussed:

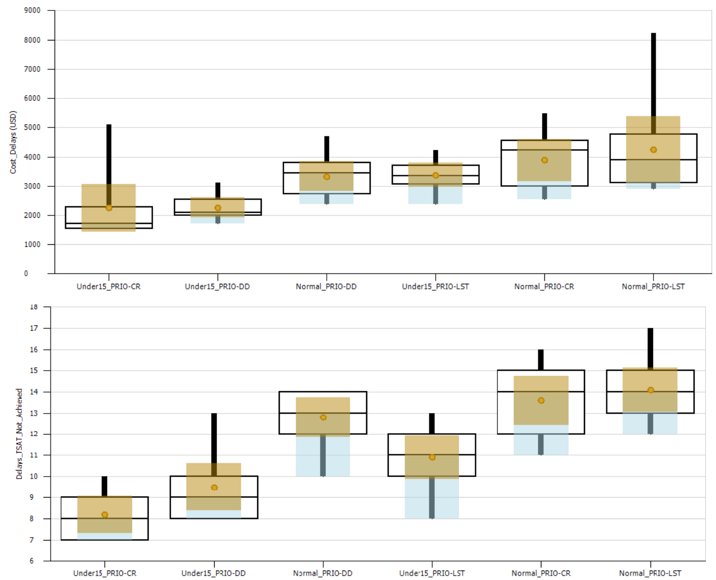

Figure 6.5 Final result between the different control scenarios using the OptQuest optimizer and the static and dynamic flight priorities. Top figure shows the costs, bottom figure the number of delays.

Overall Results

Because the database does not contain information on how often and when the TSAT is shifted due to delays, the SPT results are used to compare the simulation results with the current state. Table 6.8 compares the output from the simulation model with the values found in the data analysis.

Looking at the number of delays and the SPT results, the simulation model performs a lot better compared to the current state using exactly the same amount of resources and flight entities that have to be handled. For the flight entities the real arrival time is used and the departure time that is based on the arrival time + the available ground time. The results show that the global push-pull

Table 6.8 Comparison of the SPT results for the real system and the simulation model, using the average values of the static simulation runs.

	SPT			
	Current State	*Model*	*Increasement*	*Factor*
ICA	30	8	-73%	3,8
EUR	135	29	-79%	4,6
Commuters	115	31	-73%	3,7
Average:	*93*	*23*	*-75%*	*4,0*

strategy has the potential to perform much better than the current strategy. Not every process is trying to schedule the tasks when it suits them best, but each process gets a request to perform a task based on the best time for the flight entity. The global queue for entities gives us the same scheduling problem, but instead of each process looking for its own optimal solution, each task is scheduled to find the best solution for the overall turnaround process. This method works much better and will eventually lead to a more global and integral solution.

A note should be made that the simulation model does use a lot less constraints when scheduling its tasks than there are in the real system. Constraints such as: worker skills, available flight slots, or external disturbances that can influence the performance of the system are excluded from the simulation, and therefore a higher performance is expected of the simulation model. It does show however that the problem does not lie in a shortage of workers, and shows us that the airport could, in theory, handle much more aircraft. In section 6.3 the results from the model will be compared to that from the real system and scaled accordingly to give an indication of the potential the new control scenarios have.

OTS, OTP and OTD Performance

The OTS, OTP and OTD are used to get a closer look at the different processes and which processes cause the most turbulence in the turnaround process. They can help understanding the reasons for delays, and what causes them. For the final experiment the results for the OTS, OTP and OTD can be seen in tables 6.9, 6.10 and 6.11.

OTS

The processes where the OTS is not achieved are the processes that do not have enough workers available, or have longer driving times and therefore start their task too late. Again, the pull strategy for all arriving oriented processes can clearly be seen when comparing these results with the results from the data analysis in section 4.3.2. Much better results can be seen for all processes.

Table 6.4 shows that the toilet and water processes perform the worst on this KPI. As seen in table 6.4, the driving time for these processes can sometimes be very long, resulting in a delayed start. Both processes have a small resource pool, and the relatively short tasks are not very critical. It is therefore not strange that these processes score bad on this KPI, just like in the real system.

In general, ramp workers score much better with almost no starts that are too late. The ramp workers can switch between areas if no workers are available, just like in the real system. But in the real system the decision to do this is much slower and controllers have to consult with other processes if it is possible to switch workers between different areas. The result is that decisions are sometimes made too late, resulting in an OTS that is not achieved.

Table 6.9 The number of times the OTS is not achieved for the final comparison between the different control scenarios.

Name	*Refueling*	*Cleaning*	*Catering*	*Toilet*	*Water*	*Ramp K1*	*Ramp K2*	*Ramp K4*	*Ramp K5*	*Pushback*
Costs_Normal_PRIO_CR	19	10	33	53	48	0	6	1	0	11
Costs_Normal_PRIO-DD	31	34	32	127	91	0	8	1	4	83
Costs_Normal_PRIO-LST	22	10	30	58	47	0	6	0	0	12
Costs_Under15_PRIO_CR	11	8	28	100	175	1	9	1	0	5
Costs_Under15_PRIO-DD	34	14	126	207	36	1	13	0	5	6
Costs_Under15_PRIO-LST	31	13	59	166	31	0	11	0	5	6

OTP

The OTP is very useful for the data analysis because it shows that the real problem for most processes lies in the fact that the CHIP tasks do not correspond well with the real duration. The simulation model uses process times as input and multiplies these with a normal distribution. If the process time takes longer than the average value + 1 minute, the OTP is not achieved. The OTP is therefore not as important as for the data analysis because it is an input parameter. It can be seen that processes with a longer process time are less influenced by the correction of 1 min and a distribution of almost 50% can be seen. Because toilet, water and pushback tasks are relatively short, these almost all fall within the 1 min error margin.

Table 6.10 The number of times the OTP is not achieved for the final comparison between the different control scenarios.

	On Time Performance									
Name	*Refuel*	*Cleaning*	*Catering*	*Toilet*	*Water*	*Ramp K1*	*Ramp K2*	*Ramp K4*	*Ramp K5*	*Pushback*
Costs_Normal_PRIO_CR	74	29	37	2	10	47	35	54	28	0
Costs_Normal_PRIO-DD	72	30	35	2	10	48	35	57	31	0
Costs_Normal_PRIO-LST	70	29	37	2	9	46	39	54	31	0
Costs_Under15_PRIO_CR	75	29	40	2	9	47	32	58	30	0
Costs_Under15_PRIO-DD	74	28	39	2	11	46	35	55	30	0
Costs_Under15_PRIO-LST	75	28	37	3	12	47	34	53	31	0

OTD

The OTD is one of the most important KPIs because the result has a direct impact on the turnaround process. Just like in the data analysis it can be seen that the OTS is not achieved many of the times for toilet and water tasks but does perform really good on the OTD. Arrival oriented processes score good on this KPI, and the only process that shows a bad performance on this KPI is the pushback. Pushback is the last process in the TA process and therefore each task influences the performance. Because there is no slack time added in the calculation of the OTD, a delay of less than a second will results in the OTD not achieved. For the SPT an extra 10 seconds was used (traveling time entities), resulting in delays that are much lower. This also shows us that most of the OTDs that are not achieved fall within this 10 seconds gap, and could therefore be neglected.

For the data analysis a closer look was given to the OTD to see how much percent of the delays was caused by the process itself and how much delays came from other reasons. Again, the model uses the process times as input and this KPI is therefore less useful to asses the results of the simulation model.

Table 6.11 The number of times the OTD is not achieved for the final comparison between the different control scenarios.

On Time Delivery										
Name	*Refuel*	*Catering*	*Cleaning*	*Toilet*	*Water*	*Ramp K1*	*Ramp K2*	*Ramp K4*	*Ramp K5*	*Pushback*
Costs_Normal_PRIO_CR	10	4	13	0	0	10	6	2	0	226
Costs_Normal_PRIO-DD	9	4	9	0	0	8	3	3	0	254
Costs_Normal_PRIO-LST	10	3	11	0	0	10	4	3	0	227
Costs_Under15_PRIO_CR	9	2	8	0	0	7	2	3	0	224
Costs_Under15_PRIO-DD	10	3	10	0	0	6	3	3	0	219
Costs_Under15_PRIO-LST	10	4	9	0	0	7	3	4	0	220

Conclusions on the new Control Scenarios

Call Time Workers

When looking at the results the values for each process look similar to the findings found after the first experiment. The only differences are the ramp K1 workers which have longer call times than expected. The reason for this can be found in the way the model is set up. When entities arrive and are waiting in the global queue to be handled, they request the secondary resources they need. A worker then decides to seize that particular entity and plans a visit to the same gate as the entity. Half of the ramp tasks is departure oriented, and the tasks are scheduled based on the departure time. Requesting workers close to the departure time and therefore closer to the minimum driving time is tricky, if no workers are available this will immediately result in a delay and extra costs. Especially for short turnarounds at K1, where all processes have to work within a tight time schedule. By choosing higher call times for the workers, more slack time is created in case there are no workers available at that particular moment. The OptQuest optimizer tries many different scenarios and can then make this decision. The result is a higher call time for ramp workers but lower delay costs.

All other processes have call times close to the minimal driving time or even slightly under. For arriving oriented processes, starting a task a little later than possible is not necessarily bad, and reduces the chances of unnecessary waiting at the gate for the entity to arrive or the previous process to finish. It is difficult to decide when a delay outweighs an earlier call time with so many variables, using an optimizer is therefore extremely useful in these cases.

Looking at the delay costs found by the optimizer (€2200) compared to that from experiment 1 (€5000), a decrease in the costs of more than 50% can be seen. It can be concluded that different call times should be used by the processes because workers work in areas with different sizes and have different drive velocities.

Control Scenarios

To focus purely on the control scenario without looking at the different flight priorities, the costs for both the static and dynamic priority are added together in table 6.12.

Table 6.12 The results for different control scenarios. The costs for both the static and dynamic flight priorities are added together for each control scenario.

Control Scenario	*Costs (€)*	*Increasement*	*Delays*	*Increasement*
PRIO-DD	5605	*Base scenario*	25	*Base scenario*
PRIO-CR	6135	+9,4%	22	-12%
PRIO-LST	7636	+36,2%	25	0%

The PRIO-DD control scenario shows the best performance overall with a total cost of €5605 when the static and dynamic scenarios are added together. The PRIO-CR scenario performs 9,4% worse than the PRIO-DD, but is the best performing scenario when dynamic priorities are used, both for costs as well as number of delays. This shows us that the PRIO-CR control scenario looks the most promising when dynamic flight priorities are used, but because the difference between the two is only 20 euros, no conclusion can be made between these two control scenarios. The PRIO-LST scenario scores on average 36,2% worse than the PRIO-DD control scenario. The LST scenario is the only scenario that does not look at the departure time, but the slack time between the process and its successor. This shows us that looking at individual processes is less efficient than looking at the overall turnaround. Individual processes tend to work to local optimum instead of the preferred global optimum. For both the static and dynamic priorities this control scenario performs the worst and is therefore not advised.

Flight Priorities

A bigger difference can be seen for the different flight priorities. All scenarios using the dynamic flight priorities have better results than the same scenarios using the static flight priorities when looking at the costs. The number of delays is not always lowest for the best performing control scenario, showing again the importance of looking at the delay costs instead of the number of delays.

Table 6.13 Performance of the dynamic flight priorities compared to the static flight priorities.

	Costs (€)			**Delays**		
Control Scenario	*Static*	*Dynamic*	*Increasement*	*Static*	*Dynamic*	*Increasement*
PRIO-DD	3335	2270	-31,9%	13	10	-23,1%
PRIO-CR	3885	2250	-42,1%	14	8	-42,9%
PRIO-LST	4254	3382	-20,5%	14	11	-21,4%
Total	*11474*	*7902*	*-31,1%*	*41*	*29*	*-29,3%*

On average an improvement of 31,1% in the costs and 29,3% in the number of delays. For the PRIO-CR control scenario these numbers are even higher with 42,1% and 42,9% respectively. All control scenarios perform significantly better than the same control scenarios used in combination with static flight priorities.

The delay costs have a very dynamic behavior and are different for each flight, the current static method is therefore not sufficient. Using dynamic flight priorities that will choose EUR flights over ICA flights for short delays shows a significant improvement in the costs and number of delays. In the current state, controllers try to minimize the number of delays without looking at the impact a delay has on the delay costs. Only for longer delays the OCC and HCC will be consulted to make a decision. This experiment shows the importance of looking at the shorter delays as well. For short delays, switching ICA and EUR priorities can significantly reduce the total costs due to delays.

6.3 Comparison of Current and Modeled System Performance

As already mentioned in section 6.2.4, the simulation model performs a lot better than the real system. The simulation model is a simplification of the real system and shows how good the real system could perform if there were no external disturbances. To give an indication of the number of delays this could save, the simulation results are compared with that from the real system using the SPT results as seen in table 6.8. Because there are no good cost estimations for the day of simulation, costs are not compared because it will contain too much uncertainty to compare them with the real system. For different control scenarios that are simulated with the same model, cost comparisons can be made.

Table 6.14 shows the comparison between the SPT results as determined by KLM with the results of the average value of the static simulation runs. The SPT correction is applied and it can be seen that using the global queue from which entities pull their resources has a positive effect on the performance of the model. On an Annual basis this could reduce the number of delays with 53692 delays or 72,4% using the same amount of workers.

Table 6.14 Comparison between model results and real results.

	Model Delays	SPT Correction	Corrected Model Delays	Yearly basis	Percentage
Real System	203	1,0	203	74146	100%
Simulation Model	14	4,0	56	20454	27,6%
Difference			*147*	*53692*	*72,4%*

Table 6.15 gives a good comparison between the used static and dynamic priorities. The same model with the same input is used, giving a reliable outcome which is good to compare. The dynamic priorities show their potential, with savings between 800 and 1600 euros per day. Calculating the annual saving, and average saving of 1,7 million can be seen and for the PRIO-CR even 2,4 million euros and a reduction of almost 9000 delays.

Table 6.15 Annual savings between static and dynamic priorities.

	Model		**Correction**	**Corrected Model**		**Annual Savings**	
	Delays	*Costs (€)*	*SPT multiplier*	*Delays*	*Costs (€)*	*Delays*	*Costs (€)*
Difference_PRIO_DD	3	1065	4,0	12	4224	4383	1542816
Difference_PRIO_CR	6	1635	4,0	18	6540	8766	2388735
Difference_PRIO_LST	3	872	4,0	12	3488	4383	1273992
Average	*4*	*1191*	*4,0*	*14*	*4751*	*5844*	*1735181*

CHAPTER

7

CONCLUSION AND RECOMMENDATIONS

This chapter provides the main conclusions and recommendations for further research coming forth from this thesis. A recapitulation of the research is given to answer the main research question presented in section 7.1. This is followed by the conclusions on the process analysis and the results of the performed experiments by the developed model.

7.1 Main Research Question

Based on the scope of the research and the research objective, the main research question has been formulated in chapter 1 as:

> *Which integral control scenario should be used by a Transport Logistic Control Tower in order to optimize the overall performance of KLM's apron processes?*

Based on the current state and data analysis the developed simulation model showed a good representation of the real system. The model is used to assess different control scenarios for the apron and showed a significant improvement on the performance of the apron processes. Using different control scenarios showed us that departure oriented control scenarios perform better than others. Also, it can be concluded that a control scenario should be chosen that takes the departure time into account, and not only to the latest completion time of a particular process. The two most promising control scenarios are based on the Due Date or the Critical Ratio.

The proposed dynamic flight priorities showed an advantage over the original static flight priorities for all control scenarios. Choosing EUR flights over ICA flights for short delays showed an average improvement of 31,1% or 1,7 million euros a year in the delay costs.

To improve the overall performance of KLM's apron processes, a departure oriented control scenario that takes the departure time into account should be implemented in combination with dynamic flight priorities.

In the different chapters, the findings of the sub questions supported the answer on the main research question and are summarized in the following sections:

7.2 Conclusions on the Current State Analysis

This chapter answers the first two sub questions. The first sub question:

1. *What is the current state practice of KLM's apron processes?*
    - *What is the current operational practice: process steps, information flows, decision rules, operational control and execution.*

The current state analysis showed us that KLM has very limited insight in the integral picture of their apron control. It is highly complex, non-transparent and has many different input variables. This is mainly driven by:

- **Control standards -** are only limited available in the form of flight priorities. Controllers plan their tasks using their own methods and assumptions, without the use of a uniform planning strategy. Often controllers make decisions on gut feeling without really understanding why they make these decisions. Because strategies vary for each controller for each day, little is known about what the best planning strategy is.
- **Minimal integration -** can be seen between the different processes. Controllers for each process plan their own tasks as good as possible, only consulting the other processes if they are of direct influence on their own process. This leads to processes optimizing their own process instead of optimizing the overall TA.
- **Important decisions -** such as shifts in the flight schedule or choosing which flights to delay are taken by the HCC and OCC and not by the controllers.

- **Big differences between the different departments -** can be seen, both in their methods of working as well as technological advancements. These big differences make it difficult to set up an integral control scenario which works for all departments.

Given the current state of the process, the second sub-question can be answered:

> 2. *What alternative scenarios can be used to control the apron processes?*

The following alternative control scenarios are used for further research:

- The use of an umbrella body in the form of the introduced TLCT, that will be used for the communication between the processes.
- A pull strategy where aircraft pull the necessary resources from a global pool of resources.
- Varying the time before workers move to their destination.
- Different control scenarios for workers to choose their next tasks based on: priority, first in queue, due date, critical ratio, least slack time or a combination of these control scenarios.
- Use dynamic flight priorities instead of the current static method.

7.3 Conclusions on the Data Analysis and KPIs

The third sub question is answered in this chapter:

> 3. *How is the process currently monitored: data availability, KPIs?*

Performance of the different apron processes are currently not measured adequately, clear links between performance drivers and accountability to control of these drivers is lacking. This conclusion is supported by the following findings:

1. **Data analysis:**
    - *The data quality* - shows big differences between the different apron processes. Especially the processes which do not use the optimizer. These processes do not log important data about the process because this has to be done manually and is often forgotten. This makes it difficult to realize an integral control approach.
    - *OTS performance* - shows us that processes almost never start at the earliest possible time, loosing valuable slack in the remainder of the TA process.

- *OTP performance* - shows that the task lengths as calculated by CHIP does not correspond well with the duration of tasks in the real system. This makes it difficult for controllers to schedule tasks because the chance that a preceding process has a delay that will have an impact on next processes is significant.
- *OTD performance* - is the result of the bad performance of both the OTS and OTP. By comparing the OTP and OTD it could be seen that many of the delays could be prevented if the process started on time, showing the importance of starting on the scheduled time.

2. **Key Performance Indicators:**

   (a) *The costs* - This is the most important KPI and is used to make a final decision.

   (b) *Number of Delays* - are used to see the overall performance of the new control scenarios.

   (c) *Effectiveness:*

   To measure the effectiveness of the TA process and apron processes the OTS, OTP and OTD are introduced. These KPIs have proven to be the most suitable when tracking the control performance of the apron and show the root causes of the delays.

7.4 Conclusions on the Simulation Model

This chapter is used to answer the fourth sub question:

4. How can alternative control scenarios be assessed using simulation?

The developed model in this research simulates the effect of different control scenarios for the apron process to assess the impact on performance. The following characteristics support these findings:

- The model has been validated by:
  - Key stakeholders of the apron processes
  - Comparing the model input-output transformation to corresponding input-output transformations for the real system by comparing results from the model with actual aircraft turnarounds.
  - Using real inputs for the duration of processes and the amount of workers.

- The model KPIs are used as output of the model to make a trade-off between different control scenarios:
  - Costs to combine all KPIs to one final KPI.
  - The OTS, OTP & OTD to asses the overall performance as well as individual processes.
- The various data sources have been merged and filtered to create input that resembles the actual process. Especially for all departure oriented processes adjustments had to be made to make them suitable for model input.

7.5 Conclusions on the Results of the Model: Possible Improved Control Scenario

This chapter is used to answer the final sub question:

> *5. How do the redesigned control scenarios improve the performance of KLM's apron processes?*

The model has given insight of the behavior of the different apron processes and how different control scenarios affect the performance of the model. The flight schedule and corresponding processes differ per day but with the one day simulation run the following can be concluded:

- An integral control scenario using a global queue shows the advantages of the TLCT having all information at one location. Flight entities determine when process tasks have to be scheduled instead of the current situation where each process determines when their tasks are being scheduled. In theory this could reduce the number of delays by 72,4% compared to the real system.
- A departure oriented control scenario using the Critical Ratio or the Due Date should be chosen that uses the departure time of the flight, and not the latest finish time of a process.
- Controllers should know the costs that belong to a delay, or an estimation of these costs. The results showed that the minimum number of delays does not mean the lowest delay costs. By adding costs to tasks, a controllers can make a better decision on which flight to delay.
- Dynamic flight priorities should be implemented because they result in an average cost reduction of 31,1% or 1,7 million euros a year compared to the same control scenarios using the current static priorities.

7.6 Recommendations

This section will describe the recommendations following this research. The following recommendations are needed to make a final decision on which control scenario must be used to control the apron processes of KLM:

Recommendations for a Future Control Scenario:

1. **Improve the way data is logged by the HHT.** A lot of the data show measurement errors because task times take longer than the actual ground time of an aircraft. Workers on the platform often forget to end their tasks when finished or to log important points of their process making it difficult to use the data as input. To improve the quality of the data an automated way of logging key points in the TA process should be implemented for the HHT. This way human errors can be minimized, resulting in better data quality.
2. **Improve the way the task duration is calculated.** The task duration as calculated by CHIP does not correspond to the duration of the tasks in the real system. A better calculation method is needed to improve the input used by the CHIP optimizer. The advise is to use machine learning, which combines the current calculation methods with historical data from the HubDB to get task duration that are more comparable with the real task duration.
3. **Add extra priority layers in CHIP.** This is needed to further improve the dynamic flight priorities as well as the way controllers schedule their tasks. By adding extra layers of priorities or an indication of the delay costs, controller get a better understanding about which flight to delay. Colors could be used to make it easy to see important flights for example.
4. **Further improve the dynamic flight priorities.** This research showed the advantages of using dynamic flight priorities over static flight priorities. Still, only three priority levels are used while there are big differences in importance and costs within these levels as well. Especially when an optimizer will be used in the future, each flight should have a cost attached to it that will be used to schedule each flight. More research should therefore be performed on the different priorities between the flights.

Recommendations for the Organizational Structure:

1. **Merge processes with the same kind of tasks.** Some processes are very similar but have different controllers that each schedule their own process tasks. This means extra and unnecessary constraints to solve the same optimization problem, resulting in sub optima for the different processes. Good examples of departments that could be merged are:

   (a) *Towing and pushback department:* These departments use the same resources and merging the two could directly benefit the efficiency of both departments.

   (b) *Hydrant and tank department:* Not all aircraft can be refueled by tankers, but merging these departments allows for more interchangeability, and in a better efficiency.

   (c) *EUR and ICA cleaning teams:* These are currently performed by two separate companies, not allowing any interchangeability between the two. It is not preferred to let one company handle all aircraft, but because the cleaning tasks for both are very similar it is recommended to introduce a small group of workers that could handle both type of aircraft. This allows interchangeability between the two on a small scale when one of the two has shortages for example, improving the overall efficiency.

Recommendations for Future Research:

1. **Use more days to assess simulation outcome.** To identify the best performing control scenario a one-day run is not sufficient. Different control scenarios could perform better depending on different situations. More days have to be assessed to make a decision which scenario best suits all days of operation. Should only days without major disturbances be considered as input for the model, or include days with major disturbances such as bad weather conditions as well?

2. **Vary the number of workers.** In the current model the amount of workers is considered fixed and by trying different control scenarios the goal is to reduce the number of delays and costs. By varying the number of workers as well, a trade-off between the number of delays and the amount of workers could be made.

APPENDICES

Swim Lane Analysis

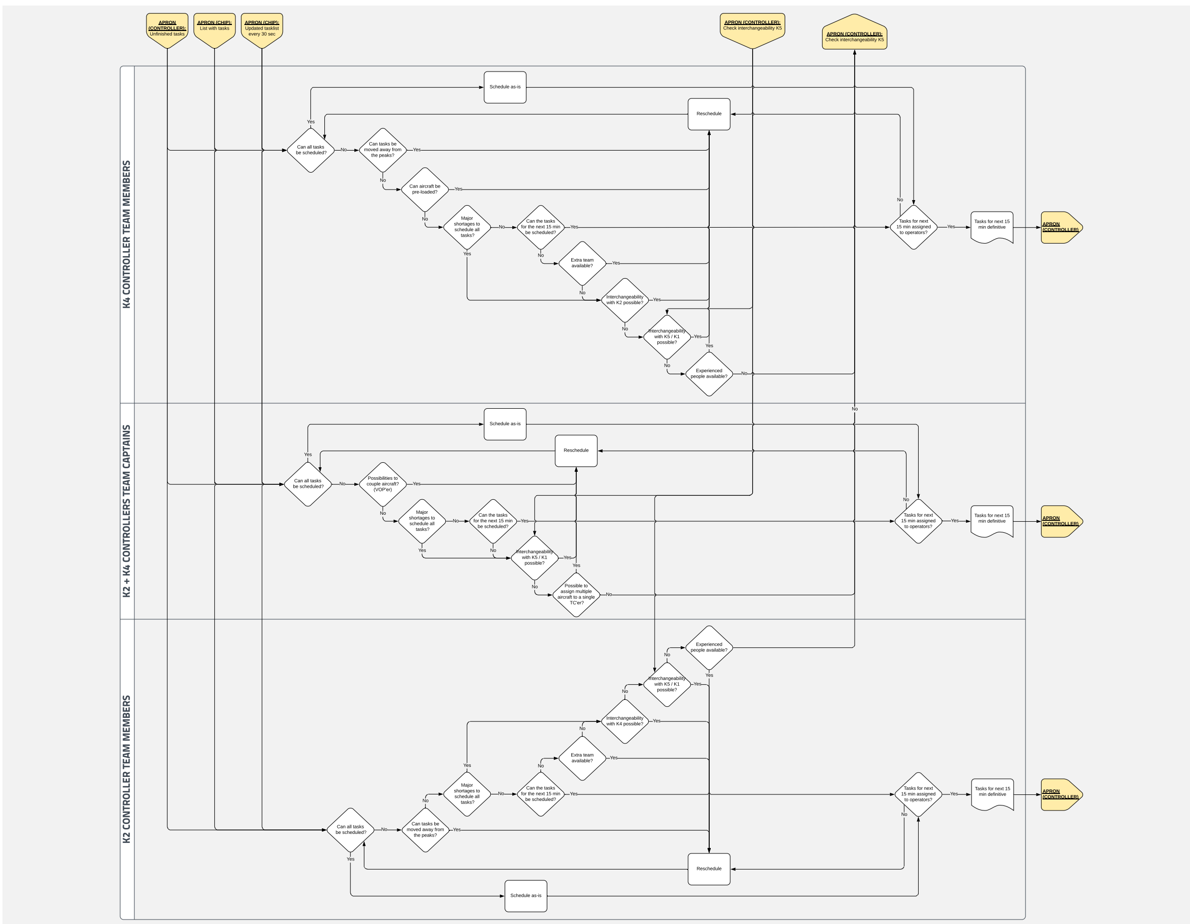

Figure A.1 Swim lane analysis for ramp K2 and K4 controllers.

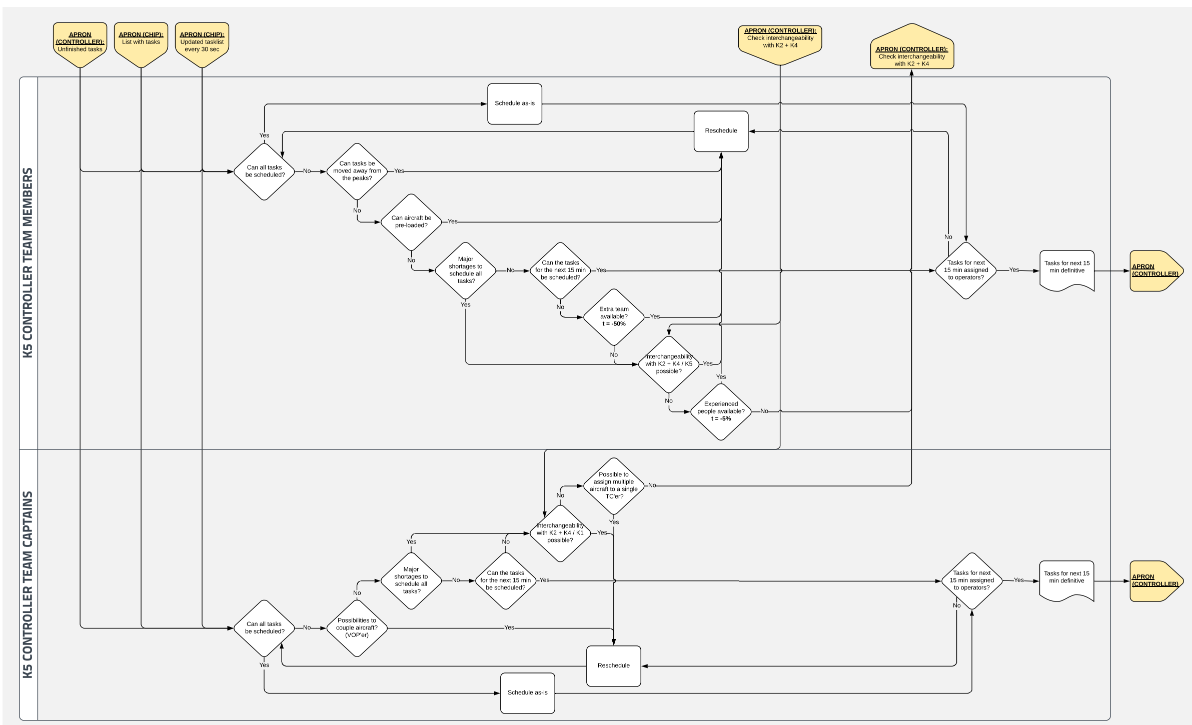

Figure A.2 Swim lane analysis for ramp K5 controllers.

Data Overview

Table B.1 Results CHIP data wide-body aircraft.

Refuel:			**Pushback:**		**Water:**	
Duration:	**CHIP**	**Real**	**CHIP**	**Real**	**CHIP**	**Real**
Rows	3793	3793	3655	3655	4069	4069
Average	50	49	23	25	27	25
Mode	55	55	17	11	32	28
Median	53	49	21	23	30	28
SD	21	18	11	14	9	10
KPI not met:	*Nr. of times*	*Percentage*	*Nr. of times*	*Percentage*	*Nr. of times*	*Percentage*
OTS	3724	98.18%	2759	75.47%	3517	86.44%
OTP	1379	36.36%	2483	67.92%	1379	33.90%
OTD	621	16.36%	2759	75.47%	276	6.78%
OTD & OTP	69	1.82%	1724	47.17%	138	3.39%
OTD & OTP & OTS	69	1.82%	1379	37.74%	138	3.39%
OTD & OTS	621	16.36%	2345	64.15%	276	6.78%

Table B.2 Results CHIP data wide-body aircraft.

Toilet:		**Cleaning:**		**Ramp (load):**		**Ramp (unload):**	
CHIP	**Real**	**CHIP**	**Real**	**CHIP**	**Real**	**CHIP**	**Real**
3311	3311	1241	1241	21519	21519	21036	1241
13	11	48	53	54	62	36	48
11	9	44	56	49	68	30	44
12	11	48	52	50	62	33	48
3	2	10	17	9	14	9	10
Nr. of times	*Percentage*	*Nr. of times*	*Percentage*	*Nr. of times*	*Percentage*	*Nr. of times*	*Percentage*
3311	100.00%	1172	94.44%	2552	11.86%	10139	48.20%
828	25.00%	828	66.67%	14898	69.23%	9035	42.95%
345	10.42%	345	27.78%	12346	57.37%	12415	59.02%
0	0.00%	276	22.22%	10483	48.72%	8759	41.64%
0	0.00%	207	16.67%	345	1.60%	3586	17.05%
345	10.42%	276	22.22%	1793	8.33%	7242	34.43%

Table B.3 Results CHIP data wide-body aircraft.

Deboarding:		**Boarding:**		**CabinCheck**		**CabinSecurityCheck**	
CHIP	**Real**	**CHIP**	**Real**	**CHIP**	**Real**	**CHIP**	**Real**
16484	16484	16484	16484	-	-	3655	3655
37	39	73	77	9	9	14	14
45	49	90	97	9	9	14	14
38	39	75	78	9	9	14	14
20	24	20	24	3	3	3	3
Nr. of times	*Percentage*	*Nr. of times*	*Percentage*	*Nr. of times*	*Percentage*	*Nr. of times*	*Percentage*
5587	33.89%	5587	33.89%	-	-	-	-
10139	61.51%	10139	61.51%	-	-	-	-
8414	51.05%	8414	51.05%	-	-	-	-
7311	44.35%	7311	44.35%	-	-	-	-
2897	17.57%	2897	17.57%	-	-	-	-
4000	24.27%	4000	24.27%	-	-	-	-

Table B.4 Results CHIP data narrow-body aircraft.

Refuel:			**Pushback:**		**Water:**	
DurationCHIP	**CHIP**	**Real**	**CHIP**	**Real**	**CHIP**	**Real**
Rows	9449	9449	9587	9587	7725	7725
Average	21	21	15	18	5	3
Mode	16	21	12	16	4	3
Median	21	20	13	17	4	3
SD	7	6	9	10	3	2
KPI not met:	*Nr. of times*	*Percentage*	*Nr. of times*	*Percentage*	*Nr. of times*	*Percentage*
OTS	6414	67.88%	6966	72.66%	7449	96.43%
OTP	2483	26.28%	6276	65.47%	1586	20.54%
OTD	2138	22.63%	6690	69.78%	0	0.00%
OTD & OTP	483	5.11%	3862	40.29%	0	0.00%
OTD & OTP & OTS	414	4.38%	3449	35.97%	0	0.00%
OTD & OTS	2069	21.90%	6276	65.47%	0	0.00%

Table B.5 Results CHIP data narrow-body aircraft.

Toilet:		**Cleaning:**		**Ramp (load):**	
CHIP	**Real**	**CHIP**	**Real**	**CHIP**	**Real**
9725	9725	13311	13311	22967	22967
5	4	9	13	31	31
6	4	11	10	26	26
5	4	8	12	29	29
2	1	7	8	9	10
Nr. of times	*Percentage*	*Nr. of times*	*Percentage*	*Nr. of times*	*Percentage*
9449	97.16%	12759	95.85%	414	1.80%
2414	24.82%	10277	77.20%	1931	8.41%
621	6.38%	2069	15.54%	1931	8.41%
0	0.00%	1586	11.92%	1931	8.41%
0	0.00%	1517	11.40%	0	0.00%
621	6.38%	2000	15.03%	0	0.00%

Table B.6 Results CHIP data narrow-body aircraft.

Ramp (unload):		Ramp (TurnAround)		Deboarding:	
CHIP	**Real**	**CHIP**	**Real**	**CHIP**	**Real**
23657	23657	6621	6621	27726	27726
30	30	64	64	27	28
25	25	62	74	33	30
28	28	62	63	33	30
10	10	9	9	9	10
Nr. of times	*Percentage*	*Nr. of times*	*Percentage*	*Nr. of times*	*Percentage*
3173	13.41%	1172	17.71%	10070	36.32%
759	3.21%	1241	18.75%	14484	52.24%
966	4.08%	1104	16.67%	12691	45.77%
759	3.21%	966	14.58%	10759	38.81%
0	0.00%	138	2.08%	2552	9.20%
138	0.58%	276	4.17%	4414	15.92%

Table B.7 Results CHIP data narrow-body aircraft.

Boarding:		CabinCheck		CabinSecurityCheck	
CHIP	**Real**	**CHIP**	**Real**	**CHIP**	**Real**
27726	27726	-	-	1035	1035
55	56	7	7	10	11
65	61	7	7	9	10
65	61	7	7	9	11
18	20	3	3	4	4
Nr. of times	*Percentage*	*Nr. of times*	*Percentage*	*Nr. of times*	*Percentage*
10070	36.32%	-	-	-	-
14484	52.24%	-	-	-	-
12691	45.77%	-	-	-	-
10759	38.81%	-	-	-	-
2552	9.20%	-	-	-	-
4414	15.92%	-	-	-	-

Table B.8 Results CHIP data commuters.

Refuel:			**Pushback:**		**Water:**	
DurationCHIP	**CHIP**	**Real**	**CHIP**	**Real**	**CHIP**	**Real**
Rows	9863	9863	7518	7518	10828	10828
Average	18	18	8	11	4	2
Mode	15	16	10	0	4	3
Median	18	17	8	8	4	2
SD	7	6	9	10	2	1
KPI not met:	*Nr. of times*	*Percentage*	*Nr. of times*	*Percentage*	*Nr. of times*	*Percentage*
OTS	7311	74.13%	3862	51.38%	9725	89.81%
OTP	4000	40.56%	4069	54.13%	1517	14.01%
OTD	2276	23.08%	0	0.00%	414	3.82%
OTD & OTP	966	9.79%	0	0.00%	69	0.64%
OTD & OTP & OTS	966	9.79%	0	0.00%	69	0.64%
OTD & OTS	2276	23.08%	0	0.00%	414	3.82%

Table B.9 Results CHIP data commuters.

Toilet:		**Cleaning:**		**Ramp (load):**	
CHIP	**Real**	**CHIP**	**Real**	**CHIP**	**Real**
6345	6345	12415	12415	7932	7932
6	5	7	10	44	41
4	6	8	10	28	1
5	5	7	10	38	38
3	2	5	6	19	24
Nr. of times	*Percentage*	*Nr. of times*	*Percentage*	*Nr. of times*	*Percentage*
6069	95.65%	12208	98.33%	828	10.43%
2345	36.96%	8207	66.11%	2966	37.39%
759	11.96%	966	7.78%	3242	40.87%
276	4.35%	345	2.78%	1793	22.61%
276	4.35%	276	2.22%	0	0.00%
759	11.96%	897	7.22%	483	6.09%

Table B.10 Results CHIP data commuters.

Ramp (unload):		**Ramp (TurnAround)**		**Deboarding:**	
CHIP	**Real**	**CHIP**	**Real**	**CHIP**	**Real**
7794	7794	3724	3724	20277	20277
27	23	58	55	13	16
10	1	47	55	10	14
22	20	55	55	12	15
19	19	18	22	4	7
Nr. of times	*Percentage*	*Nr. of times*	*Percentage*	*Nr. of times*	*Percentage*
1035	13.27%	345	9.26%	4138	20.41%
2690	34.51%	1448	38.89%	14346	70.75%
3724	47.79%	1172	31.48%	9656	47.62%
1724	22.12%	690	18.52%	8690	42.86%
0	0.00%	0	0.00%	897	4.42%
828	10.62%	207	5.56%	1793	8.84%

Table B.11 Results CHIP data commuters.

Boarding:		**CabinCheck**		**CabinSecurityCheck**	
CHIP	**Real**	**CHIP**	**Real**	**CHIP**	**Real**
20277	20277	-	-	483	483
26	32	5	5	10	11
20	28	5	5	11	11
25	31	5	5	11	11
8	15	2	2	4	3
Nr. of times	*Percentage*	*Nr. of times*	*Percentage*	*Nr. of times*	*Percentage*
4138	20.41%	-	-	-	-
14346	70.75%	-	-	-	-
9656	47.62%	-	-	-	-
8690	42.86%	-	-	-	-
897	4.42%	-	-	-	-
1793	8.84%	-	-	-	-

Verification and Validation

Verification

The purpose of model verification is to assure that the simulation model reflects the operational process [Jerry, 2005]. A close and thorough examination of the different objects and their outputs is needed to assure the simulation model is accurate and credible [Jerry, 2005]. During the development of the model the objects as described in section 5.1.1 are individually verified with expected and realistic output. This way the result of each object is tracked and verified against expected results and the simulation model remains reliable and stable.

Validation

Naylor and Finger formulated a three-step approach to model validation that has been widely followed [Naylor and Finger, 1967]:

1. Build a model that has high face validity.
2. Validate model assumptions.
3. Compare the model input-output transformations to corresponding input-output transformations for the real system.

Step 1 - Face validity

A model that has face validity appears to be a reasonable imitation of a real-world system to people who are knowledgeable of the real world system [Carson and John, 2002]. Face validity is tested by having users and people knowledgeable with the system examine model output for reasonableness and in the process identify deficiencies [Jerry, 2005]. Together with the departments flow control, tactical planning and the CHIP team the results will be assessed and checked if they are realistic when compared to the real system.

Step 2 - Validation of model assumptions

Two categories can be distinguished: (1) structural assumptions about how the system works and (2) data assumptions.

1. *Structural assumptions* - are assumptions made about how the system is physically arranged and operates. Many structural problems in the model come from poor or incorrect assumptions [Carson and John, 2002]. The systems structure and operation should also be verified with users of the actual system [Jerry, 2005]. All assumptions are discussed with the departments (flow control and tactical planning) and described in this chapter. In combination with the departments the assumptions in this chapter are verified and validated.

2. *Data assumptions* - need to be performed in order to transfer the data from the different data sources to realistic input for the simulation model. There must be a sufficient amount of appropriate data available, and it should be verified whether they come from a reliable source to validate the model. The assumed statistical model should be tested using goodness of fit tests and other techniques [Jerry, 2005] [Sargent, 2009]. Examples of goodness of fit tests are the chi-square test and Kolmogorov-Smirnov test. Any outliers in the data should also be checked [Sargent, 2009]. Chapter 4 discussed the reliability of data sources as well as assumptions that were made when creating the input files.

Step 3 - Validating input-output transformations

For these tests the model is viewed as an input-output transformation by comparing outputs from the system to model outputs for the same set of input conditions [Sargent, 2009]. The model output that is of primary interest should be used as the measure of performance [Jerry, 2005].

For the validation only two gates and one buffer are used and a small data set where each process has the same process time. For the first flight all processes are one minute, for the second two, and so on. This way it is easy to see if entities and workers move to the right gate at the right time, and processes take as long as they are expected to take. Each worker is sent one minute before the process task has to start, and for each process there is one worker available. To verify the model, it will be run in a deterministic mode with no uncertainty to generate a resource plan. All distributions used in the model will return their expected values and all failures in the model will be disabled. This makes it easier to see if the model works as it should, before running the full scale model with stochastic distributions.

When using Task Sequences, each task may have required secondary resources and materials before processing may begin. These additional resources and materials, if not available, can cause the processing at a Server to be constrained. If resource processing is constrained by a task's constraint, the Resource Plan Gantt bar will be hatched. This should indicate to the user to open the resource and to investigate further any constraints with the task. A resource is always constrained when it is traveling towards its destination, but as long as this traveling time remains within the

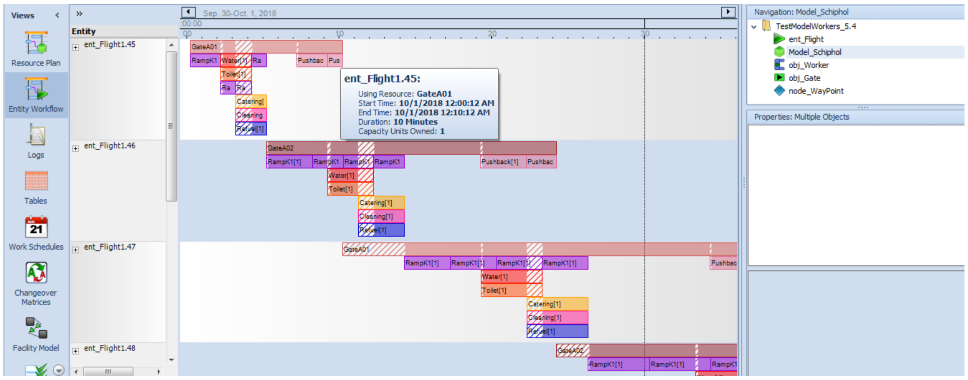

Figure C.1 Worker schedule for the test data set.

boundaries that are used to send workers before the task begins it doesn't affect the performance.

Figure C.1 shows the result for the test data set in a Gantt chart. In the results this happens at minute 3 for flight1.45. The reason for this is that the unloading and loading task are two separate tasks, which could be done by two separate teams. This means that the resources for the load task are requested while the unloading task is still busy, resulting in an overlap where the loading task is "constrained".

When looking at the flowchart (figure 3.3) the critical path shows nine processes, which would mean that the TA process for the flight1.45 should be 9*1=9 minutes (+1 minute for the first worker that is send to the gate before starting its task). Looking at the results it can be seen that the TA takes exactly the expected 9 (+1) minutes. Flight1.46 takes twice as long (9*2=18 minutes) as expected, before flight1.47 is delayed due to the fact that the ramp workers are still busy processing the previous flight. In figure C.2 the resource is opened and it can be seen that the process is constrained by the ramp worker. Also all subsequent processes are delayed with the same amount of time to minimize unnecessary waiting at the gate by successive processes. The small constraint blocks which can be seen for the other processes are due to the traveling time the workers have when requested by the gate. As long as these are within the predefined time window that is used to send workers to their location before their starting time (here 1 min), these will not cause delays for the process and do not affect the results. Table C.1 and C.2 show the results in table format.

Buffer or Gate change Tow

To see if the model also performs as expected when a tow to buffer of another gate has to be performed, two flights that needs to be towed are used as input for the model. One tow to the buffer, and one tow to another gate. The results are shown in figure C.3.

Table C.1 Results test data set verification.

	Flight1.45					Flight1.46				
	Arrive	Start	Finish	# Tasks	Duration	Arrive	Start	Finish	# Tasks	Duration
Total TA	0:12	1:12	10:12	1	9	05:13	06:13	24:13	1	18
Ramp	0:14	1:12	5:12	4	4	05:15	06:13	14:13	4	8
Toilet	2:15	3:12	4:12	1	1	09:18	10:13	12:13	1	2
Water	2:16	3:12	4:12	1	1	09:19	10:13	12:13	1	2
Catering	3:15	4:12	5:12	1	1	11:18	12:13	14:13	1	2
Cleaning	3:15	4:12	5:12	1	1	11:18	12:13	14:13	1	2
Refuel	3:13	4:12	5:12	1	1	11:16	12:13	14:13	1	2
Pushback	7:12	8:12	10:12	2	2	19:17	20:13	24:13	2	4

Table C.2 Results test data set validation.

Flight1.47					Flight1.48				
Arrive	Start	Finish	# Tasks	Duration	Arrive	Start	Finish	# Tasks	Duration
10:12	14:15	41:15	1	27	24:13	26:17	1:02:17	1	36
14:13	14:15	26:15	4	12	26:15	26:17	42:17	4	16
19:18	20:15	23:15	1	3	33:22	34:17	38:17	1	4
19:19	20:15	23:15	1	3	33:23	34:17	38:17	1	4
22:18	23:15	26:15	1	3	37:23	38:17	42:17	1	4
22:18	23:15	26:15	1	3	37:23	38:17	42:17	1	4
22:16	23:15	26:15	1	3	37:20	38:17	42:17	1	4
34:17	35:15	41:15	2	6	53:21	54:17	1:02:17	2	8

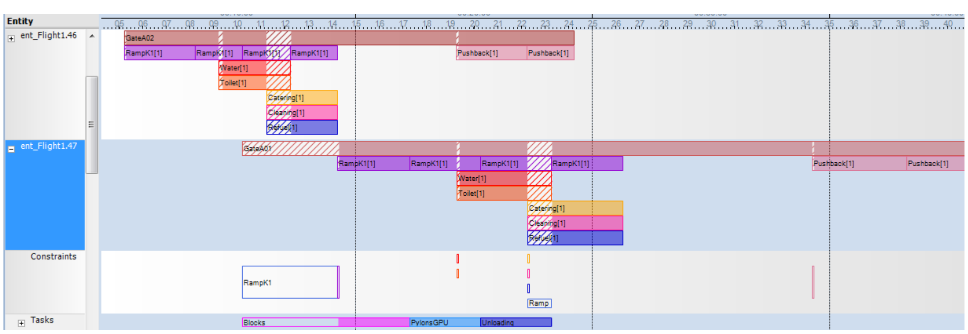

Figure C.2 Constraints

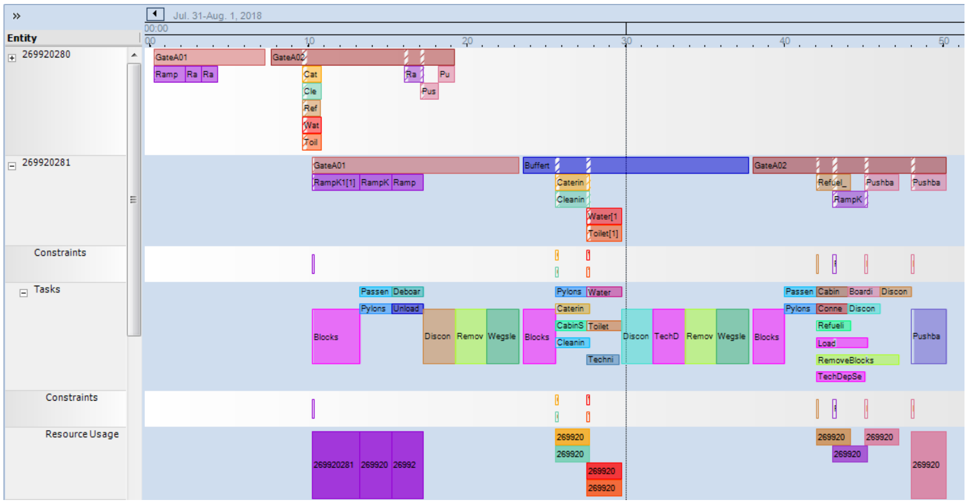

Figure C.3 Buffer and Gate change tow.

It can be seen that the right processes are performed at the right location, and are performed in the right order. All arrival processes are performed the same way as a normal turnaround, and on the buffer the catering, cleaning, water and toilet tasks are performed. These processes can all start at the same time because they do not have to wait for the deboarding of the aircraft. All departing related processes such as loading and pushback are then performed at the departing gate as expected. Refueling is included at the departing processes because the amount of fuel is determined within an hour of departure and never on the buffer.

Number of Replications

The model is stochastic resulting in variation in the output measures. It is therefore necessary to run a number of simulation replications to reduce the chance of making wrong recommendations. The number of replications is calculated based on the level of precision the outcome will have; the standard error. Compared to the sampled mean, the standard error should be relatively small for a robust statistical analysis of the model.

For models of ratio or interval data Byrne uses a unitless measure of w, the target confidence interval width. This simplifies the equation and allows intervals to be compared across measurements. In order to do this, w is defined here as "proportion of the mean [Byrne, 2013]." That, a w of 0.1 represents 10% of the mean, whatever that mean is [Byrne, 2013]. This results in the final form of the equation for determining the minimum number of simulation runs:

$$n = \left(\frac{z_{\alpha/2}}{w} CV \right)^2 \tag{D.1}$$

Where n is the number of replications rounded up to the nearest integer, $z_{\alpha/2}$ is the value of the normal distribution that represents the upper tail of the distribution set by the confidence level. (For a 95% interval, this value is 1.96.). CV is the coefficient of variation which is defined as:

$$\frac{\sigma}{\mu} \tag{D.2}$$

Formula D.1 is used to create table D.1 which shows the minimum number of model runs required to produce the desired interval with w for a given coefficient of variation (CV), assuming 95% confidence. Table values with shading have been adjusted for smaller n using the t-distribution to compute the required critical value [Byrne, 2013].

For the simulation model an α of 0.05 is chosen to achieve a 95% confidence level to be sufficient for a robust statistical comparison with other scenarios. The only stochastic values in the model are the process and travel times, which are both multiplied with a normal distribution with mean 1 and a standard deviation of 0.05 giving a value of 0.05 for the CV. For w a value of 0.05 is chosen as well and using table D.1 a minimum number of runs of 7 can be found. Jerry advices to be conservative in choosing the number of simulation runs, **10 replications** should therefore be sufficient for comparison of different scenarios.

Table D.1 Minimum number of model runs (n) to achieve desired confidence interval width (w) for model with coefficient of variation (CV), assuming 95% confidence level. Shading indicates correction for small n (see the paper from Byrne for full details [Byrne, 2013].)

		Coefficient of Variation (CV)									
		0.05	**0.1**	**0.15**	**0.2**	**0.25**	**0.3**	**0.35**	**0.4**	**0.45**	**0.5**
Width (w)	**0.005**	385	1537	3458	6147	9604	13830	18824	24587	31117	38416
	0.01	99	385	865	1537	2401	3458	4706	6147	7780	9604
	0.02	27	99	217	385	601	865	177	1537	1945	2401
	0.05	7	18	37	64	99	139	189	246	312	385
	0.1	5	7	11	18	27	37	50	64	81	99
	0.15	5	5	7	10	13	18	24	30	37	46

Fuel Flow vs Speed

Long haul flights have an average duration of around 8 hours, which is 480 minutes. To make up 15 minutes during this flight, the aircraft has to increase its speed with 100% - (480-15)/480% = 3%. Aircraft normally fly close to the green dot speed seen in figure E.1. Close to the green dot, the speed differences result in almost no increase in fuel consumption. The extra fuel consumption of an increase in speed of 3% is therefore negligible.

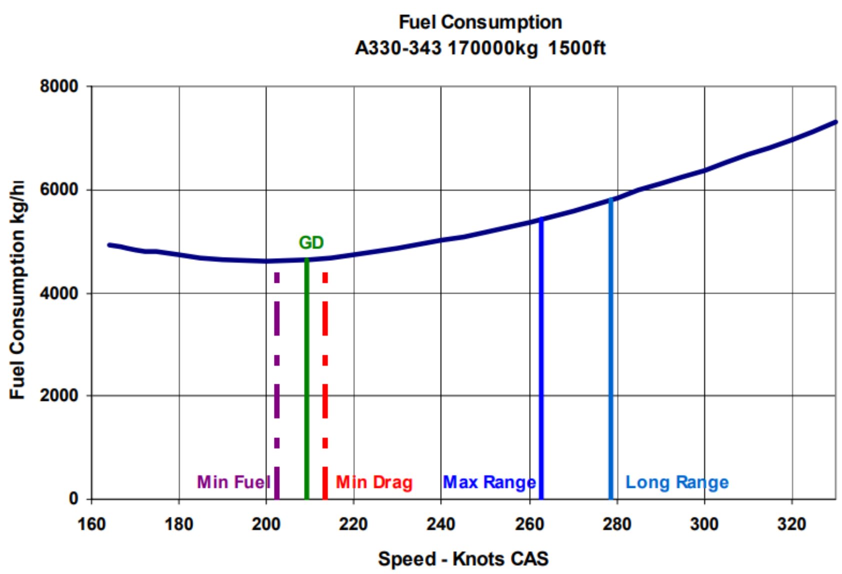

Figure E.1 Fuel flow vs speed [Airbus, 2004].

Experimental Results OptQuest Optimizer

Table F.1 Part I: Best results running the OptQuest optimizer to asses different control scenarios.

Experiment Info					**Seize Time Workers**												**Flights**		
Name	*Objective*	*Scenario*	*Priorities*	*Replications*	*Refuel Hydrant*	*Pushback*	*Catering*	*Cleaning Asito*	*Toilet*	*Water*	*RampK1*	*RampK2*	*RampK4*	*RampK5*	*Cleaning Kluh*	*Refuel Tank*	*Costs*	*Delays (TSAT)*	SPT
Costs_Under15_PRIO_CR	Costs	PRIO_CR	Under15	10	14	19	20	16	9	7	17	6	6	14	19	7	2250	8	41
Costs_Under15_PRIO-DD	Costs	PRIO-DD	Under15	10	5	14	5	11	6	23	16	5	9	7	11	6	2270	10	49
Costs_Normal_PRIO-DD	Costs	PRIO-DD	Normal	10	6	9	15	9	8	10	18	8	9	8	19	9	3335	13	65
Costs_Under15_PRIO-LST	Costs	PRIO-LST	Under15	10	6	13	8	10	7	24	17	5	13	7	13	6	3382	11	56
Costs_Normal_PRIO_CR	Costs	PRIO_CR	Normal	10	11	12	11	13	13	15	11	11	12	12	12	13	3885	14	70
Costs_Normal_PRIO-LST	Costs	PRIO-LST	Normal	10	10	12	12	13	12	14	12	10	12	12	14	14	4254	14	68

Table F.2 Part II: Best results running the OptQuest optimizer to asses different control scenarios.

On Time Start										**On Time Performance**									
Refueling	*Cleaning*	*Catering*	*Toilet*	*Water*	*Ramp K1*	*Ramp K2*	*Ramp K4*	*Ramp K5*	*Pushback*	*Refuel*	*Cleaning*	*Catering*	*Toilet*	*Water*	*Ramp K1*	*Ramp K2*	*Ramp K4*	*Ramp K5*	*Pushback*
11	8	28	100	175	1	9	1	0	5	75	29	40	2	9	47	32	58	30	0
34	14	126	207	36	1	13	0	5	6	74	28	39	2	11	46	35	55	30	0
31	34	32	127	91	0	8	1	4	83	72	30	35	2	10	48	35	57	31	0
31	13	59	166	31	0	11	0	5	6	75	28	37	3	12	47	34	53	31	0
19	10	33	53	48	0	6	1	0	11	74	29	37	2	10	47	35	54	28	0
22	10	30	58	47	0	6	0	0	12	70	29	37	2	9	46	39	54	31	0

Table F.3 Part III: Best results running the OptQuest optimizer to asses different control scenarios.

On Time Delivery										**On Time Delivery not Achieved while On Time Performance is Achieved**									
Refuel	*Catering*	*Cleaning*	*Toilet*	*Water*	*Ramp K1*	*Ramp K2*	*Ramp K4*	*Ramp K5*	*Pushback*	*Refuel*	*Catering*	*Cleaning*	*Toilet*	*Water*	*Ramp K1*	*Ramp K2*	*Ramp K4*	*Ramp K5*	*Pushback*
9	2	8	0	0	7	2	3	0	224	5	7	7	0	0	0	0	0	0	224
10	3	10	0	0	6	3	3	0	219	5	9	9	0	0	0	0	0	0	219
9	4	9	0	0	8	3	3	0	254	6	9	8	0	0	0	0	0	0	254
10	4	9	0	0	7	3	4	0	220	5	8	9	0	0	0	0	0	0	220
10	4	13	0	0	10	6	2	0	226	6	13	8	0	0	0	0	0	0	226
10	3	11	0	0	10	4	3	0	227	7	10	8	0	0	0	0	0	0	227

Table F.4 Part IV: Best results running the OptQuest optimizer to asses different control scenarios.

Efficiency										Drive Time									
Refuel	*Catering*	*Cleaning*	*Toilet*	*Water*	*Ramp K1*	*Ramp K2*	*Ramp K4*	*Ramp K5*	*Pushback*	*Refuel*	*Catering*	*Cleaning*	*Toilet*	*Water*	*Ramp K1*	*Ramp K2*	*Ramp K4*	*Ramp K5*	*Pushback*
44,9%	30,5%	26,1%	42,3%	43,0%	59,1%	42,6%	55,7%	26,3%	29,0%	9	0	0	41	118	8	1	0	0	0
40,0%	32,7%	26,1%	42,6%	43,6%	60,7%	41,3%	53,5%	25,5%	24,6%	16	2	102	149	6	21	0	4	0	0
44,1%	31,1%	26,1%	42,1%	43,0%	59,4%	42,4%	56,3%	26,0%	29,1%	6	20	1	68	42	0	0	3	77	0
46,6%	32,9%	26,2%	42,5%	43,5%	60,3%	40,4%	51,7%	27,5%	39,9%	13	3	30	112	5	20	0	3	0	0
39,1%	24,9%	26,1%	42,7%	43,8%	60,5%	40,3%	53,1%	24,9%	32,2%	2	1	6	16	18	0	0	0	4	0
40,2%	28,0%	26,1%	42,2%	43,6%	60,6%	40,5%	55,2%	24,7%	30,7%	1	1	3	18	19	0	0	0	5	0

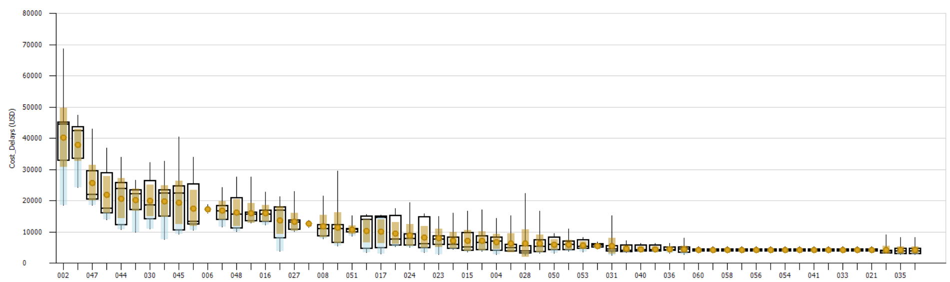

Figure F.1 OptQuest results. Priorities: Static, Control scenario: PRIO-LST

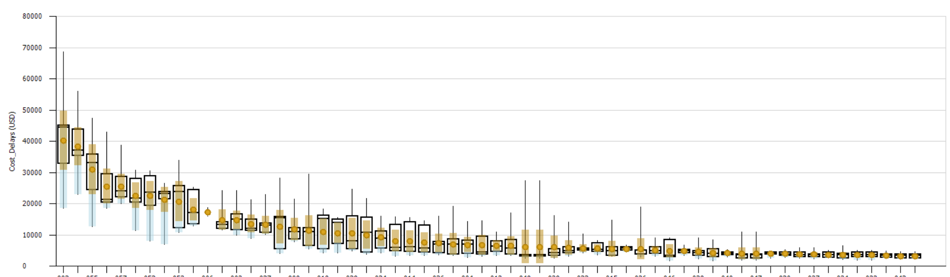

Figure F.2 OptQuest results. Priorities: Static, Control scenario: PRIO-DD

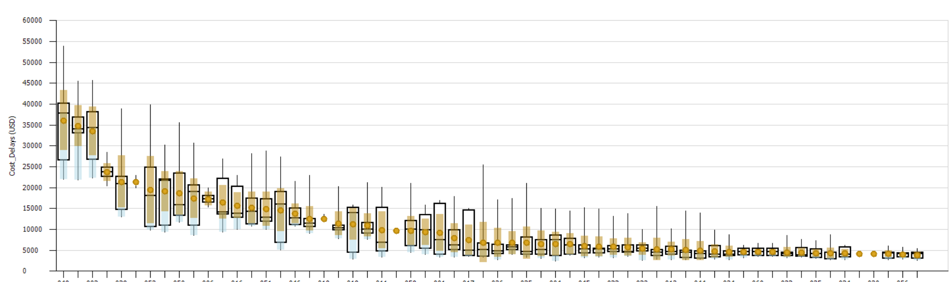

Figure F.3 OptQuest results. Priorities: Static, Control scenario: PRIO-CR

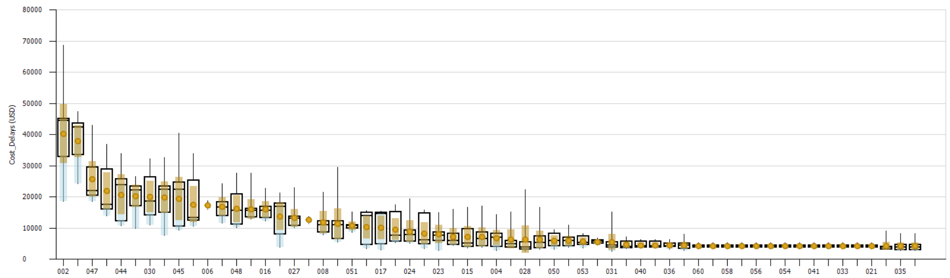

Figure F.4 OptQuest results. Priorities: Dynamic, Control scenario: PRIO-LST

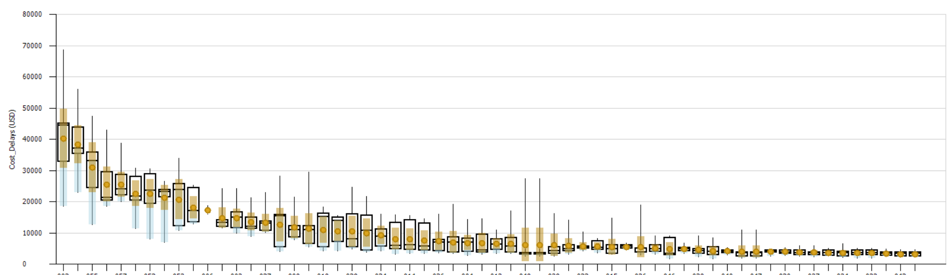

Figure F.5 OptQuest results. Priorities: Dynamic, Control scenario: PRIO-DD

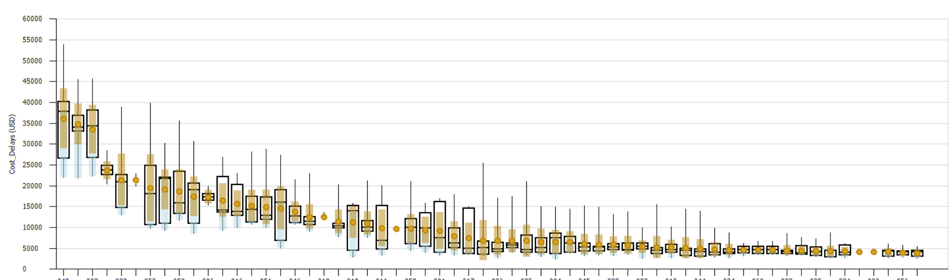

Figure F.6 OptQuest results. Priorities: Dynamic, Control scenario: PRIO-CR

Delays due to bad data input

Figure G.1 The results show the snowball effect which is the result from wrong data input for the ramp process times.

References

Airbus (2004). Getting to grips with fuel economy. `http://ansperformance.eu/references/library/airbus-fuel-economy.pdf`. Accessed: 01-2019.

airfleets.net (2018). Klm fleet details. `http://www.airfleets.net/flottecie/KLM.htm`. Accessed: 06-2018.

allaboutlean.com (2015). Visualize your system - overview of diagrams in manufacturing. `https://www.allaboutlean.com/manufacturing-system-diagrams/`. Accessed: 05-2018.

Andreatta, G., De Giovanni, L., and Monaci, M. (2014). A fast heuristic for airport ground-service equipment–and–staff allocation. *Procedia-Social and Behavioral Sciences*, 108:26–36.

Anthony, R. N. (1965). *Planning and Control Systems: A Framework for Analysis [by]*. Division of Research, Graduate School of Business Administration, Harvard University.

Bauer, K. (2004). Kpis-the metrics that drive performance management. *Information Management*, 14(9):63.

Beelaerts, Wouter W. A. Beelaerts van Blokland, S. J. d. J. (2016). Measuring lean implementation for maintenance service companies. *International Journal of Lean Six Sigma*.

Bonney, M., Zhang, Z., Head, M., Tien, C., and Barson, R. (1999). Are push and pull systems really so different? *International Journal of Production Economics*, 59(1-3):53–64.

Byrne, M. D. (2013). How many times should a stochastic model be run? an approach based on confidence intervals. In *Proceedings of the 12th International conference on cognitive modeling, Ottawa*.

Carson, I. and John, S. (2002). Verification validation: model verification and validation. In *Proceedings of the 34th conference on Winter simulation: exploring new frontiers*, pages 52–58. Winter Simulation Conference.

Chan, A. P. and Chan, A. P. (2004). Key performance indicators for measuring construction success. *Benchmarking: an international journal*, 11(2):203–221.

Damelio, R. (2011). *The basics of process mapping*. CRC Press.

dictionary.com (2018). Effectiveness. `http://www.dictionary.com/browse/effectiveness`. Accessed: 07-2018.

euro cdm (2009). Airport cdm - steps to boost efficiency. `https://www.schiphol.nl/nl/download/b2b/1492598742/6wGWW70OqQi8kEIYyUye4Q.pdf`. Accessed: 06-2018.

EUROCONTROL (2012). Airport cdm implementation. `https://www.schiphol.nl/nl/download/b2b/1492598870/5mjo4p0Onu4wQy0WskuiuQ.pdf`. Accessed: 06-2018.

Eurocontrol (2015). European airline delay cost reference values report for 2014, version 4.1. `https://www.eurocontrol.int/publications/european-airline-delay-cost-reference-values`. Accessed: 01-2019.

EUROCONTROL (2018). Airport collaborative decision making (a-cdm). `http://www.eurocontrol.int/articles/airport-collaborative-decision-making-cdm`. Accessed: 05-2018.

Fitz-Gibbon, C. T. (1990). *Performance indicators*. BERA Dialogues. ISBN: 978-1-85359-092-4.

Harrison, T. P. (2005). *The practice of supply chain management: where theory and application converge*. Springer Science & Business Media.

ICAO (2017). Air transport development. `https://www.unitingaviation.com/strategic-objective/general-interest/on-valentines-day-the-things-we-love-about-aviation/`. Accessed: 06-2018.

Ionica, A. C. and Leba, M. (2015). Gamification & research–partnership for innovation. *Procedia Economics and Finance*, 23:671–676.

Ismail, M. N. and Zin, A. M. (2008). Development of simulation model in heterogeneous network environment: Comparing the accuracy of simulation model for data transfers measurement over wide area network. *Information Technology Journal*, 7(6):897–903.

Jerry, B. (2005). *Discrete event system simulation*. Pearson Education India.

Jeyaraj, A. and Sauter, V. L. (2014). Validation of business process models using swimlane diagrams. *J Inf Technol Manag*, 25(4):27.

Jones, D. T., Hines, P., and Rich, N. (1997). Lean logistics. *International Journal of physical distribution & logistics management*, 27(3/4):153–173.

KCS (2018). Feiten en cijfers. `http://www.kcs.nl/nl/organisatie/feiten-en-cijfers/`. Accessed: 04-2018.

KLM (2016). High performance organization ground services schiphol. `https://www.consultancy.nl/nieuws/11200/klm-wil-high-performance-organization-worden`. Accessed: 04-2018.

KLM (2018a). Aircraft refuelling department. `https://www.klm.com/aircraftservices/en/refuelling/refuelling.html`. Accessed: 04-2018.

KLM (2018b). Aircraft towing department. `https://www.klm.com/aircraftservices/en/towing/towing.html`. Accessed: 04-2018.

KLM (2018c). Apron services - wpi. `https://www4.klm.com/department/groundservices/gsnet2/Afdelingen/AircraftServices/SitePages/Apron%20Services%20-%20WPI%20button%20page.aspx`. Accessed: 06-2018.

KLM (2018d). Ground operations manual schiphol. https://www4.klm.com/department/groundservices/GGHportal/KLMManuals/Schiphol/GOMS%20Downloadable/GOMS%20Downloadable.pdf. Accessed: 06-2018.

KLM (2018e). Klm apron services. https://www4.klm.com/department/groundservices/gsnet2/Afdelingen/AircraftServices/SitePages/WPI_EUR.aspx. Accessed: 04-2018.

KLM (2018f). Klm network makes schiphol into a mainport. https://www.klm.com/corporate/en/images/008372%20Factsheet%20network_tcm729-342204.pdf. Accessed: 06-2018.

KLM (2018g). Klm royal dutch airlines. https://www.klm.com/aircraftservices/. Accessed: 06-2018.

KLM (2018h). Record passenger numbers for klm in 2017. https://news.klm.com/record-passenger-numbers-for-klm-in-2017/. Accessed: 06-2018.

Klüh (2018). Klüh multiservices. https://www.klueh.de/en/Aircraft-cleaning.htm. Accessed: 04-2018.

Longman (2018). Efficiency. https://www.ldoceonline.com/dictionary/efficiency. Accessed: 07-2018.

modernanalyst (2011). What is a swimlane diagram? http://www.modernanalyst.com/Careers/InterviewQuestions/tabid/128/ID/1804/What-is-a-Swimlane-Diagram.aspx. Accessed: 05-2018.

Naylor, T. H. and Finger, J. M. (1967). Verification of computer simulation models. *Management science*, 14(2):B–92.

of Technology, M. I. (1997). System dynamics. http://web.mit.edu/sd-intro/. Accessed: 05-2018.

of Westminster, U. (2015). The cost of passenger delay to airlines in europe. http://ansperformance.eu/references/library/passengerdelaycost.pdf. Accessed: 01-2019.

OpTek (2015). The opttek documentation on optimization. http://www.opttek.com/documentation/v70engine/OptQuest%20Engine%20Documentation/Default.htm. Accessed: 01-2019.

planespotters.net (2018). Klm royal dutch airlines fleet details and history. https://www.planespotters.net/airline/KLM-Royal-Dutch-Airlines. Accessed: 06-2018.

Robinson, S. (2004). *Simulation: the practice of model development and use*. Wiley Chichester.

Roser, C. (2015a). Manufacturing structure visualization methods. https://www.allaboutlean.com/manufacturing-system-diagrams/method-structure-overview/. Accessed: 05-2018.

Roser, C. (2015b). Swim lane diagram. https://www.allaboutlean.com/manufacturing-system-diagrams/swim-lane-diagram/. Accessed: 05-2018.

Sargent, R. G. (2009). Verification and validation of simulation models. In *Simulation Conference (WSC), Proceedings of the 2009 Winter*, pages 162–176. IEEE.

Schiphol (2018a). Airport collaborative decision making (a-cdm) operations manual.

Schiphol, A. A. (2018b). Map of amsterdam airport schiphol. https://bin.snmmd.nl/m/m1nycdawjiym.gif. Accessed: 04-2018.

Sedee, M. (2018). Schiphol net onder maximum aantal vliegbewegingen. https://www.nrc.nl/nieuws/2018/01/08/schiphol-net-onder-maximum-aantal-vliegbewegingen-a1587513. Accessed: 06-2018.

Services, K. G. (2018). Klm apron services. https://www4.klm.com/department/groundservices/gsnet2/organogrammen/Apron%20Services.pdf. Accessed: 04-2018.

Simio (2017). Simio manual. https://www.simio.com/. Accessed: 08-2018.

Simio (2019). Simio reference guide. http://logsis.com.ar/wp-content/uploads/Archivos/Biblioteca_Recursos/Simio/Simio%20Reference%20Guide.pdf. Accessed: 01-2019.

Thierry, C., Bel, G., and Thomas, A. (2008). Supply chain management simulation: an overview.

Toledo, W. (2018). Push-pull strategy. https://en.wikipedia.org/wiki/Push%E2%80%93pull_strategy. Accessed: 11-2018.

van Blokland, W. B. (2018). Generic research process for thesis project. Accessed: 05-2018.

Veeke, H. P., Ottjes, J. A., and Lodewijks, G. (2008). *The Delft systems approach: Analysis and design of industrial systems*. Springer Science & Business Media.

Vos, P. (2017). Optimizing the use of autonomous surface vehicles in the port of rotterdam: The dial-a-ride problem. Accessed: 06-2018.

wikipedia.org (2018). Efficiency. https://en.wikipedia.org/wiki/Efficiency. Accessed: 07-2018.

Yuvaraj, V. R. and Zhang, S. (2013). Reducing wip inventory of production line in aq segerström & svensson ab.