UNIVERSITY OF TWENTE.

FACULTY OF BEHAVIOURAL, MANAGEMENT & SOCIAL SCIENCES

INDUSTRIAL ENGINEERING & MANAGEMENT

BACHELOR THESIS

IMPROVING THE SURGERY PROCESS OF THE DAY CARE DEPARTMENT AT ERASMUS MEDICAL CENTRE

BY

STEFAN DANIËL SCHRIJER – UNIVERSITY OF TWENTE

ADVISORY COMMITTEE

PROF. DR. IR. E.W. HANS – UNIVERSITY OF TWENTE

DR. IR. A.G. MAAN-LEEFTINK - UNIVERSITY OF TWENTE

DRS. SYLVIA VAN DER VEEN – ERASMUS MC

DECEMBER 2020

Erasmus MC
Universitair Medisch Centrum Rotterdam

UNIVERSITY OF TWENTE.

PREFACE

The report in front of you concludes my bachelor assignment which I conducted at Erasmus Medical Centre in Rotterdam. It also concludes my bachelor study Industrial Engineering & Management at the University of Twente. It is the result of several months of hard work in which I experienced how it is to work at a large organization consisting of several departments with their own interests. I performed this research during the coronavirus pandemic of 2020 what resulted in an unusual internship period. I am grateful that my assignment could take place during a period where intensive care capacities were overloaded and the regular healthcare was scaled down in order to catch the amount of infections. It was an honour to work together with staff that advised the Dutch government on taking corona measures and staff that worked day and night to fight the virus and to treat patients.

Therefore, I want to thank the Project Team Integral Capacity Management to make it possible to complete this assignment during this unusual period. I want to express my gratitude especially towards Sylvia van der Veen for her supervision during my internship period. Her critical vision and honest opinion helped me to achieve a next stage in my personal development. I also want to thank Cobus van Wyk and Ruben Goedhart for their substantive guidance. Next, I want to thank my first supervisor from the University of Twente, Erwin Hans, for his inspirational conversations and his inexhaustible positivism that pulled me through my writer's block. I hope you will keep teaching first year students VBA programming, that is the first time you inspired me as a student and I know I am not the only one. I also want to thank my parents and my sister Nora for the encouragement and the possibility to stay at their place in The Hague during my internship period, my friends from my student home Patatras for their support, and finally my study advisor Cornelis ten Napel for his help during my bachelor studies. Thanks to his ability to see potential I was able to finish my study in such a successful way.

Enschede, December 2020

Stef Schrijer

MANAGEMENT SUMMARY

Background & Motivation

Erasmus MC is a large tertiary hospital in The Netherlands that delivers mainly specialized care. One of the departments of the hospital is the day care department. This department is used for treatments that do not require an overnight stay at the hospital. Day care surgeries are the main type of treatments that are performed via the day care department. When patients undergo surgery they have to be transported from the day care department on the fourth floor to the OR-department on the sixth floor and delivered at the holding at the Middle wing. Currently the hospital experiences that day care surgeries often start too late. Surgeries starting too late leads to overtime, patient cancellations and non-occupied ORs. Since ORs are one of the most expensive assets of a hospital, the late start of surgeries has a negative impact on the performance of the hospital. Therefore, this research suggests a solution to decrease the number of day care surgeries starting too late.

Approach

After a practical experience at the day care department of the hospital we assumed that the amount of surgeries that start too late is the result of the distribution of patient transportations throughout the day which day care nurses cannot handle. In order to confirm this finding we investigated the current process of day care surgery. First, the current situation was analysed by making a flowchart diagram and a problem cluster. We concluded that the split-up of the holding and recovery at the OR-department resulted in increased pick-up time for day care patients since nurses have to walk through the OR-department and need to put an OR-suit on to pick up patients from the recovery at the South wing. This also increased the number of transport movements around the elevator on the Middle wing of the OR-department resulting in more congestion throughout the day. Secondly, the number of surgeries starting throughout the day was analysed. This made clear that there is a large peak of day care surgeries that are planned to start at 08:00, but often these surgeries start too late. Throughout the rest of the day repetitive peak moments of day care surgeries that start in the same timeslot were observed. Thirdly, the surgery properties are analysed by means of a case-mix classification based on the coefficient of variation and average occupancy of an OR-session per surgery type. This made clear that day care surgeries do not deviate much from their expected surgery duration and occupy relatively small parts of OR-sessions. Clinical surgeries generally deviate more from their expected value and occupy larger parts of OR-sessions. Lastly, the surgery data was analysed to investigate the current planning method. The analysis was performed by making use of an algorithm that produced per specialty the three most occurring sequences of admission types in an OR-session. This analysis was also performed for the three most occurring sequences based on surgery length. Such a sequence represents the order in which surgeries take place in an OR-session. Since the OR-department did not have a clear vision of their current planning method the analysis helped to point out opportunities for improvement and served as a starting point for recommendations for the planning methods of the OR-department. Together with the analysis of the surgery process of day care patients we concluded that the transportation moments of day care patients should be better spread throughout the day.

Problem-Solving Approach

The first possible solution of the problem originated from the hospital staff. Currently there are two ORs next to the day care department. Patients can be delivered much faster to these ORs than to the ones on the sixth floor. Recently, the Project Team ICM came up with a plan to add two new ORs to the ORs on the fourth floor. However, this is a long-term plan that will require a large investment. The second intervention is an adjustment of the current planning method. We introduce the concept of a “differentiated start” where not all surgeries start at the same time to spread the number of surgeries more throughout the day. Another intervention would be to take a look at the recovery time of patients to also predict the pick-up moments of patients and get a precise picture of the amount of patient transports per timeslot. In this way one can try to find a planning method that spreads the transportation as evenly as possible. We leave this intervention for further research and focus on the intervention of a differentiated start. Since such a decision can have several consequences for the OR-department we decided to test this solution and create a Monte-Carlo simulation to perform experiments with different starting times. We created a simulation model including a dashboard to manage the simulations. The simulation model can be used as a powerful management tool by the hospital and as a starting point for future research.

Results & Conclusion

Different experiments regarding the starting times of the ORs are performed and demonstrate that applying the differentiated start on ORs of specialties with a considerable amount of day care surgeries can already have a large impact on the system. Based on the experiments we recommend to adjust the starting times of the ophthalmology (OOG), ENT (KNO), vascular surgery (VAT) and pain surgeries (PYN). In the first place these adjustments result in a smaller peak of day care surgeries starting at the beginning of the day and less specialties with day care surgeries starting at the same time resulting in a less repetitive pattern of surgery starting times. In practice this could lead to a better spread of patient transport throughout the day.

Recommendations

For future research we recommend in the first place to try to better approximate the surgery distributions per specialty, Secondly, we recommend to investigate the recovery duration per surgery type in order to get a more detailed view on the number of transport moments. This can be used to even better spread the number of transport moments. It is possible to schedule more transport moments between 13:00-15:00, since the morning shift and afternoon shift for day care nurses are both present at this time. The last recommendation would be to investigate the optimal division of day care patients between the ORs on the fourth and sixth floor if the two extra ORs are added on the fourth floor to still keep the occupancy of the ORs on the fourth floor high enough.

TABLE OF CONTENTS

Definitions & Abbreviations

Definitions

- **Day care centre/Day care department/Day care unit:** Department of the hospital that is used for small surgeries that do not require an overnight stay at the hospital. The Dutch translation is "dagbehandelingscentrum".
- **Day care lounge:** General name for the wards and personal rooms of the day care unit.
- **The management:** The managers that represent the stakeholders at the various departments of Erasmus Medical Centre. These departments include the day care unit, holding, recovery and OT-department.
- **Project Team ICM/Capacity management:** The Project Team Integral Capacity Management is an advisory body for the Management Board of the Erasmus Medical Centre. The team is integrated within various departments of the hospital to enhance the decision-making on capacity issues. This research is held within the Project Team ICM.
- **Holding:** Special ward where patients are prepared for surgery.
- **Recovery:** Special ward where patients wake up from surgery and are monitored until they are ready to return to the day care department or another department.
- **HiX:** Healthcare Information X-change. Information system used by Erasmus Medical Centre for the exchange and storage of patient data.

Abbreviations

- **OT:** Operating Theatre/Operating Room
- **ICM:** Integral Capacity Management
- **PACU:** Post Anesthesia Care Unit, special department that intensively monitors patients returning from surgery, intensive recovery
- **COW:** Computer On Wheels, transportable computer with access to HiX, mostly used for intakes at the day care department

This research is conducted as part of the Project Team Integral Capacity Management at Erasmus Medical Centre in Rotterdam, The Netherlands. By making use of a simulation study we visualize the effects of an alternative planning strategy for surgeries in the day care unit. The goal of the alternative planning strategy is to minimize the number of patients that arrive too late at the holding from the day care unit, as these cause a delay in surgery processes.

CHAPTER 1: INTRODUCTION

This chapter explains the context of the experienced problem. Section 1.1 explains the research motive and introduces the organization. Section 1.2 elaborates on the scope of the research by placing the research within the framework for healthcare planning and control. Section 1.3 states the goal of the research and section 1.4 introduces the problem cluster and explains it step by step. This chapter ends with the research questions being explained in section 1.5.

1.1 Background Information

This section explains the context of the experienced problem. This is done by explaining the research motive in section 1.1.1 and by introducing the organization in section 1.1.2, the management team where this research is performed in section 1.1.3, the day care unit in section 1.1.4 and the OR-department in section 1.1.5.

1.1.1 Research Motive

At the beginning of 2020, the OR-department of Erasmus MC decided to make some changes to the layout of their recovery and holding. Initially the OR-department had two wards functioning as a combined holding/recovery. Since this resulted in ambiguity for the staff at both locations, the management of the OR-department decided to make one recovery and one holding (with only recovery for eye surgery patients). This resulted in a different walking route for the day care staff since they had to pick up patients at the other end of the OR-department. When patients were picked up, they had to be transported through the waiting area of the outpatient clinic since patients could not simply be transported through the OR-department because of safety reasons. Because of this change in transport route the pick-up time of patients increased, and day care nurses were longer absent from the day care unit. The day care nurses indicated that the workload had clearly increased, the transportation route was patient unfriendly, and patients were delivered later at the holding throughout the day since it took more time to pick up patients. Therefore, the day care unit asked for an analysis of the walking route for the surgery patients of the day care unit and to analyse possible solutions. However, the management of the OR-department and the day care unit came up rather quickly with two possible solutions. These solutions were tried out by means of two pilots and the management of the OR-department and the day care unit decided that one of the solutions was the only workable solution. Patients could now be picked up at the recovery while walking through the OR-department, but the day care staff had to wear a special OR-suit over their nurse clothes. This suit must be taken off every time the nurse leaves the OR-department and is disposed. The process had still some inefficiencies because of the multiple changing times of clothes, but the process was rated efficient enough to hold on to. However, the OR-department and day care staff still experienced the problem of late delivery of day care patients to the OR-department. Therefore the capacity management decided to perform an analysis of the surgery process at the day care unit. This scientific report contains the results of that analysis.

1.1.2 Erasmus MC

This research is performed at Erasmus Medical Centre in Rotterdam. Erasmus MC is a tertiary academical hospital connected to the Erasmus University of Rotterdam. Erasmus MC is currently one of the largest hospitals in The Netherlands. The hospital has a capacity of around 1.300 beds and more than 11.000 employees. Sophia Kinderziekenhuis (children's hospital) and Erasmus MC Kanker Instituut (cancer institute) are also part of Erasmus MC. The hospital focuses mainly on specialised care. This means that patients from a wide area outside Rotterdam can come to the hospital or are referred to the hospital in order to get a specialised treatment. The hospital is divided in nine themes. This research is carried out within the theme "Spoed, Perioperatief en Intensief" (abbreviated SPIN) which stands for urgent, perioperative and intensive. The OR-department (acute care) falls within the theme of SPIN and the day care department is considered as part of the OR-department.

1.1.3 Project Team ICM

This research is performed as part of the Project Team Integral Capacity Management (ICM). This project team started in 2019 to coordinate the capacities inside the hospital's patient chain. Some examples of problems the project team is concerned with include operating room planning and bed distribution among the various departments. Decision-making is supported by performing quantitative analyses. This is mainly done by data analysts making use of all the data stored in the information system of the hospital: HiX.

1.1.4 Day Care Unit

The day care unit is a department of Erasmus MC that focuses on short treatments. The rule of thumb for a stay at the day care unit is that patients should not stay longer than one day at the hospital. The treatments that take place are small surgeries where patients can recover at home. The day care unit is run by several nurses that are divided among the various lounges. The unit consists of single lounges and 6-person lounges. Depending on the type of surgery, the patient can be placed in a bed or in a chair. Chairs are used since some specializations prefer doing surgery while a patient sits in a chair (eye surgery for example). Several treatments can be distinguished at the day care unit: small surgeries, infusion therapy, pre- and after-care for radiological research, nebulization therapy, function tests, outpatient procedures and Mohs surgery. This research is focused on procedures of the day care unit that require the use of an OR. From now on we will call this type of treatment "outpatient surgery." Currently, the day care unit has four ORs for their own use. Two ORs are exclusively used by dermatology and the other two are also used for other types of surgery. Next to the ORs on their own floor the day care unit makes also use of the ORs at the OR-department on the sixth floor. The OR-department has 26 ORs at their disposal of which some can be used by the day care unit.

1.1.5 OR-Department

When patients arrive at Erasmus MC for day surgery, they first come to the day care unit such that nurses can prepare the patient and he or she can wait to be transported to the operating theatre. There are 2 locations where day care patients can undergo surgery: at the OR-department of the day care unit or at the OR-department on the 6^{th} floor. The two ORs at the day care unit (NDC & NDD) are of a lower degree. Therefore, not every kind of surgery can be performed there. At the OR-department on the 6^{th} there are 22 ORs (N01, N02, ... , N21 and N26) available to plan day care surgery. However, N01 is always reserved for emergency patients. In order to plan patients, Erasmus MC makes use of a blueprint. There are 2 blueprints: one for the odd weeks and one for the even weeks. The blueprint is almost never changed and only has a couple of small adjustments per year. Appendix A shows in figure 22 an example of such a blueprint. As can be seen in the blueprint some ORs have special equipment available such as a laser or Da Vinci robot, but we leave this out of the scope of this research. The OR-planning has to be finished a week before the surgery takes place on Monday 12:00. Therefore we assume that the ORs are planned per week. The ORs that are used for day care surgery are also used for surgeries with different admission types. Table 1 shows the percentage per admission type of the selected ORs.

Admission Type	Percentage of surgeries
Intensive Care	3 %
Clinic	47 %
Day Care	33%

Other	18 %

Table 1: Percentage of surgeries per admission type (n=28234; T=2019; source: HiX)

Next to admission types we also distinguish between elective patients and emergency patients. Erasmus MC also makes a distinction between different emergency patients. Table 2 shows the emergency types and the time within which the surgery needs to start.

Emergency type	**Surgery needs to start in**
A - Acute	< 30 minutes
A1 - Acute	< 2 hours
SA - Urgent	< 6 hours
S - Urgent	< 24 hours
SE – Semi-Elective	< 72 hours

Table 2: Emergency types

1.2 Scope

This section describes in what perspective the research is placed within the different managerial areas of healthcare. Section 1.2.1 introduces the framework for healthcare planning and control (Hans, Van Houdenhoven, & Hulshof, 2012) and section 1.2.2 places this research within a specific part of the framework.

1.2.1 Framework

This research is conducted within a large organization consisting of numerous employees, departments and management layers. Within large organizations it can be difficult to coordinate the different involved parties. Especially in large tertiary hospitals such as Erasmus MC it can be difficult to coordinate the different departments and align their patient streams. Because of the complexity of human care within a hospital we often see different departments and professions with their own objectives that can conflict with other departments. Especially an OR-department can conflict with other departments since patients of different departments have to undergo surgery. Furthermore, the OR-department is one of the most expensive departments of the hospital because of the complex equipment used and the specialized staff that is available. Therefore, the objective is to optimally use the OR-department and at the same time adjust it to other departments. Since this research is concerned with the planning of ORs it is relevant to get a clear view of the different management layers involved. In doing so we use the Framework for Healthcare Planning and Control (figure 1) that was proposed by Hans et al. in 2012 (Hans, Houdenhoven, & Hulshof, 2012). The framework distinguishes hierarchal levels and management areas and is considered universally applicable in healthcare organizations.

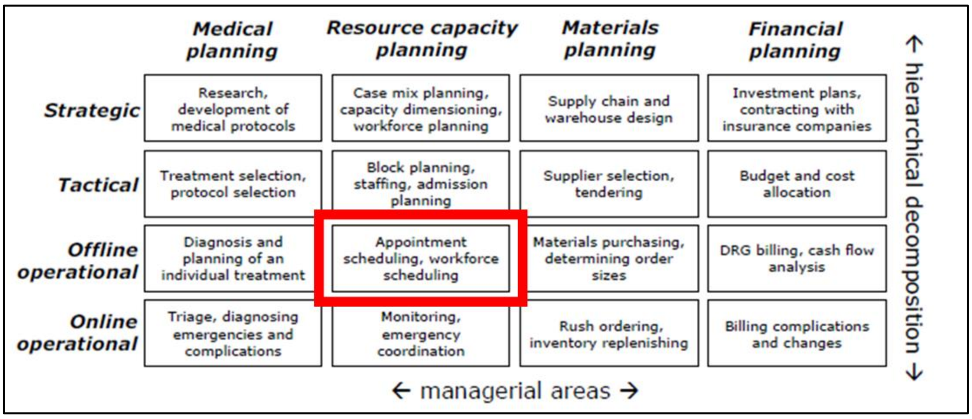


Figure 1: Framework for Healthcare Planning and Control (Hans, Houdenhoven, & Hulshof, 2012)

1.2.2 Positioning within Framework

This research focuses on the management area of resource capacity planning since we want to make optimal use of the OR-department. Within this area we distinguish four hierarchical layers. At the strategic level the capacity dimensioning of the OR-department is determined, and the capacity is divided among different specializations. At the tactical level we look at a shorter time horizon and blocks of the OR are assigned to a specific surgeon and surgical staff is planned. At the operational level the processes that are determined at the higher levels are executed. The offline operational level involves short-term decision making. This is where elective patients are scheduled in advance and the workforce is scheduled for specific ORs. The online operational level consists of day-to-day monitoring of the ORs, last-minute changes in the schedule for the day care unit and the coordination of emergency patients. This research focuses on the offline operational level of the OR-department and the day care unit. We investigate the possibilities of changing the starting times of the surgeries for the day care unit to maximize the utilization of the day care unit and maximize the number of patients that arrive on time at the holding of the OR-department.

1.3 Research Goal

As discussed in chapter 1.1.1 the OR-department is currently dealing with ORs starting too late. The problem this research aims to tackle is the late delivery of patients at the holding of the ORs. Since there is a suspicion that delays are caused by the day care unit of the hospital, we first want to examine if these delays are caused by the day care unit and in what amount. If this hypothesis can be assumed, we want to investigate how we can minimize the delays by looking at alternative planning methods. Therefore, we state the following research goal:

To develop and prospectively assess interventions to reduce late patient arrivals to the holding.

1.4 Problem Cluster

This section describes the problem more into detail. Section 1.4.1 states the management problem that lead to this research and comes up with a problem cluster to describe the problem context of the management problem. Sections 1.4.2 up to 1.4.5 give an explanation of the problem cluster, where 1.4.5 describes the core problem of the research.

1.4.1 Management Problem

analysing the process of outpatient surgery and by conducting semi-structured interviews with day care nurses, we came up with a problem cluster. Figure 2 shows the visual representation of this problem cluster. The problem cluster is made to identify all the problems that occur during the process of outpatient surgery and lead to the following stated management problem:

"Patients are delivered later than planned at the holding of the OR-department."

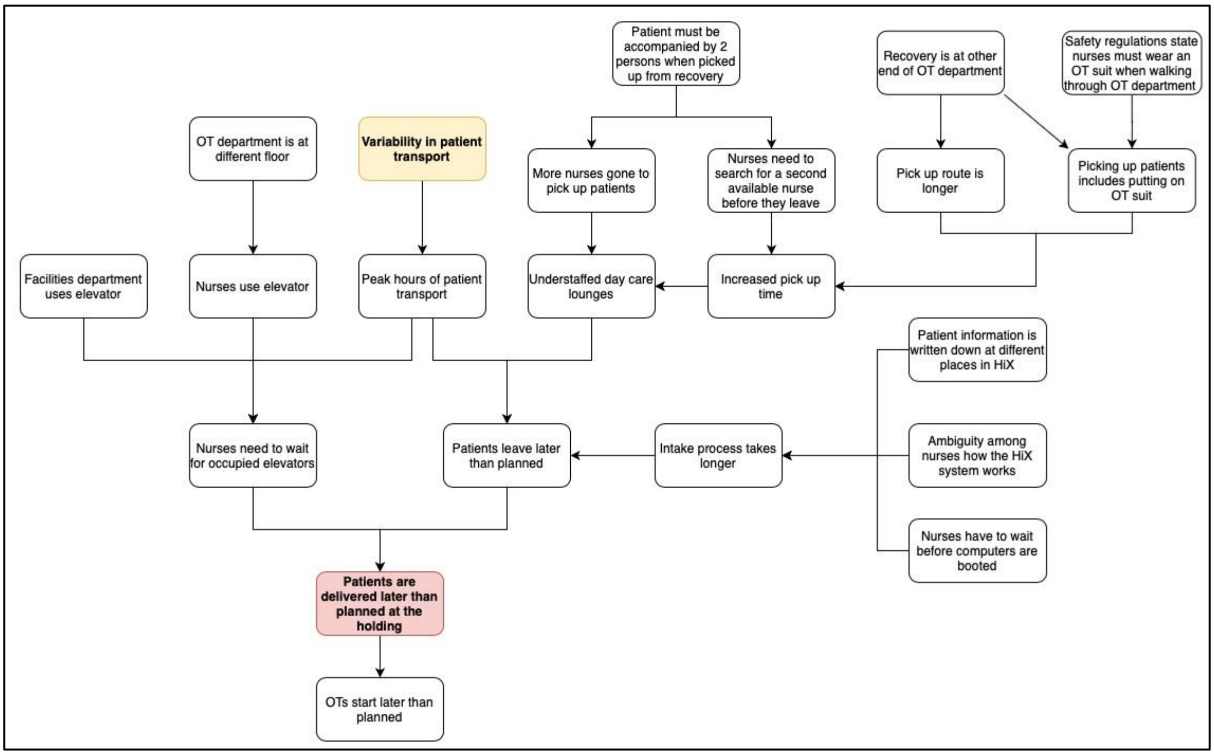


Figure 2: Problem Cluster

The cluster was made in consultation with the management of the day care unit, management of the OR-department and one of the members of the ICM that coordinates the planners of the OR-department. The principle of making a problem cluster originates from the Managerial Problem-Solving Method (Heerkens & Van Winden, 2012). The MPSM suggests analysing the problem context by making a problem cluster and then select a core problem based on several criteria. In the following subsections we will first elaborate on the different sections of the problem cluster and we end this section by explaining our choice for the selected core problem, which leads to the stated management problem.

1.4.2 Intake Process Delays

While we were observing the process of outpatient surgery, one of the first problems we encountered was during the intake process. Patients are picked up from the waiting area of the day care unit and brought to their lounge. The patient is prepared for surgery and the day care nurse confirms personal information with the patient, the nurse also checks if there are any specialties to consider. A problem that occurs during the intake of the first patients is that the nurses must wait until the Computer on Wheels (CoW) and other computers have to boot. The day care nurses indicated that this takes a considerable amount of time and leads to the first delays. The second problem indicated by the day care nurses is that patient information is written down at different places in HiX. Before patients undergo surgery, they must go through a pre-operative screening. Since the HiX interface has different possibilities and places to write down patient information it can differ where patient information is written down. Therefore, it can take extra time for the day care nurses to find the right patient information. Because of this inconvenience day care nurses sometimes even miss valuable information such as pre-operative medication. Another problem that increases the time of the intake process is ambiguity on how the HiX system works. Some nurses do not encounter any difficulties when using HiX, while other nurses find it too difficult to use the system. For these nurses it takes more time to do an intake with a patient. These three problems result in a longer intake process than planned, which leads to patients being transported later than planned to the holding.

1.4.3 Understaffed Day Care Lounges

The second section in the problem cluster concerns understaffed day care lounges. We can divide this problem in several other core problems. The first problem occurs when patients must be picked up from recovery. If a patient has recovered from surgery the day care staff is called to pick them up at the recovery. Since patients must be accompanied by two persons when picked up from recovery this means that in most of the times two nurses are absent from the day care lounge, hence there are less nurses to transport new patients to the holding. Since during peak hours it can be rather hectic at the day care unit and it can be the case that there are not enough nurses, it can take more time to find a second nurse to pick up a patient. Together with the result of the split up of the holding and recovery this leads to a longer pick up time. In the previous situation patients could be picked up and delivered at the same holding/recovery near elevator

Middle, but now the patients must be picked up at the recovery at the other end of the OR-department. Since safety regulations concerning contamination risk obliques the nurses to wear an OR suit over their normal clothes this takes extra time for nurses to put on every time they enter and take of when they leave the OR-department. Since the suits can easily tear when put on or taken off, they are often thrown away after a single use. On the long term this will lead to a great expense for the OR-department as well.

1.4.4 Variability in Patient Transport

The last section in the problem cluster is concerned with variability in patient transport. As earlier mentioned, we experienced periods where more patients as possible had to be transported by the day care nurses and periods of time where no patients had to be transported at all. This variability in patient transport leads to peak hours for patient transport. This holds for picking up patients from the recovery as well as delivering patients to the holding. Nurses indicate that they must wait for occupied elevators since patients had to be transported from and to the OR-department on a different floor. At the same time the elevators are also used by the facilities department. It turns out that it is difficult for day care nurses to deliver all the patients in time during peak hours. Even though planners try to take the availability of nurses into account, it happens that more patients at the same time must be transported since surgeries take longer or shorter than planned or patients have to be picked up from recovery while at the same time patients have to be delivered to the holding.

1.4.5 Core Problem

Now that we have explained the context of the experienced problem by means of a problem cluster a core problem can be selected. This is done by going "back" in the problem cluster to find the first problem that can be solved and has no external causes (Heerkens & Van Winden, 2012). In addition to this requirement we want to focus in this research on a problem that has as much impact as possible on the experienced management problem. In the first place we see that nurses experience difficulties during the intake process. These can be best described as deficiencies in ICT. Since we expect that solving these problems could be rather time consuming because of the varied nature of the problems but a small yield in amount of time in the intake process we do not select this as a core problem. Secondly, the day care staff experiences understaffed day care lounges throughout the day. As we concluded this is mainly caused by the fact that patients must be accompanied by two nurses when picked up from the recovery and because the pick-up process takes longer. Since these causes could be rather difficult to influence since the transportation route was recently adjusted by the management of the day care unit and OR-department and the new route came out as the best option for the involved departments, we decide to not adjust these causes. However, if the pick-up moments could be scheduled more efficiently it could be possible to experience less problems. Since the delivery times cohere with the pick-up times, we decide to focus on the variability in patient transport because of the way day care surgeries are scheduled. Therefore, we chose variability in patient transport as our core problem.

1.5 Research Questions

To structure the steps in this research we come up with research questions to find a solution to the stated core problem. In the following chapters we will answer these research questions and come up with a conclusion.

1. What does the process of day care surgeries look like? (Ch. 3)
    1.1. What effect had the split-up of holding and recovery on the day care unit? (Ch. 3.1)
    1.2. How are the surgery starting times distributed throughout the day? (Ch. 3.2)
    1.3. How can we classify the surgeries at Erasmus MC? (Ch. 3.3)
    1.4. How are surgeries currently planned? (Ch. 3.4)
2. What are possible solutions for the experienced problem? (Ch. 4)
    2.1. What are the current solutions proposed by the hospital staff? (Ch 4.1)
    2.2. What are other suggestions to solve the problem? (Ch. 4.2-4.3)
    2.3. How can we imitate the OR-department to test the solutions? (Ch. 4.4)
3. How can we use a Monte-Carlo simulation model to test an intervention? (Ch. 5)
    3.1. What is a Monte-Carlo simulation? (Ch. 5.1)
    3.2. How does the simulation model work? (Ch. 5.2-5.6)
    3.3. How can we perform experiments? (Ch. 6.1-6.3)
    3.4. What conclusions can we draw from the experiments? (Ch. 6.4)
4. What interventions can be recommended to the hospital? (Ch. 7.1)
5. What are possibilities for future research? (Ch. 7.2)

CHAPTER 2: ANALYSIS OF CURRENT SITUATION

This chapter analyses the current situation at Erasmus MC. Section 2.1 describes the findings from a practical experience at the day care department. Section 2.2 explains the consequences of the split-up of the holding and recovery. Section 2.3 analyses the current situation of the surgery process by means of an extensive data analysis. It is examined whether ORs indeed often start with day care surgeries and how often day care patients arrive too late at the holding of the OR-department. Section 2.4 classifies the surgeries of the hospital by means of a case-mix evaluation. Section 2.5 explains how the characteristics of day care surgeries can be beneficial and explains why these characteristics are currently working in a disadvantageous way. This chapter ends with section 2.6 analysing the planning method by describing the current method that is used and by analysing data to give an indication of the planning method. We use a dataset containing all the surgeries of the selected ORs of 2019 since in the first half of 2020 the COVID-19 virus arrived in the Netherlands. This resulted in unreliable data of the OR-department since the hospital had to deal with an exceptionally high number of intensive care admissions. This resulted in an unusual occupancy of the OR-facilities.

2.1 Observations from Practice

In order to get a first impression of the situation at the day care department we observed the process of day care surgery in person. We participated actively in the pick up and delivery of patients to experience the process. During the observations it became clear that there were peak moments with high transport activity. During these peak moments, patients had to be picked up from the recovery and other patients had to be brought to the holding. This often leaded to unmanned day care lounges and late patient deliveries. However, in between the peak moments there were long periods where no patients had to be transported and nurses were waiting until the next peak. The first peak was indeed at the beginning of the day and this confirmed the assumption of the OR-department that the outpatient surgeries were often starting too late. Our first thought on this problem was that we had to spread the transportation moments of the patients more evenly throughout the day. To ensure that this was the right solution to our problem we analysed the process of outpatient surgery. Appendix A shows in figure 23 a visual representation of this process by means of a flowchart diagram. This flowchart is made from the patient's perspective.

2.2 Consequences of the Split-Up

In the first place we want to know what the split-up of the holding and recovery did for the day care department. Figure 3 shows the layout of the OR-department before the split-up and figure 4 shows the layout after the split-up of the holding and recovery. As can be derived from the figure on the right-hand figure the recovery is only accessible for day care nurses by walking through the OR-department (light blue lines).

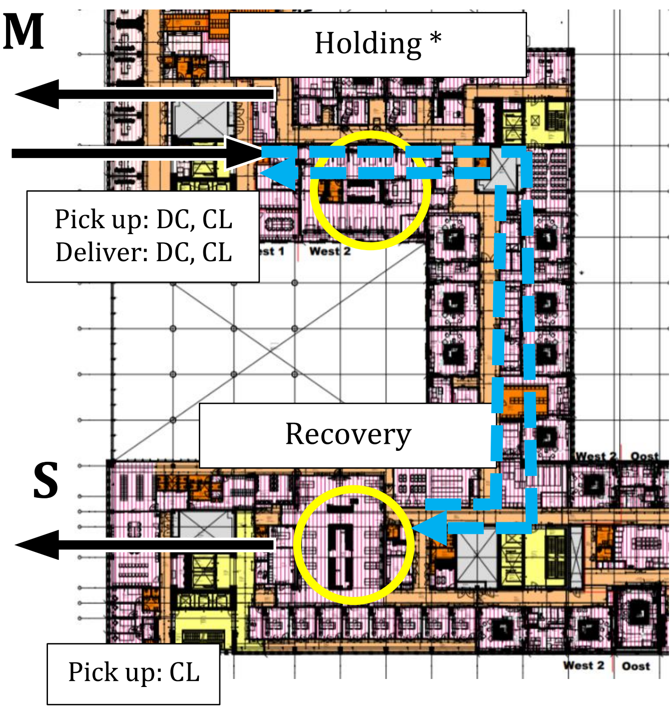


Figure 6: OR-department after split-up

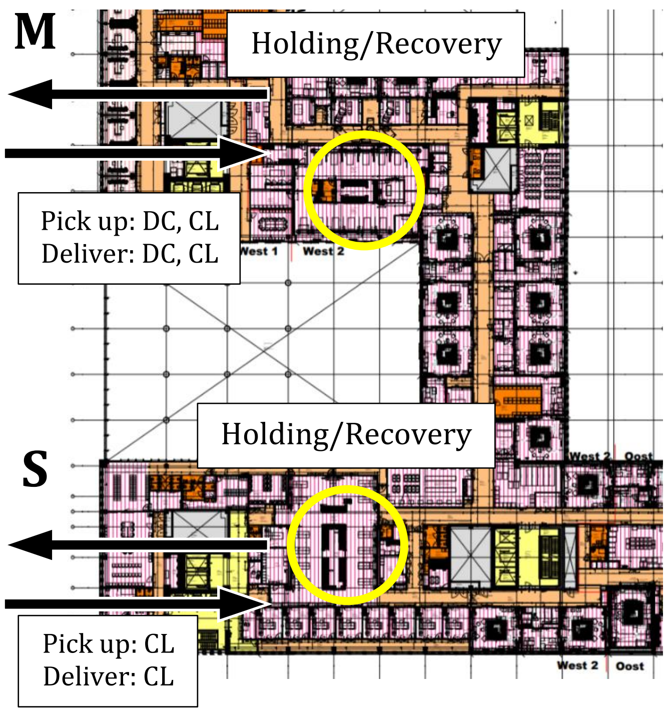


Figure 5: OR-department before split-up

For the day care nurses there is no possibility to pick up their patients via the elevator on the South wing of the sixth floor since the day care department is situated at the Middle wing (indicated with a large "M") of the fourth floor and the intensive care on the South wing (indicated with a large "S") of the fourth floor. Therefore, there is no possibility for day care nurses to walk on the fourth floor from the South elevator shaft to the day care unit and do the day care nurses have to walk through the OR-department wearing a special OR-suit. This also means that day care nurses only make use of the elevator shaft on the Middle wing. In the new situation all patients must be delivered at the holding on the Middle wing. So where in the previous situation clinical patients could also be delivered at the South wing making use of the Southern elevator shaft, this number of patients must be delivered at the Middle wing as well. On peak moments such as the beginning of the day around 07:45, this will result in even more congestion around the Middle wing than already was experienced in the situation before the split-up. However, some patients operated by the OOG (ophthalmology) specialization need less attention during recovery and can recover at the holding. These are the patients that undergo surgery from a chair. A second consequence of the split-up is that the occupation of the holding and recovery differs more throughout the day. At the beginning of the day all the patients will be delivered at the holding on the Middle wing resulting in a high occupancy peak. After all the patients are brought to the ORs the holding will be empty until the next patients can be delivered for their surgery. And as the day goes on less patients will be delivered at the holding. Meanwhile, the recovery will be slowly filled up with recovering patients while at the beginning of the day the recovery on South is practically empty. These fluctuations in occupancy indicate a suboptimal use of the capacity of the holding and recovery. Of course, the two beforementioned consequences of the split-up are viewed from a logistic or capacity-oriented point of view. The decision for the split-up was based on staff-oriented reasons.

2.3 Distribution of Starting Times throughout the Day

2.3.1 Day Care Surgeries

In the first place we want to verify if the ORs indeed often start with day care surgeries. To check this, we divide the day in several timeslots of 15 minutes and count the amount of day care surgeries that are planned and indeed performed per time slot. Figure 5 shows the results in a graph. We can directly see that at the beginning of the day (time slot 08:00-08:15) the number of day care surgeries that are planned and started are a lot more than throughout the rest of the day. The cause of this peak in the number of surgeries that start is the fact that the OR-department has a routine of starting all ORs at 08:00. Figure 5 also shows that a considerable amount of ORs do not start at their planned time but start later (and sometimes earlier) than planned. Because of the late start at the beginning of the day the OR-schedule for the rest of the day can be influenced and surgeons need to catch up throughout the day, leading most of the time to overtime or cancellation of patients at the end of the day.

2.3.2 Clinical Surgeries

Since clinical patients account for almost 50% of all the patients in Erasmus MC and clinical surgeries take more time than day care surgeries, we added figure 23 to appendix A that visualises the starting times of the clinical patients . We see a similar pattern at the beginning of the day as with the day care surgeries. However, the number of clinical patients that start at the beginning of the day is even higher than the number of day care surgeries. Therefore, we can conclude that the OR-department does not only start with day care surgery, but with even more clinical patients. However, similar as to the day care patients the most surgeries start or have been planned in time slot 08:00-08:15.

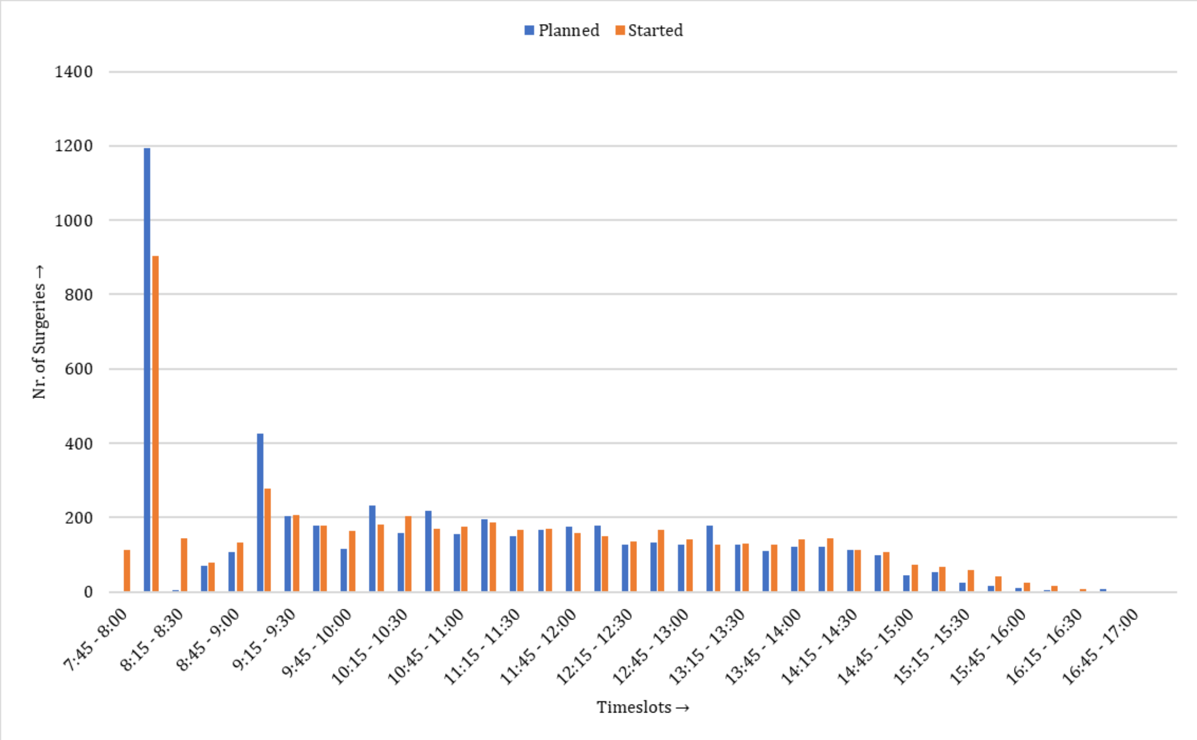


Figure 7: Starting times of day care surgeries in 2019 (n=28234; T=2019; source: HiX)

2.3.3 Starting Delays

The next question is in what amount day care surgeries start too late. In order to give an indication of the number of ORs that start too late we divided the day in time slots of 5 minutes and compared the number of planned surgeries with the number of actually started surgeries per time slot. From our dataset we found out that less than 1% of the day care surgeries had a different admission type than "elective". Therefore we concluded that the number of started day care surgeries near a timeslot with an amount of planned day care surgeries, were the same surgeries but performed earlier or later and were not changed because of emergency patients. In the first place we want to identify how the start of the day for day care surgeries usually takes place. In order to identify the problem of the late start the number of planned and started surgeries around 08:00 is analysed. Figure 6 shows the

Time	Classification	Percentage
[07:50, 07:55)	Earlier	0.34%
[07:55, 08:00)	Earlier	9.14%
[08:00, 08:05)	On time	34.62%
[08:05, 08:10)	Later	27.33%
[08:10, 08:15)	Later	13.66%
[08:15, 08:20)	Later	6.45%
[08:20, 08:25)	Later	3.27%
[08:25, 08:30)	Later	1.84%
[08:30, 08:25)	Later	0.75%

Figure 8: Percentage of day care surgeries planned at 08:00 starting a time slot earlier, later or in time slot 08:00.

percentage of day care surgeries that started in earlier timeslots, later timeslots or in the time slot of 08:00 can be found. Surgeries starting at 08:00 are important for the OR-department since these surgeries are the day start. If ORs already start too late at the beginning of the day, there is a high chance that the OR-schedule for the rest of the day is affected and other surgeries are likely to start later as well. In the end this could lead to overtime and surgery cancellations at the end of the day. Therefore, we want to know in what amount surgeries start later throughout the rest of the day. Since a calculation for every time slot as performed for the day start would not give

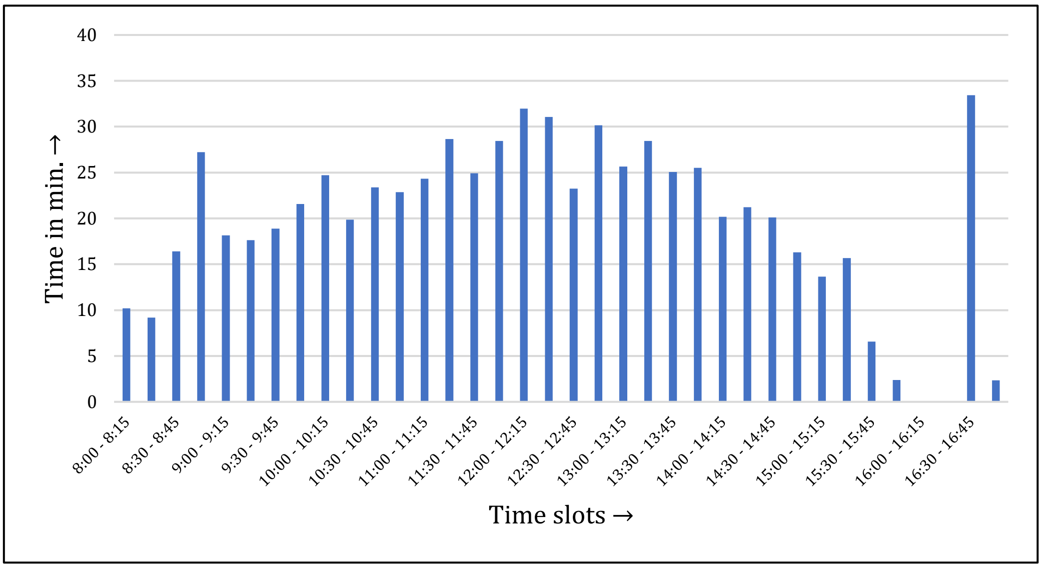


Figure 9: Average starting delay of day care surgeries per timeslot. To visualize only the amount of time surgeries start too late, the surgeries that started earlier than the planned surgery time were not treated as negative numbers but as zero. (n=28234; T=2019; source: HiX)

any clear insight, we use a different approach to visualize the late start of surgeries throughout the rest of the day. We calculated the average time a surgery starts later than planned per time slot. In this fashion we got insight in the average starting delay of day care surgeries per time slot. One could say that when planning a day care surgery in for example time slot 12:00-12:15 we could expect approximately 32 minutes delay for the start of this surgery when using the current way of planning. Figure 7 visualizes the average delay per time slot by making use of a column chart.

2.4 Case-Mix Classification

When doing calculations for OR-scheduling it is important to get information on the kind of surgeries that take place in the involved ORs. In order to classify the type of surgeries that are performed at Erasmus MC we use a classification based on surgery type duration and the coefficient of variation (Leeftink & Hans, 2018). These parameters are good indicators of the complexity of a scheduling problem. Since we want to tackle the problem stated earlier in this research by proposing an alternative planning method, this classification plays a key role. The first parameter is the duration divided by the OR-block capacity. This parameter is an indicator of the scheduling flexibility. If a hospital has primarily surgeries with a high duration, this leads to schedules with gaps. This is the result of too little short duration surgeries to fill these gaps between surgeries. The second parameter is the coefficient of variation. This is an indicator of the variability of a system and equals the standard deviation divided by the mean. This parameter affects the performance of realized schedules related to overtime, utilization or cancellations. A higher coefficient of variation indicates that surgeries have high variability in surgery duration. This leads to more uncertainty in the realization of schedules. A lower coefficient of variation results in easier scheduling problems since there is less risk of overtime and a higher probability of good utilized OR-blocks. To classify case mixes based on the two parameters, a visualization making use of a reference frame is used. On the x-axis represents the expected duration in relation to the total capacity of an OR-block and the y-axis represents the coefficient of variation.

For this research we used the dataset of Erasmus MC to make a classification per surgery type and used the average capacity of the OR-blocks where the surgery was performed in. The reason for this is the change of OR-block duration throughout the year and per OR. Figure 8 shows the classification for all the surgery types at Erasmus MC where every surgery type is represented by a point in the reference frame. From this case-mix classification we conclude that the different types of surgeries at Erasmus MC are wide spread over the reference plane and have a bigger clustering in the left lower quartile of the reference plane. Since Erasmus MC is a tertiary academic hospital this means that there is a larger amount of complex surgeries. Since these surgeries are more likely to take longer or differ more in duration it can be explained that the case-mix shows a more spread classification.

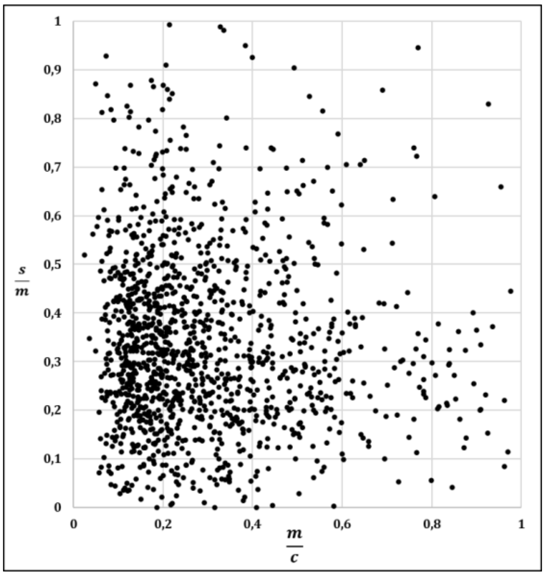


Figure 11: Case-mix classification for surgeries at Erasmus MC

A case-mix classification with several surgery types in the right lower corner can lead to a low utilization of ORs. As earlier mentioned, short surgeries with low variability can be very helpful to fill up the gaps for surgeries with a high duration. These surgeries can be used as small building blocks that fill up the created gaps in an OR-block. Since there is a cluster of surgery types in the left lower corner, it is valuable to identify these surgery types to explore the possibilities of better OR-scheduling. When splitting up the several admission types of the OR-department it became clear that these surgery types mainly originated from the day care unit. Figure 9 shows the

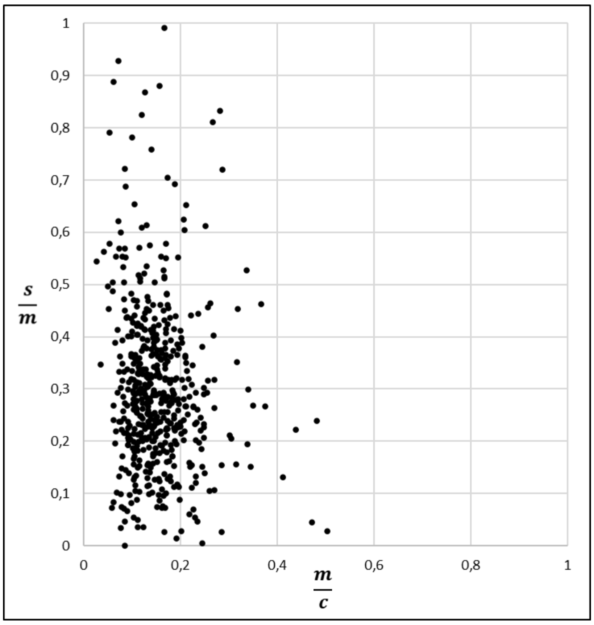


Figure 13: Case-mix classification for day care surgery

classification of these surgeries. In the first place this classification visualises that most of the day care surgeries occupy a small part of the capacity of the OR-block. Hence, day care surgeries are good surgeries to fill gaps in OR-blocks. Secondly, we see a big cluster of surgery types in the lower section of the coefficient of variation. Therefore, we can conclude that day care surgeries are less likely to differ from their expected duration and can make up a solid OR-schedule. For hospitals such as Erasmus MC, with a bigger amount of complex surgeries, it is very beneficial to have a day care unit that receive patients that undergo day care surgery. When making use of these surgery types in a smart way and combine them with the more complex surgery types, high utilization of ORs can be realised and less overtime and patient cancellations can be realised. However, when not taking the several characteristics of the surgery types into account, low utilization of the ORs can easily occur.

2.5 Suboptimal Use of Day Care Surgeries

As earlier mentioned, the day care department of Erasmus MC is at a different floor then the OR-department (4th and 6th floor respectively). Since the day care department has a higher patient throughput and must move more patients via the same elevator shaft, it is more likely that congestion during patient transport occurs and patients arrive too late for surgery. From section 3.2 we can conclude that indeed a lot of day care patients start at the beginning of the day and are mostly planned at 08:00. Section 3.3 taught us that the day care surgeries have a relatively short duration and have a low coefficient of variation, what means that they often do not differ in duration. This means that the presumption that often peak hours occur at the day care unit is very likely to be true. If we look again at figure 9 we can indeed identify several peak moments throughout the day. Since the day starts with all the ORs at 08:00, and these are often day care surgeries, there is a high chance that these ORs all end at the same time resulting in another peak of patient demand. This means that a lot of new patients from the day care unit must be transported at the same time to the OR-department resulting in late arrivals due to congestion or understaffed day care units. A second factor that influences the possibility of congestion is the split up of holding and recovery. Since peak hours are embedded in the current OR-schedule, congestion around the holding and elevator Middle is more likely to occur. Previously, when the recovery near elevator South was a holding and recovery and the holding near elevator Middle was a holding and recovery, patients (mostly clinical patients) could be transported to elevator South as well before surgery. However, in the new situation all departments are more likely to use elevator Middle since this elevator shaft is next to the holding. Therefore, the number of late patient arrivals is very likely to increase in the new situation. We can conclude that there is a call for smarter OR-scheduling (Hribar, Read-Brown, Reznick, & Chiang, 2017).

2.6 Analysis of Planning Methods

The last step that completes the analysis of the current situation of the day care surgery process is to identify the planning method of the OR-department. If we can determine the current planning method used by the OR-department, we can draw conclusions on the amount the planning methods contribute to the experienced problem.

2.6.1 Personal Heuristics

First, we tried to identify the planning method by asking the analysts of the ICM Project Team that work together with the planners of the OR-department if there is a predetermined scheduling method that is used to plan surgeries in ORs. It turned out that the OR-department makes use of a two-weekly blueprint with predetermined session times that are assigned to certain specializations. However, within the boundaries of the blueprint the planners use their "personal heuristics" to plan the surgeries. Since there are several planners that schedule the surgeries it is an unrealistic task to imitate these "personal heuristics" we decided to turn towards the dataset to get a better grasp on the current scheduling method.

2.6.2 Data Analysis

Since we are interested in what sequences surgeries are planned, we used three methods to identify the surgery sequences. The first method is based on OR-sessions and admission types. We came up with an algorithm that analyses the admission types of surgeries in a single OR-session and in what sequence these surgeries were performed. The following steps explain how this algorithm works:

1. Sort all surgeries of the dataset based on their session number.
2. Place all surgeries in a 3-dimensional matrix where every session inherits all its surgeries and every surgery its admission type, specialty, planned starting time and surgery duration.
3. Make use of a Bubble Sort to sort all surgeries in a session based on their surgery starting time.

4. Loop through all surgeries of every session and add the letter that represents its admission type to the session's result string.

5. Write every session's result string in a worksheet and make use of a pivot table to count the number of corresponding strings per specialty type.

6. Find the three most occurring strings per specialty type.

Table 3 shows the 3 most occurring admission strings per specialty type. The algorithm distinguishes 3 different admission types. Here "D" stands for day care surgeries, "K" stands for clinical and "O" stands for other admission types. To the left of the admission string the frequency of occurrence in the dataset is added.

Specialty	1		2		3	
	Amount	Sequence	Amount	Sequence	Amount	Sequence
CAR	55	KKKK	49	KKK	40	K
CHI	185	K	165	KK	102	KKK
CTC	744	KK	327	K	79	KKK
DER	8	D	8	K	7	KKK
GON	29	K	29	K	22	KKK
GYN	22	KK	12	KKK	9	KDK
KAA	51	K	21	KK	18	DKK
KNO	50	DK	44	K	38	DDK
NCH	246	KK	183	KKK	88	K
OCH	60	K	49	KK	24	KKK
OOG	61	DDDDD	49	DDDDDD	26	DDD
ORT	117	KK	100	KKK	41	DKK
PLA	100	K	46	KK	36	DK
PYN	27	DDDK	15	DDKK	10	DDK
RTH	32	D	13	O	7	DD
TRAU	43	KKK	31	KK	19	DKK
TXC	84	K	74	KK	53	KKK
URO	76	KK	51	K	43	KKK
VAT	87	KK	28	K	24	KKK
Other	38	K	17	KK	4	D

Table 3: Most occurring admission type sequences

The second method is equivalent to the first one but applied to the surgery duration. An extra step is added where the duration of a single surgery is compared to the average surgery duration of a specialty. If the surgery is shorter the letter "S" (shorter) is used and if the surgery takes longer than the average surgery length the letter "L" (longer) is used. In this way we can obtain the surgery duration string per session and find the most occurring strings per specialty type. Table 9 in appendix A shows the result of this algorithm. For the third method we divided a day in four timeslots and looked for every specialty type what the average surgery duration is per timeslot. After this we gave every timeslot per specialty type a number from 1 to 4 to indicate in what order the specialty prefers their surgeries to be scheduled. In this order 1 means the timeslot with on average the longest surgery duration and 4 the timeslot with on average the shortest surgeries. A timeslot can also have the tag "not planned" what means that no surgeries are planned in this timeslot for this specialty. Table 10 in appendix A shows the result.

2.6.3 Conclusions on Planning Methods

We conclude that it is difficult to describe a uniform planning method that is used to plan the surgeries at the OR-department. However, from the data analysis on the sequence of admission types per specialty we conclude that there are several specializations that have day care surgeries in their most occurring sequence. These specialties are: DER, GYN, KAA, KNO, OOG, ORT, PLA, PYN, RTH, TRAU and Other. All specialties that have day care surgeries in their most frequent sequences, except for GYN, start the sequence with one or more day care surgeries. From the case-mix visualisation we learned that day care surgeries have a rather short duration and do not deviate a lot from their expected surgery duration. Since several specialties and therefore several ORs start their day with day care surgeries, this will create a repetitive pattern throughout the day. Figure 10 shows this principle. If we have two ORs that start their session at the same time and with a day care surgery, these surgeries are likely to also end at approximately the same time. If the next surgery for both ORs is also a day care surgery this means that these surgeries are also likely to start at approximately the same time. And so on. Hence, if we have several ORs that start their day in this fashion, this means that the transportation moments for day care patients will occur at peak moments. Additionally, if the recovery time of day care patients have also approximately the same duration, also peak moments for patient pick-up could occur. If these peak moments occur at the same time as the peak moments for patient delivery a lot of congestion and late patient delivery, leading in the end to ORs that start too late. Situations where ORs alternate the admission types for their surgeries throughout the day, or plan mainly clinical patients have a higher chance of spreading the transportation moments throughout the day since the coefficient of variation is larger (surgery duration differs more) and the surgeries have different surgery lengths. Figure 11 shows an example of this principle.

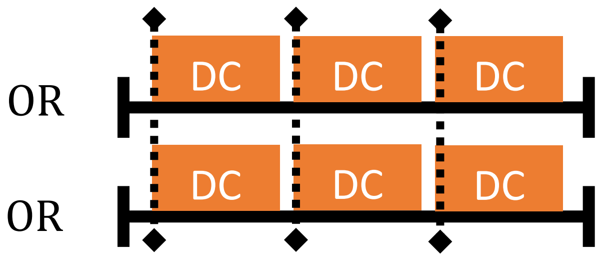


Figure 14: Example of a repetitive pattern for day care surgeries

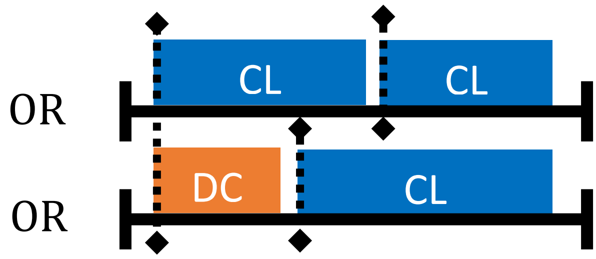


Figure 15: Example of an alternating pattern in surgery scheduling

CHAPTER 3: POSSIBLE SOLUTIONS

Chapter 2 described the current situation at Erasmus MC and we concluded that there was a call for smarter OR-scheduling where the characteristics of day care surgeries are used in a beneficial way. This chapter elaborates on the possible solutions that could contribute to a better OR-schedule. In order to come up with solutions we elaborate on a solution that came up by the hospital staff and introduce our personal approach to tackle the problem.

3.1 Expand Number of ORs

In The Netherlands but also in other countries there is a large trend of standardizing healthcare in hospitals. Since the techniques to cure diseases and possibilities to live with impairments are rapidly developing it is easier to keep people longer alive. The life expectancy in The Netherlands has increased tremendously in the last couple of years and therefore people become older on average. A simple result of this is the increasing pressure on healthcare institutions such as hospitals (Peters, 2015). Therefore, hospitals need to expand and must be able to provide care in a faster way. Standardizing healthcare is a sustainable solution to speed up the process of providing care. This is also the reason why outpatient clinics or day care departments are rising in popularity. Patients hold a bed for a much shorter time and do not need to stay overnight. The surgeries often take a short amount of time resulting in better scheduling possibilities (Lemos, Jarrett, & Philip, 2006).

As a result of this national trend hospitals try to shift more patients to their outpatient clinic and day care department. As earlier mentioned, Erasmus MC has currently two ORs next to their day care department. Currently there are plans to expand the number of ORs at the day care department and add two more ORs to the department. Of course, this is a large investment and it is questionable whether the plans are approved within a short time span. Since the number of elevator movements will decrease it is much likelier that less day care surgeries will start later than planned. A footnote to this possible solution is that consequently less day care surgeries will be performed at the OR-department at the sixth floor, meaning that the beneficial properties of day care surgeries (short and constant surgery times) are less available at the OR-department.

3.2 Improved Day Care Scheduling

In chapter 3 we concluded that all ORs plan to start their first surgery at 08:00. By analysing the most occurring sequences of admission types per specialty, we also saw that specialties often start their session with day care surgeries and that there is a tendency to plan several day care surgeries consecutively. The combination of these two scheduling properties result in repetitive transportation patterns throughout the day. Since a nurse at the day care department must handle relatively more patients throughout the day than a nurse at the clinic, these repetitive patterns in surgery starting times result in high peak moments for patient transportation that can be difficult to handle by the day care nurses. After analysing the number of surgeries that start per timeslot, we directly saw that there was a large peak of day care surgeries starting at 08:00. With the information we now have on the process of day care surgery we introduce the concept of a "differentiated start". In fact, this means that the first surgery of an OR-session does not start always at 08:00 but that we try to differ the starting times around 08:00 and let some ORs start at 07:45 and some ORs at 08:15. By doing so we try to break the repetitive pattern of starting day care surgeries throughout the day. A second solution that we propose is to try to better alternate the admission types in an OR-session. Figure 12 shows an example of this method where two OR-sessions have more alternating admission types resulting in spread starting times for day care patients. The main idea behind this solution is to look at all the OR-schedules throughout a day and try to minimize the number of day care surgeries that start within 15 minutes of another day care surgery.

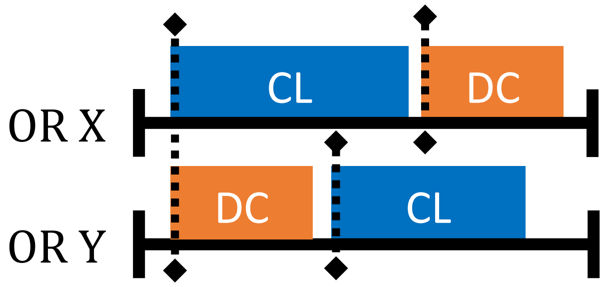


Figure 16: Example of alternating scheduling method

3.3 Problem-Solving Approach

Since implementing the above-mentioned solutions can have a large impact on the OR-department and can even have consequences outside the OR-department, one needs to test such interventions. To investigate the implementation of new OR-scheduling we will use a Monte Carlo simulation method. In academic literature this is a widely used method for OR-scheduling (Cardoen, Demeulemeester, & Beliën, 2010). In the first place the current situation is imitated until the system can be used validly to represent the reality. The next step is to implement proposed changes in OR-scheduling. For this research we decide to focus on the concept of differentiated starting times and leave the solution for alternating admission types of surgeries throughout the day for further research because of the difficult nature of optimally scheduling surgeries within an OR-session. In the next chapter we will elaborate on the Monte-Carlo simulation.

CHAPTER 4: SIMULATION MODEL

This chapter describes the Monte Carlo simulation proposed in chapter 4 that is used to explore changes in OR-scheduling at Erasmus MC. The first step is to imitate the current situation such that the model can be validated. The second step is to implement the proposed changes and evaluate the system. The evaluation is based on key performance indicators to measure the system performance.

4.1 A Monte-Carlo Simulation

In a Monte Carlo simulation, the goal is to model risk in an environment where the outcome is subject to chance (Robinson, 2014). In this case the environment that needs to be modelled is conceived as a set of combined distributions. These distributions are combined in such a way that an outcome can be determined. Figure 13 illustrates the idea behind a Monte Carlo simulation. In this case three sources of chance are combined in the simulation and produce a distribution of outcomes. In a Monte Carlo simulation, a physical process is not repeated one time (such as in a discrete-event simulation for example) but is repeated several times, every time with a different starting condition.

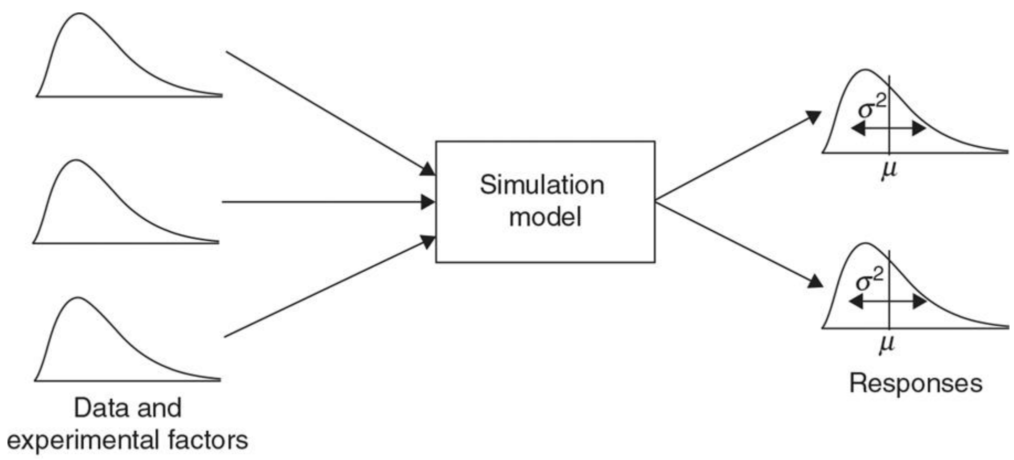


Figure 17: Idea behind a Monte-Carlo simulation

4.2 Conceptual Model

If we place the concept of Monte Carlo simulation on our case, we want to model the planning and realization of surgeries of the SPIN-department at Erasmus MC. Since we have access to several data points of surgeries, we can use this data to make a representation of the real world by fitting distributions around the data. When a solid representation of the real world is made, we can start changing several variables of the model. In this way we test our proposed solution of differentiated starting times. After that the model could also be used as a tool by the hospital to test further interventions on the process of scheduling surgeries. Before we start creating the simulation model, we construct a conceptual model of the simulation (Robinson, 2014). In this conceptual model the process is visualized such that we have contours to create the simulation model. Figure 14 shows the conceptual model of the simulation.

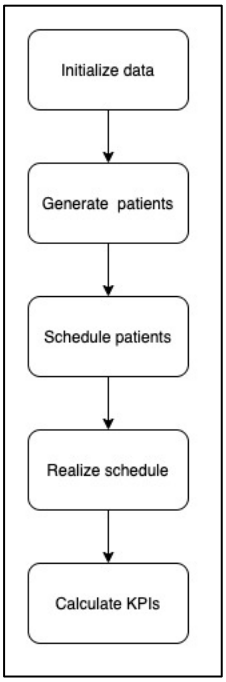


Figure 18: Conceptual Model

4.3 Simulation Software

The next step of constructing a Monte Carlo simulation is choosing the right software. Simulation models can be developed using general purpose languages such as Visual Basic, C# and Java. The use of these languages gives lots of freedom to create models but can cost a lot of work since models have to be built from scratch. To create a simulation model for operating room planning we decided to make use of the inside programming language of Excel: Visual Basic. When making use of a language such as Java or C# a good understanding of the programming language is needed to adjust and interpret the model and it can be costly to purchase the software for organizational usage. Since Erasmus MC already makes broad use of Excel and Visual Basic can be understand rather quickly with some experience in other programming languages, it is a straightforward choice to use Visual Basic as our programming software (Farrance & Frenkel, 2014) (Botchkarev, 2015). In this way the simulation model can be easily used for future research concerning capacity management at Erasmus MC.

4.4 Data Structure

Before starting to write the program, we set up the data structure of the program. In order to visualize the data structure, we make use of an entity relationship diagram. In this way the relationships between the different data types used in Visual Basic are visualized. Figure 15 shows the relationships between the different types of the simulation model. The different types function as the basis of the entity relationship diagram. from the diagram we see that we consider a type *TDay*. Every day has several ORs that operate on that day and these ORs are of type *TOR*. Such an OR has a schedule with the planned surgeries for that day of type *TORSchedule*. The OR-schedule consists of several surgeries that take place on the corresponding day, each of type *TSurgery*. Then we have the different specialties that can be scheduled in an OR. Since these specialties have certain predefined data such as the length of a surgery, the supertype of specialties is of type *TData*. This type can have several specialties of type *TSpecialty*. A specialty has different types of surgeries with specifications. In this case these surgery types correspond to the admission types "clinical", "day care" and "other". These surgery types are of type

TSurgeryType. Based on the OR-regulation of Erasmus MC, we assumed that the different specialties at Erasmus MC plan their surgeries per week since every specialty must deliver their OR-planning before Monday 12:00 of the week before the surgery takes place. Therefore, every specialty has for every week a surgery list of type *TSurgeryList* consisting of several surgeries of type *TSurgery*. Every type consists of several variables. An OR has for example an OR-number, a specialty ID scheduled with a corresponding specialty name and an OR-schedule.

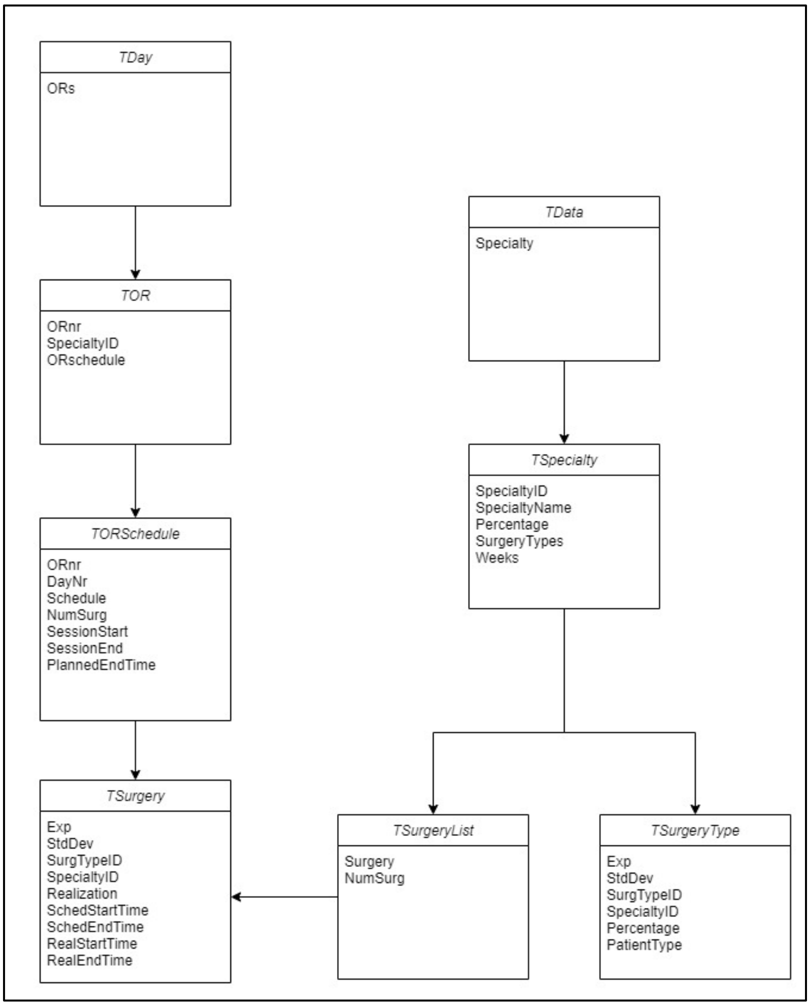


Figure 19: Entity relationship diagram

4.5 Subroutines

The next step in creating the simulation model is to translate the conceptual model to Visual Basic by programming the several steps of the planning system. In this chapter we explain the code that was used to program the simulation model. The simulation model consists of several steps that are mainly performed by so-called subroutines. In the Visual Basic language such a subroutine can be used to do a specific task. Subroutines are indicated by the word *Sub* followed by the name of the subroutine and an opening and closing bracket with some possible input variables in between. The subroutine is ended by the statement *End Sub*. In between these statements the code to perform the task is written. In the following sections we will elaborate on the subroutines that form the steps of the simulation model by dividing them among the steps in the conceptual model. The Visual Basic code can be found in appendix D.

4.5.1 Initialize Data

The first step is to initialize the data. This is done by the subroutine *InitData*. The subroutine is used to load all the input data into the memory of the system. The number of weeks is determined by multiplying the number of runs by 2. Erasmus MC makes use of a blueprint that allocates specialties to ORs and this blueprint differs per even and odd week. The number of days is determined by multiplying the number of weeks by the number of days per week. Since the blueprint is only used for the weekdays of the OR-planning we make use of 5 days per week. By making use of the blueprint we give all open ORs on every day a specialty, session starting time and session ending time and a unique session ID. After that the specialty data is initialized where every specialty has three surgery types. These types are equal to the three possible admission types of patients: clinical, day care and other. Every surgery type has a specialty ID, surgery type ID, expected surgery duration, standard deviation and relative frequency of the specialty type. The last step is to generate patients per specialty in every week. This is done by using the GeneratePatients subroutine per specialty in every week.

4.5.2 Generate Patients

This subroutine needs three input variables: the week number, specialty number and the number of patients to generate. Since we had no data available of the number of surgeries that need to be scheduled for a certain week, we used random runs to determine the mean and standard deviation of the number of surgeries that fit in the system per odd and even week for a specialty. The subroutine handles every individual patient until the generated amount is reached. First, a random surgery type must be drawn, taking the relative frequency of every surgery type into account. This is done by drawing a random number on the interval [0,1) (the RND( ) function in Visual Basic automatically includes 0) and increasing the surgery type number until the cumulative percentage of the relative frequency is greater than the random number. The last surgery type number is the one that is drawn from the surgery type population. The surgery is added to the surgery list of the specialty for the specified week and information is assigned to the surgery. The expected value of the surgery length is based on the mean of the surgery type and a random number drawn from the interval [-15, 15]. This random number is added to the expected length in order to create more different expected lengths for the surgeries. In this way we can measure the impact of scheduling algorithm such as "shortest first" or "longest first" in a better way since there are less surgeries with the same expected surgery duration. In reality the OR-department plans their surgeries based on the expected length of a surgery using the average surgery length of the last 10 interventions with the same intervention code and performed by the corresponding surgeon. This can also be adjusted by an estimation of the surgeon himself. Since we leave the possibility that a surgery can be performed by different kind of surgeons out of the scope of this research, we use this rather simple method to create different estimations for the surgery length. Next, the surgery gets the corresponding information assigned such as specialty ID and admission type. Lastly the realization of the surgery length is determined. We assume that the lengths of surgery types are log-normally distributed, making use of the mean and standard deviation of the surgery type.

4.5.3 List Scheduling

The next step is to schedule the patients in ORs. The GeneratePatients subroutine created a surgery list for every week and specialty. First the surgery lists must be ordered based on a planning method and the patients have to be allocated to an operating room based on an allocation method. The procedure is repeated for every specialty in every week. It is possible to select a planning method to perform the simulation by. The user can choose which planning method and which OR-allocation method is used to schedule the patients.

For both the shortest first and longest first planning algorithm we make use of a bubble sort algorithm to sort the surgery lists. Bubble sort is a rather inefficient sorting algorithm, but since the surgery lists per week and specialty type are not that long it is enough to work with in the simulation model. The bubble sort algorithm is an adjusted version of the original bubble sort algorithm and can be used to sort from small to large and from large to small. In order to swap the surgeries on the surgery list a special subroutine is used to swap all patient information in the right way. The algorithm works as follows:

1. Loop through the surgery list of n surgeries and compare every surgery with the first surgery. Swap both surgeries if their surgery lengths are in the wrong order. After that, move on to the next surgery.

2. Loop again through the surgery list, but then start the same procedure at the second element, because the first surgery had the smallest/largest surgery length in the surgery list.

3. Repeat but start at the third element.

4. Move on in this fashion.

n-1. Repeat the procedure but start at the n-1th surgery.

n. Done.

Next to the SF- and LF-algorithms there is also an option to use a random sorter. The random sorter creates a surgery list where the surgeries are ordered in a randomized way. The algorithm works as follows:

1. Select the first surgery on the surgery list. Draw randomly another surgery from the list and check if it is not the first surgery on the list. If it is not the same, swap the surgeries.

2. Select the second surgery on the surgery list and do the same but check if it is not the second surgery on the list.

n. Select the last surgery and perform the procedure one last time.

n+1. Done.

After the patients have been scheduled in the right order on a surgery list for a specialty in a specified week, the patients must be allocated to an OR. Since there can be several ORs in a week that have the same specialty scheduled there are different possibilities to schedule a patient in an OR. In order to allocate patients to ORs we make use of two algorithms that originate from the field of memory management [reference to memory management algorithms used for planning]: the First Fit Algorithm and the Best Fit Algorithm. The First Fit Algorithm schedules the patient in the OR that meets the requirements to let the surgery take place and has the earliest starting time. In fact, this means that the surgery is scheduled in the largest available block. The requirements that the OR must fulfil are:

- The corresponding specialty is planned in the OR.
- The expected duration of the surgery fits in the time left for the OR.

This algorithm is fast and simple but highly inefficient and usually results in a low occupancy of ORs**.** The second allocation algorithm is the Best Fit Algorithm. The Best Fit Algorithm takes all the available ORs into account that meet the earlier mentioned requirements and allocates the patient to the OR where the expected duration fits best in the available time left. Hence, the algorithm searches for the smallest available block. This algorithm is more efficient than the First Fit Algorithm and results in a higher occupation of the ORs. The downside of this algorithm is that there will be more overtime and more patient cancellations if the real surgery duration differs a lot from the expected surgery duration. Both algorithms return the OR-number and OR-day that meet the requirements of the algorithm and plan the surgery in this OR on the specified day. This means that the surgery and its information is added to the OR-schedule of this OR and the planned end time is adjusted by adding the expected surgery length to the scheduled starting time plus a possible changeover time.

4.5.4 Schedule Realization

The next step is to simulate the realization of the schedule. The schedule can be realized in two different ways: cancelling surgeries that have a real starting time that takes place after the session end time or cancel no surgeries. The user can choose between these options. If the patient is not cancelled, we check if the patient is the first one scheduled in an OR. If this is the case, we have:

$$real\ starting\ time\ S_1 = \ scheduled\ starting\ time\ S_1 \tag{1}$$

For $i \ = \ 2, \dots, N$ we have:

$$real\ starting\ time\ S_i = scheduled\ starting\ time\ S_i \tag{2}$$

Or:

$$real\ starting\ time\ S_i = real\ end\ time\ S_{i-1} \tag{3}$$

If the patient is not the first one scheduled in an OR we check if the real end time of the previous patient is earlier than the scheduled starting time. If this is the case the real starting time of the surgery is equal to the scheduled starting time since we assume that a surgery cannot take place earlier than its appointment time. However, this can be the case since clinical patients are often already a couple of hours before the surgery in the hospital. On the other hand, day care patients are not likely to be earlier ready since the admission of patients happens shortly before they get surgery. If the real end time of the previous patient is not earlier than the scheduled starting time of this patient the real starting time of the surgery is the real end time of the previous surgery (changeover time already included). The next step is to realize the end time of the surgeries. If the surgery is the last surgery for this day, the surgery does not need any changeover time and we have:

$$real\ end\ time\ S_N = \ real\ starting\ time\ S_N + realization\ S_N \tag{4}$$

And if the surgery is not the last one for this day we have for day number $i \ = \ 1, 2, \dots, N - 1$:

$$real\ end\ time\ S_i = \ real\ starting\ time\ S_i + realization\ S_i + changeover\ time \tag{5}$$

The last step is to set the real end time of the schedule equal to the real end time of the surgery.

4.5.5 Calculate KPI

After the simulation has run and the data is loaded in the system's memory some measurements must be created. In order to measure the adjustments to the system we have created a KPI. The KPI can be described as the percentage of surgeries in timeslot t on day d per admission type γ. Where A can be a timeslot of 15 minutes during a day, B is the day of the week and the admission type can be clinical, day care, other or all admission types combined. First, we consider all days of the simulation and check if the selected day is equal to the day for which we must calculate the KPI. The simulation model only considers 5 (working) days per week where the selected day for the KPI is changed to a number (Monday = 1, Tuesday = 2, ..., Friday = 5). In order to get the right day number from the total number of days the simulation performs, to compare to the day number for the KPI we use the following formula with the number of days $d = 1, \ldots, N$ is the day number of the simulation:

$$Day\ of\ the\ week = d - \left\lfloor \frac{(d-1)}{days\ per\ week} \right\rfloor \times days\ per\ week \qquad (6)$$

In the formula we make use of the floor function that takes a real number x as input and gives as output the greatest integer less than or equal to x, denoted as $\lfloor x \rfloor$. The next step is to consider all patients scheduled in every OR for this day and check what admission type the patient should be selected on. If the patient has the required admission type, we should check if the surgery takes place (the realization of the surgery is not equal to zero). If this is the case, we increase the total number of surgeries on this day by 1. Then we must check if the real starting time of the surgery falls into the interval of the selected timeslot. If this is the case, we increase the total number of surgeries taking place in this timeslot by 1. Since the KPI is calculated for every week the simulation runs, we need to obtain the KPI after the last day of the week. We check whether we arrived at the last day of the week by checking if the week number belonging to the next day is not equal to the week number of the current week. To calculate the week number of day $d + 1$ we use the following formula with the number of days $d = 1, \ldots, N$ is the day number of the simulation:

$$W_{d+1} = 1 + \left\lfloor \frac{d}{days\ per\ week} \right\rfloor, \qquad (7)$$

Since we make a distinction between even and odd weeks in the blueprint, we check whether the following equation holds for d the day number of the simulation:

$$W_d \equiv 0 \pmod 2, \qquad for\ d = 1, \ldots, N \qquad (8)$$

Where we have:

$$W_d = 1 + \left\lfloor \frac{(d-1)}{days\ per\ week} \right\rfloor, \qquad for\ d = 1, \dots, N \tag{9}$$

If this the equation holds, we have W_d is an even week and if it does not hold, we have W_d is an odd week. To calculate the KPI for an even or odd week we have:

$$KPI = \frac{\#\ surgeries\ starting\ in\ timeslot\ t\ of\ admission\ type\ \gamma}{\#\ total\ surgeries\ on\ day\ d\ of\ admission\ type\ \gamma} \tag{10}$$

After the KPI is calculated for the corresponding week, the KPI is written in the corresponding worksheet to be used to construct a boxplot.

4.5.6 Surgeries Starting Per Timeslot

In order to make a visual representation of the number of surgeries starting per timeslot a subroutine is used to count the number of surgeries starting per timeslot. All patients in every OR on every day are considered. For every patient the planned and real starting time is considered by looping through all timeslots of 15 minutes from 07:15 until 20:00. If the surgery is scheduled (the scheduled starting time does not equal zero) and the scheduled starting time is later than the lower bound of the timeslot and earlier than the upper bound of the timeslot, the number of planned surgeries in the corresponding timeslot is increased by one. After that the admission type is checked to increase the number of surgeries for the corresponding admission type in this timeslot. For every timeslot we do this in the same fashion for clinical surgeries. By doing so we get 6 different categories: clinical planned, day care planned, others planned, total planned, clinical real, day care real, others real, total real. Since we make use of a matrix where we count all surgeries per category per timeslot, the last step is to print the 53×9 matrix in a worksheet to be able to visualize the values in the dashboard.

4.5.7 Surgeries On-Time

In order to determine the number of surgeries that arrive too late we make use of a subroutine that considers all patients in every OR on every day and checks per specialty type how much surgeries are on time and what the total number of surgeries that take place is. The surgery is assumed on time if the surgery takes place and meets the following requirement:

$$T_{Real} \leq T_{Scheduled} + accepted\ delay, \qquad with\ T_{Real} \neq 0 \tag{11}$$

If the surgery takes place but does not fulfil the above requirement, the surgery is too late. After all surgeries are considered, the result matrix with the number of surgeries on time and the total number of surgeries taken place per specialty is written in the table in the corresponding worksheet.

4.5.8 OR-Occupancy

In order to calculate the occupancy of the ORs we use a subroutine that considers every surgery in every OR on every day of the week. The occupancy for a single session is calculated as follows:

$$Occupancy = \frac{\sum_{k=1}^{s} t_k}{t_{session\ end} - t_{session\ start}}, \qquad k = 1,2,\ldots,s \tag{12}$$

And therefore, the total occupancy of the simulation is calculated as follows:

$$Total\ occupancy = \sum_{i=1}^{d}\sum_{j=1}^{24}\frac{\sum_{k=1}^{s} R_{ijk}}{E_{ij} - S_{ij}}, \qquad i = 1,2,\ldots,d;\ j = 1,2,\ldots,24;\ k = 1,2,\ldots,s \tag{13}$$

Where E_{ij} is the session end time matrix with time in minutes of the OR-session in OR j on day i, S_{ij} is the session start time matrix with time in minutes of the OR-session in OR j on day i and R_{ijk} is the surgery duration matrix with duration of the surgery in minutes of surgery k in OR j on day i.

4.5.9 Surgery Duration

The last subroutine is used to create a visual representation of the surgery duration by making a histogram for every specialty and of the surgery duration in total. In order to create a histogram, the surgery duration must be divided in bins. We choose to use a bin width of +/- 17 minutes, where the last bin has an upper bound of 500 minutes. There are some recommended methods to determine the bin size by making use of the number of observations, minimum value and maximum value (Sturges, 1926) or by using the first and third quadrant. However, in this case we use 30 bins of size 17 minutes and one bin with the residual values of more than 500 minutes so that we can use the same bins for every specialty type and the visual representation is still good enough to interpret the histogram of surgery lengths as a distribution. In order to get these values for all the bins of the specialties we consider all surgeries that take place and first check if the realization of the surgery length is larger than the upper bound of the previous bin and less than or equal to the upper bound of the current bin. If we arrive at the last bin, this means that we did not yet found the right bin and the surgery duration falls in the "more" category with a surgery duration larger than 500 minutes. In order to speed up the simulation process we make use of a

D_{ij} surgery duration matrix where for $i = 0$ and $j = 1,2,\ldots,31$ the matrix holds all the bin boundaries and for $i = 1,2,\ldots,s$ the matrix holds the number of surgeries inside bin j for specialty i.

4.6 Dashboard

In order to visualize the outcomes of the simulation and to easily adjust the input values for the simulation we created a dashboard in Excel. First, we will explain how to start the simulation, secondly, we will explain recalculate the KPI and thirdly, we will explain the different visual components of the dashboard.

4.6.1 Start Simulation

By clicking the "Start Simulation" button in the left upper corner a window with simulation options pops-up. The simulation model has the following options that can be edited by the user:

1. The first possibility is to determine the number of runs the simulation must perform. Since the simulation model makes use of a two-weekly blueprint, one run equals two weeks.

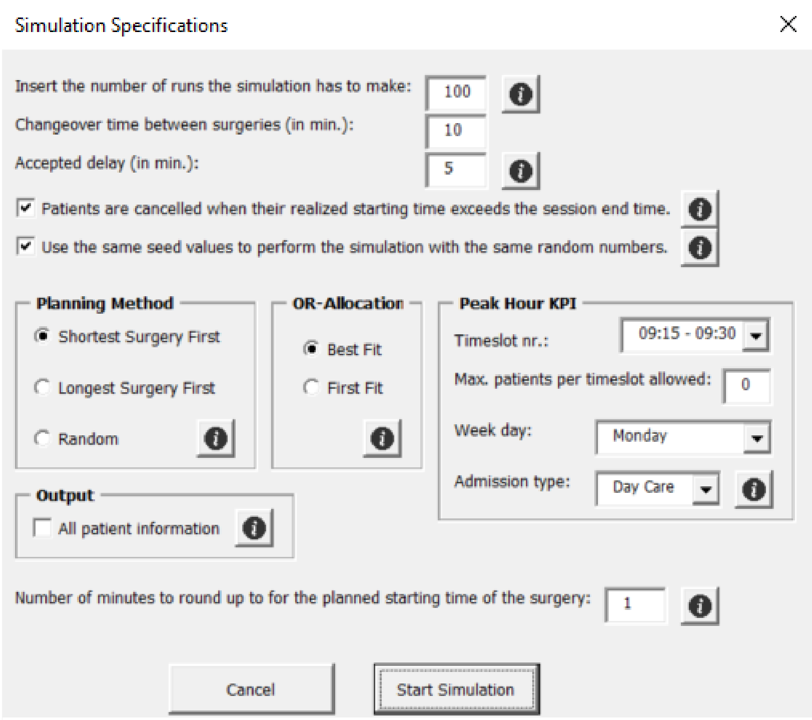


Figure 20: Simulation specifications window

2. The changeover time between surgeries is the minimum time in minutes that is needed between the two surgeries to prepare the next surgery.

3. The accepted delay in minutes is used to determine the percentage of surgeries that is considered as "on-time". If we fill in 5, the simulation will accept surgeries that start 1-5 minutes later than planned also as "on-time".

4. If the checkbox is checked that patients are cancelled when their realized starting time exceeds the session end time, the simulation model will not let all the planned surgeries take place. When the planned schedule for an OR is realized it can be the case that surgeries start later than planned. Checking the checkbox lets the simulation cancel all surgeries that have a real starting time that takes later place than the session end time. If the checkbox is not checked, all the surgeries will take place until all scheduled surgeries have performed.

5. The checkbox to use the same seed values to perform the simulation with the same random values can be used to compare different scenarios. Since the simulation model makes use of random variables, new random values are used every time the simulation is started. If this checkbox is checked the same random values will be used every time the simulation is performed.

6. In the planning method section, it is possible to select one planning method that is used to schedule patients based on surgery duration. Section 5.3.3 discusses how these planning methods work.

7. The OR-allocation can be done in two different ways. Best fit and first fit. Section 5.3.3 discusses these allocation methods.

8. The peak hour KPI section can be used to select the first KPI that becomes visible in the dashboard by using a boxplot. In de timeslot dropdown menu, the user can chose a timeslot of 15 minutes between 07:15 and 20:00 to analyse the KPI for. Next to that a maximum number of patients can be inserted. The simulation model will only count the number of patients that undergo surgery above the number inserted. A weekday can be selected to measure the KPI on different days of the week and an admission type can be used to specify the KPI for patients of an admission type or take all the patients all together.

9. The checkbox to print all the patient information as output can be checked by the user to get all patient information that is generated by the simulation model. When checking this option if the simulation model must run more than 100 runs will result in a long running time.

10. The last option is to insert the number of minutes to round up to for the planned starting time of surgeries. For convenience it can be sometimes useful to round up surgery starting times to quarters of an hour or to every ten minutes. The planned starting times of the surgeries will be rounded up to a multiple of the number of minutes inserted in this textbox.

4.6.2 Recalculate KPI

After the simulation is performed, the user can recalculate the KPI by pressing the recalculate KPI button in the upper right corner of the dashboard. The same information that can be inserted when starting the simulation can be adjusted after the simulation and can be recalculated. For the specified timeslot, preferred number of surgeries per time slot, weekday and admission type.

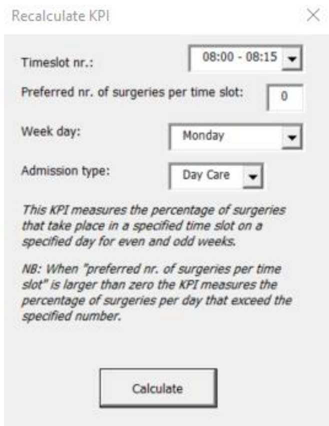


Figure 21: Recalculate KPI window

4.6.3 Visuals

When the simulation is performed the dashboard will display information about the simulation. In the right upper corner in the information frame some general information is displayed such as the number of weeks that is simulate, the total number of surgeries planned, the total number of surgeries that took place, the number of cancelled surgeries and some of the input values that the user inserted when starting the simulation. Underneath the information frame the percentage of surgeries that start on-time and the occupancy of the ORs in percentages. The pie chart in the left lower corner displays the percentage of surgeries that is performed per admission type. Next to the pie chart on the left side a slicer is inserted that can be used to select a specialty. The specialty or specialties that are selected are visible in the surgery duration graph. A histogram is used to display the distribution of the surgery length per admission type. This histogram can be used to verify the distribution of the surgeries per specialty type. Above the surgery duration histogram, a chart for the average number of surgeries starting per timeslot is displayed. The chart can be used to select the different admission types and the number of surgeries starting per timeslot in total. For every admission type the chart can also display the planned number of surgeries starting per timeslot and the real number of surgeries starting per timeslot (realization). The right part of the dashboard is used to display the KPI on surgeries starting in a timeslot per admission type. The KPI info frame shows the input values to calculate the KPI and below the KPI info frame a boxplot of the KPI in the odd and even weeks is shown. The second part of the dashboard is the schedule visualization sheet. Here it is possible to see how the operating-room schedule is realized for the first two weeks of the simulation. In the upper left corner, an operating room can be selected to display the schedules for. The upper blocks in every chart show the planned surgery times and the lower blocks the realized surgery times. Underneath every chart the planned specialty and the session starting, and end time are displayed for the OR. In this way the user can verify what the operating room schedule looks like.

CHAPTER 5: EXPERIMENTS

This chapter elaborates on the experiments that are performed by making use of the simulation model. Section 5.1 explains the importance of seed values. Section 5.2 determines the number of replications to validate the number of simulation runs. Section 5.3 discusses the interventions that were performed and section 5.4 discusses the results that came out of the experiments.

5.1 Seed Values

When performing experiments with the simulation model the outcomes must be comparable. Since the simulation model makes use of randomness, the output of one simulation run is different than the output of the next simulation run. When using the random function in VBA without specifying a seed value, the function will use the system timer as the new seed value. The seed value is used as an input for a pseudorandom number generator where VBA uses a linear congruential generator (LCG). When the pseudorandom number generator is called by different seed values it will produce different random numbers. However, if the pseudorandom number generator is called by the same seed values it will produce the same sequence of random numbers. This is particularly useful if we want to measure the impact of changes in the simulation model. In this way we can see exactly how the system performs different on several interventions.

5.2 Determine Number of Replications

A large advantage of using a simulation model to imitate a real-life situation is the possibility to perform a scenario repeatedly. In terms of the simulation model one repeat of such a scenario is called a simulation run. The OR-department of EMC makes use of a blueprint for twee weeks: an even and odd week. Throughout the year only two or three small changes are made to this blueprint and therefore we use the latest blueprint as input for the simulation model. In order to state the right conclusions about the simulation model it is important that we use the right number of runs per simulation round.

The simulation model we made is a terminating simulation. This means that a natural event specifies the end of the simulation run. For this simulation model the simulation run ends when one odd and one even week is simulated and therefore the blueprint is ended. Since we have a terminating simulation model, we do not have to take care of initialization bias by using a warm-up period. As can be seen from figure 18 it does not matter if we make one long run or if we make replications for the simulation model.

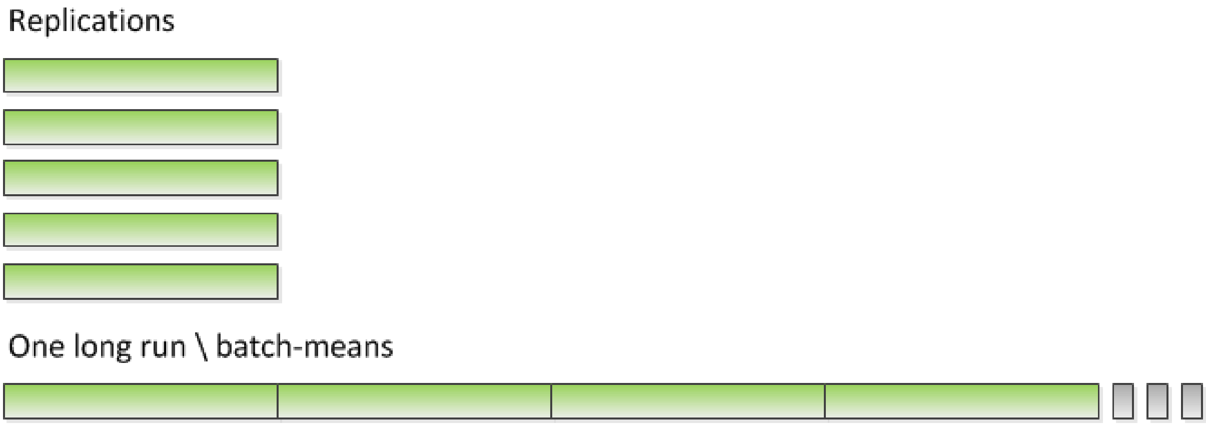


Figure 22: Replications and run length

Because of efficiency reasons we choose to make one long run and let the program write the KPI-values in an output sheet and calculate the average, standard deviation and the number of degrees of freedom up to the replication. We perform replications of the simulation until the width of the confidence interval, relative to the average, is sufficiently small. This is done by making use of the following formula:

$$\frac{t_{n-1,1-\frac{\alpha}{2}}\sqrt{\frac{S^2}{n}}}{|\bar{X}|} < d \tag{14}$$

Here d is the relative error, hence the deviation of the confidence interval about the mean. In our case we make use of an α of 5%. In fact, we identify from what number of replications the output is statistically reliable (Mes, 2018).

5.3 Interventions

The next step is to determine the interventions we want to perform and to measure their impact on the simulation model. The interventions that are tested are concerned with the starting times of the OR-sessions. The research is mainly focused on the day care unit. With the interventions we want to keep the number of patient transportations per time slot as low as possible. In chapter 4 we proposed a solution to use differentiated starting moments for the day care surgeries to alternate the number of transport moments as much as possible. Since it is difficult to imitate the planning method of the OR-department we perform the experiments with three different scheduling methods. In order to determine which session times should be adjusted, we select the specializations with the largest percentage of day care surgeries. Table 4 shows the selected specializations for day care surgeries, their total percentage of the number of patients and their relative percentage to the number of patients per specialization with respect to the other admission types.

Specialty	Admission type	Percentage	Relative percentage
OOG	Day care	6.93%	90.84%
KNO	Day care	4.33%	51.53%
VAT	Day care	4.26%	58.00%
PYN	Day care	3.60%	47.29%
CHI	Day care	2.47%	21.76%
RTH	Day care	1.99%	29.55%
ORT	Day care	1.99%	51.39%

Table 4: Percentage of day care surgeries per specialty

Table 5 shows the experiments to measure the interventions on the simulation model. The interventions on the starting times of specializations means that the total session duration is not adjusted, but the session starts the specified number of minutes earlier or later and ends the same number of minutes earlier or later.

Experiment	Planning method	Interventions on starting times in minutes						
		OOG	KNO	VAT	PYN	CHI	RTH	ORT
1	Random	0	0	0	0	0	0	0
1	SF	0	0	0	0	0	0	0
1	LF	0	0	0	0	0	0	0
2	Random	-15	+15	0	0	0	0	0
2	SF	-15	+15	0	0	0	0	0
2	LF	-15	+15	0	0	0	0	0
3	Random	-15	+15	-15	0	0	0	0
3	SF	-15	+15	-15	0	0	0	0
3	LF	-15	+15	-15	0	0	0	0
4	Random	-15	+15	-15	+15	0	0	0
4	SF	-15	+15	-15	+15	0	0	0
4	LF	-15	+15	-15	+15	0	0	0
5	Random	0	0	-15	+15	-15	+15	0
5	SF	0	0	-15	+15	-15	+15	0
5	LF	0	0	-15	+15	-15	+15	0

Table 5: Simulation experiments

5.4 Results

In this section we discuss the results from the experiments. Table 6 shows the values of the occupancy, the percentage of surgeries that start on-time, number of performed surgeries and the number of cancelled surgeries. In our experiments of adjusting the starting times these values do not change since we adjust the session end times of the ORs with the change of the starting times. However, we conclude from the experimental outcomes in table 6 that planning the shortest surgeries first leads to the lowest occupancy, the least number of cancelled surgeries and the lowest percentage of surgeries that start on time. If the shortest surgeries are planned first this will result in not enough time left in an OR-session to plan the longer surgeries and hence less surgeries will be planned and performed. Even though this scheduling method is not that efficient it can give a good representation of the schedules for some specialties since we found out that several specialties start their day with day care surgeries and day care surgeries are rather short surgeries.

Planning method	Replications	Occupancy	On-time	Performed surgeries	Cancelled surgeries
Random	500	76.6%	76.4%	335,506	14,339
SF	500	72.5%	76.2%	327,748	6,867
LF	500	76.4%	76.4%	332,042	15,527

Table 6: Experimental outcomes

Figure 19, 20 and 21 show the output values of the experiments for the KPI of the percentage of day care surgeries on Monday 08:00 in even and odd weeks. A larger decrease in the KPI means a larger impact of the adjusted starting times. If we look at the random scheduling method we already see an improvement of the KPI after the ophthalmology (OOG) start their surgeries 15 minutes earlier and ENT (KNO) their surgeries 15 minutes later. Only a small improvement is obtained from adding more differentiated starting times. If we look at the shortest first planning method we also see a large improvement of the KPI in experiment 2. We can see that the extra starting times in experiment 3 and 5 does not have that much impact, but experiment 4 has a larger impact on the number of surgeries starting per timeslot.

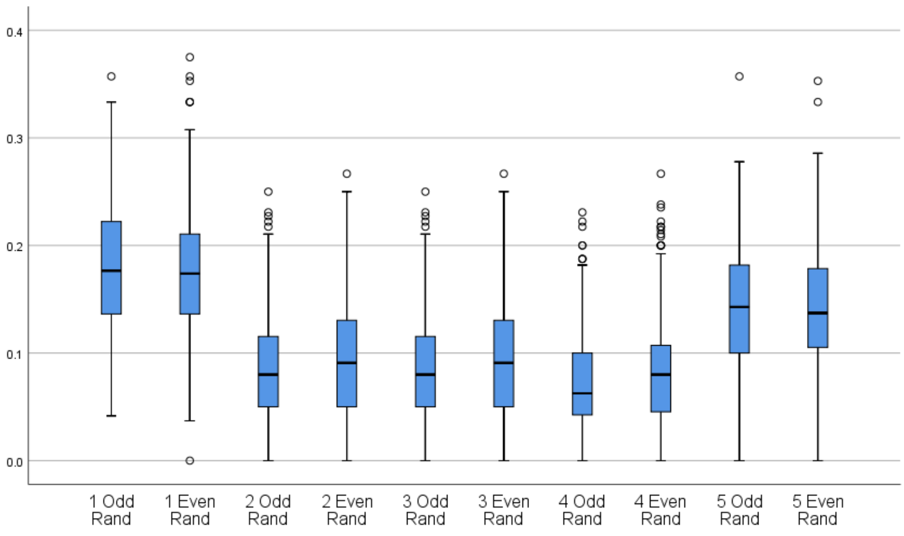


Figure 23: KPI boxplots for the random scheduling method

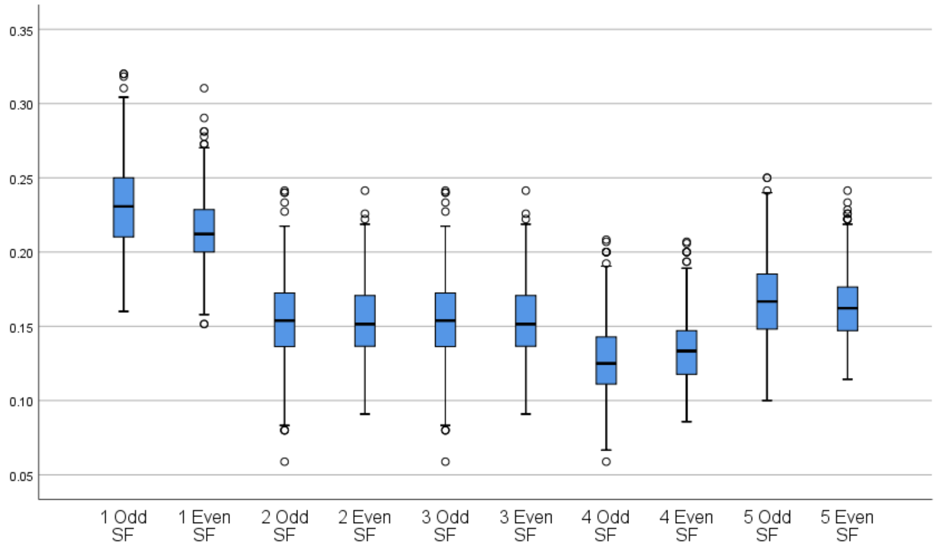


Figure 25: KPI boxplots for the SF scheduling method

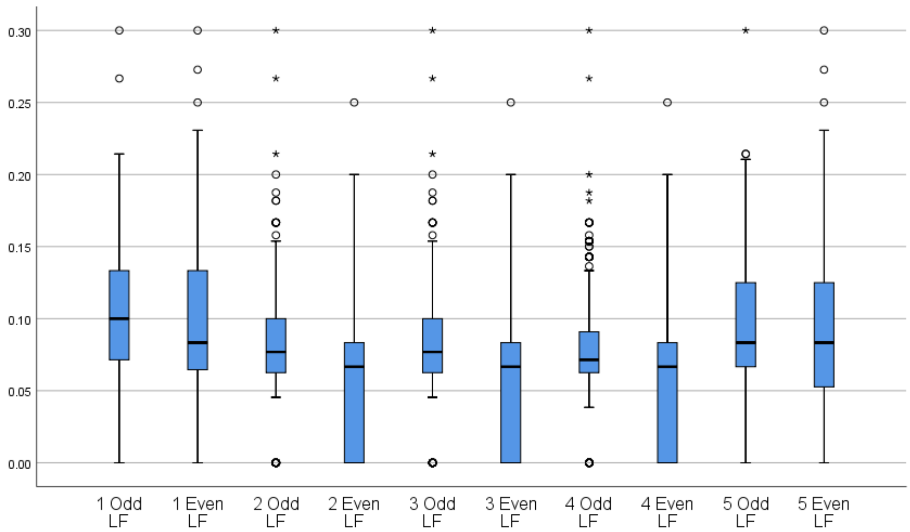


Figure 24: KPI boxplots for the LF scheduling method

CHAPTER 6: CONCLUSION

This chapter draws conclusions from the research. Section 6.1 comes up with recommendations for improvement at Erasmus MC. Section 6.2 elaborates on how the research was received by the hospital. Section 6.3 ends the chapter by doing some suggestions for future research.

6.1 Recommendations

The main conclusion of this research is that there are possibilities for Erasmus MC to decrease the number of surgeries that start too late. Especially for day care surgeries there are possibilities to decrease the number of surgeries that start too late. Since the day care nurses experience congestion at the beginning of the day and have difficulties to keep up with the number of transportation moments of patients throughout the day, we looked at the number of day care surgeries starting throughout the day. Especially at the beginning of the day there is a peak moment of the number of patients that must be transported since all ORs start at the same time with the first patient. After analysing the way of planning of the OR-department we see that several specialisations indeed start their day with day care surgeries. By making use of the case-mix classification we concluded that day care surgeries have a rather short and uniform surgery duration and deviate hardly from their expected surgery time. When several ORs start their day with day care surgeries and plan more day care surgeries after each other, this will result in peak moments for patient transportation throughout the day and especially for day care patients. The day care nurses have often trouble to deal with these amount of transportation moments for delivering patients at the holding as well as picking patients up from the recovery. Finally, this results in patients being delivered later than planned at the holding and surgeries starting later than planned throughout the day. Therefore, we suggested to make use of a differentiated start at the beginning of the day and spread the number of day care surgeries starting at 08:00. In order to show the impacts on the system we used a Monte-Carlo Simulation to perform experiments regarding the starting times of ORs. From the experiments we conclude that it can be helpful to make use of differentiated starting times for certain ORs. When different ORs want to start with day care patients on a day it can already be helpful to let one OR start 15 minutes earlier and one OR 15 minutes later. By doing so the number of transportation moments is spread more evenly throughout the day. The second recommendation is to try to better alternate the transportation moments of day care patients. If the transportation peak at the beginning of the day is brought down by letting all day care surgeries starting 15 minutes earlier or later this will still result in peak moments throughout the rest of the day. In order to bring these peak moments down as well it is of importance to look at the starting moments of day care surgeries throughout the day. The higher the number of day care surgeries that start approximately at the same time, the higher the chance of delay, overtime and cancellations. Therefore, we recommend trying to break through rhythmic patterns of day care surgeries by alternating the starting times of the surgeries and by alternating the order of day care patients by creating combined surgery sequences of clinical and day care surgeries.

There is another point of attention when letting a day care surgery start 15 minutes earlier. It can be difficult for the day care nurses to get the patient on time at the holding since the day care department starts their day with a general meeting to discuss all the patients that will come by during that day. When starting at 07:45 with the first day care surgery the patient must be at the holding at 07:30 while the day start for the day care department is at 07:15 and after that the intake of the patient should still take place. Letting the day care start take place after all the first patients are brought at the holding could be an interesting option to discover for the day care department. Currently the day care department already experiences trouble getting the patients in time at the holding even though the surgeries start at 08:00.

In chapter 3 we concluded that day care surgeries are valuable for the OR-department. Day care surgeries have a rather short and uniform duration, and the surgery duration can be predicted well. Therefore, these surgeries can be used as "socks in a drawer"; the small holes in the OR-planning that occur because of large surgeries can be filled up with day care surgeries. However, when these surgeries are planned in one sequence, the properties can have a counteracting effect. Repetitive patterns in transport moments will occur and result in peak moments throughout the day where the day care nurses cannot keep up with the number of patient transportations. Currently, the SPIN-department of Erasmus MC has made a plan to expand the number of ORs at the fourth floor, near the day care department. The department is considering this option to let more day care surgeries take place at the fourth floor. The plan will not be executed yet since adding two ORs is a large and costly operation. However, this decision is a good response to the increasing number of day care surgeries. Nevertheless, when these ORs will be added to the already existing two and day care surgeries will be mainly performed by these ORs, this decision will probably let the occupancy of the OR-department decrease heavily since there are less possibilities to fill gaps in OR-sessions.

6.2 Additional Value of the Research

The final presentation at the hospital pointed out that the involved parties found this research valuable. The staff of the OR-department could now see the effects of the split-up of the holding and recovery in a broader perspective. When making the decision of the split-up the staff was not aware of the effects on the patient transportation. Therefore the research could be beneficial for future decision making on this topic. The hospital has not yet determined if it will implement the concept of a differentiated day start. The research shows that there are possibilities to reduce the number of surgeries that start to late, but in practice it is more difficult to implement these changes. Components such as working time preferences of surgeons and nurses were left outside the scope of this research but have a large impact on decision making in hospitals. However, the capacity management made clear that this research was a good starting point for further research on the late start of surgeries. The most valuable contributions of this research to the hospital were the Monte-Carlo simulator including a user-friendly dashboard, the algorithm to determine the most occurring sequence of surgery types and the flowchart of the process of day care surgeries. One of the analysts of the capacity management was especially curious about the algorithm since it gives them more information about the way surgeries are planned. The flowchart gives insight in the day care process and was found as useful for when new adjustments to the day care department are made. The capacity management is excited to continue research on the extension of the number of ORs at the day care department and believed that the dashboard of the Monte-Carlo Simulation is a valuable starting point.

6.3 Further research

This research set a large step in the right direction to decrease the number of late starting surgeries and a large step in getting more insight from the surgery process at the OR-department of Erasmus MC. However, there are still more steps to take to improve the surgery process. Therefore, we will end this chapter with some suggestions for future research.

In the first place this research delivered a highly effective management tool in the form of a Monte-Carlo simulator. Since the simulation model is made in Excel and Visual Basic, the model can be easily adopted by the hospital and be further developed. The goal is to resemble the planning system of the OR-department as good as possible. There are 3 direct improvements that could be added to the simulation model:

1. It would be worthful to try to better predict the surgery durations by making use of a 3-parameter lognormal distribution and by making a better subdivision of surgery types instead of three admission types.

2. In the simulation model we did not take emergency patients into account. Erasmus MC makes use of 5 different emergency types with their own priority level, taking these patients into account gives a better representation of the reality. The availability of surgeons could be taken into account as well.

3. In the simulation model we took only the starting times of surgeries into account to predict the number of transportation moments per timeslot. This directly relates to the

peak moment at the beginning of the day. However, a second problem were the peak moments of transportation moments throughout the day. These peak moments include a combination of delivering patients at the holding and picking up patients at the recovery. To get a better impression of the number of transportation moments it is useful to research the time that day care patients must stay at the recovery after their surgery. If we can make a good prediction of the surgery duration and the recovery duration of day care patients, we can make an even more complete prediction of the number of transport moments throughout the day.

To make the future research as accurate as possible we also need better data than we currently have. The dataset we used for this research is based on data from 2019. The data that was produced during the period of this research is less useful because of the COVID-19 pandemic. The hospital had to scale down their number of surgeries to free up more staff and capacity for COVID-patients at the intensive care. It could therefore be a quite a task to obtain recent and trustworthy data for future research. To make time stamps such as the time the patient is ready to be picked up from the recovery it could be worthful to investigate possibilities to increase the reliability of data points since they are often exposed to human error. Such a possibility would be to make use of scanners. A bed or chair will have a barcode that is scanned when a certain process is started (the patient arrives at the holding, the surgery starts, the patient arrives at the recovery and the patient leaves the recovery). Erasmus MC makes already use of this kind of system at their clinical department to determine the number of available beds.

Finally, there lies a great possibility to investigate the ideal division of day care patients between the ORs on the fourth and sixth floor. This research could be valuable for the current situation where only a small number of surgeries can be performed at the fourth floor, but especially for the situation where two new ORs are added to the ORs at the fourth floor. The simulation model that resulted from this research could be a good starting point for both situations. With some adjustments to the simulation model, one could add ORs and one could choose to let a certain OR only take care of day care surgeries or let the division between day care surgeries and other admission types be dependent on a percentage.

Bibliography

Baesler, F., Gatica, J., & Correa, R. (2015). Simulation Optimisation for Operating Room Scheduling.

Berg, B., Denton, B., Erdogan, A., & Rohleder, T. (2014). Optimal Booking and Scheduling in Outpatient Procedure Centers. *Computers & Operations Research, 50*.

Botchkarev, A. (2015). Assessing Excel VBA Suitability for Monte Carlo Simulation. *ArXiv, abs/1503.08376.*, -.

Cardoen, B., Demeulemeester, E., & Beliën, J. (2010). Operating Room Planning and Scheduling: A Literature Review. *European Journal of Operational Research, 201*(3), 921-932.

Cayirli, T., & A Veral, E. (2009). Outpatient Scheduling in Health Care: A Review of Literature. *Production and Operations Management, 4*, 519-549.

Does, R., Vermaat, T., Verver, J., Bisgaard, S., & Van den Heuvel, J. (2009). Reducing Start Time Delays in Operating Rooms. *Journal of Quality Technology, 41*(1), 95-106.

Farrance, I., & Frenkel, R. (2014). Uncertainty in Measurement: A Review of Monte Carlo Simulation Using Microsoft Excel for the Calculation of Uncertainties Through Functional Relationships, Including Uncertainties in Empirically Derived Constants. *The Clinical Biochemist Review, 35*(1), 37.

Fiegl, C., & Pontow, C. (2009). Online Scheduling of Pick-Up and Delivery Tasks in Hospitals. *Journal of Biomedical Informatics, 42*, 624-632.

Hanne, T., Melo, T., & Nickel, S. (2009). Bringing Robustness to Patient Flow Management Through Optimized Patient Transports in Hospitals. *INFORMS Journal on Applied Analytics, 39*(3), 241-255.

Hans, E. W., Van Houdenhoven, M., & Hulshof, P. J. (2012). A Framework for Healthcare Planning and Control. In R. Hall, *Handbook of Healthcare System Scheduling.* Springer.

Hans, Houdenhoven, & Hulshof. (2012). A Framework for Healthcare Planning and Control. *Handbook of Healthcare System Scheduling*, 303-320.

Heerkens, J., & Van Winden, A. (2012). *Geen Probleem.* Nieuwegein: Van Winden Comunicatie.

Hribar, M., Read-Brown, S., Reznick, L., & Chiang, M. (2017). Evaluating and Improving an Outpatient Clinic Scheduling Template Using Secondary Electronic Health Record Data. *AMIA Annual Symposium Proceedings Archive*, 921-929.

Leeftink, G., & Hans, E. (2018). Case mix classification and a benchmark set for surgery scheduling. *Journal of Scheduling, 21*(1), 17-33. Retrieved from https://doi.org/10.1007/s10951-017-0539-8

Lemos, P., Jarrett, P., & Philip, B. (2006). Chapter 1: The development of ambulatory surgery and future challenges. In P. Lemos, P. Jarrett, & B. Philip, *Day Surgery: Development and Practice* (pp. 21-34). London: International Association for Ambulatory Surgery.

Mes, M. (2018). Lecture 5: Experimental setup. Enschede: University of Twente.

Peters, F. (2015). *Deviating Trends in Dutch Life Expectancy: Explanation and Projection.* Rotterdam: Optima Grafische Communicatie.

Robinson, S. (2014). *Simulation: The Practice of Model Development and Use.* Hampshire: Palgrave Macmillan.

Saremi, A., Jula, P., Elmekkawy, T., & Wang, G. (2013). Appointment Scheduling of Outpatient Surgical Services in a Multistage Operating Room Department. *International Journal of Production Economics, 141*(2), 646-658.

Sturges, H. (1926). The Choice of a Class Interval. *Journal of the American Statistical Association*, 65-66.

Appendix A

	OK ROOSTER even weken.LET OP!!! 09:00uur start				
	Maandag	Dinsdag	Woensdag	Donderdag	Vrijdag
OK 1	SPOED	SPOED	SPOED	SPOED	SPOED
OK 2 Hybride	CHI 08:00 - 16:30	ONG 08:00 - 16:30	CHI 08:00 - 16:30	NEC 08:00 - 16:30	CHI 08:00 - 16:30
OK 3	ONG 08:00 - 16:30	CHI 08:00 - 16:30	CHI 08:00 - 17:30	ONG 08:00 - 17:30	ONG 08:00 - 16:30
OK 4 NEC	Nec 08:00 - 16:30	Nec 08:00 - 17:30	Nec 08:00 - 17:30	Nec 08:00 - 17:30	Nec 08:00 - 16:30
OK 5 CHIR video	CHI 08:00 - 15:30	CHI 08:00 - 15:30	CHI 08:00 - 15:30	CHI 08:00 - 15:30	CHI 08:00 - 17:30
OK 6 IOBT	CHI 08:00 - 16.30	Rth tot 11:45 Chir tot 16:30uur.	Nec 08:00 - 16:30	Rth 08:00 - 15:30	Nec 08:00 - 16:30
OK 7 laser/video	CHI 08:00 - 20.00	CHI 08:00 - 20:00	CHI 08:00 - 16:30	CHI 08:00 - 20:00	CHI 08:00 - 16:30
OK 8 KNO	KNO 08:00 - 16.30	KNO 08:00 - 16.30	KNO 08:00 - 16:30	KNO 08:00 - 16:30	KNO 08:00 - 20:00
OK 9 laser	Nec 08:00 - 17:30	CHI 08:00 - 16:30	CHI 08:00 - 16:30	CHI 08:00 - 16:30	CHI 08:00 - 16:30
OK 10 laser	KNO 08:00 - 16:30	KNO 08:00 - 16.30	KNO 08:00 - 16:30	KNO 08:00 - 16:30	KNO 08:00 - 16:30
OK 11	GYN 08:00 - 16:30	PLC 08:00 - 16:30	GON 08:00 - 16:30	GYN 08:00 - 16:30	GON 08:00 - 16.30
OK 12 ORTHO	Ortho 08:00 - 17:30	Ortho 08:00 - 17:30	Ortho 08:00 - 16:30	Ortho 08:00 - 16:30	Ortho 08:00 - 17:30
OK 13 Da Vinci	Chirurgie 08:00 - 16.30	URO 08:00 - 16:30	URO 08:00 - 16:30	CHI 08:00 - 16:30	Uro 08:00 - 16:30
OK 14 zutkamer video	PLC 08:00 - 16:30	Ortho 08:00 - 16:30	Ortho 08:00 - 16:30	GON 08:00 - 16:30	Ortho 08:00 - 16:30
OK 15 video	PLC 08:00 - 16:30	PLC 08:00 - 16:30	PLC 08:00 - 20:00	PLC 08:00 - 16:30	Plas Chir 08:00 - 16:30
OK 16 OOG laser	OOG, student+anm 08:00 - 15:30	OOG, student 08:00 - 15:30	PLC 08:00 - 16:30	OOG 08:00 - 15:30	OOG, student_anm 08:00 - 15:30
OK 17	URO 08:00-16:30	KAA 08:00 - 16:30	KAA 08:00 - 20.00	PLC of KAAK of ORTH 08:00 - 16:30	KAA 08:00 - 16:30
OK 18 thorax	Pijn, sed 08:00 - 15:30	THORAX 08:00 - 16:30	THORAX 08:00 - 16:30	THORAX 08:00 - 16:30	PLC 08:00 - 16:30
OK 19 thorax	THORAX 08:00 - 16:30	THORAX 08:00 - 16:30	THORAX 08:00 - 16:30	THORAX 08:00 - 16:30	THORAX 08:00 - 16:30
OK 20 thorax	THORAX 08:00 - 18:30	THORAX 08:00 - 18:30	THORAX 08:00 - 18:30	THORAX 08:00 - 18:30	THORAX 08:00 - 18:30
OK 21 thorax	THORAX 08:00 - 16:30	THORAX 08:00 - 18:30	THORAX 08:00 - 18:30	THORAX 08:00 - 18:30	THORAX 08:00 - 18:30
OK 26 Hybride			Pijn, sed 08:00 - 15:30		THORAX 08:00 - 16:30
OK C	ESWL, sed 08:00 - 15:30	GYN 08:00 - 15:30	URO, 08:00 - 15:30	Derma, alg 08:00 - 15:30	Pijn, geen personeel 08:00 - 11:45
OK D laser	URO 08:00 - 15:30	Derma, sed 08:00 - 15:30	URO, 08:00 - 15:30	KAA 08:00 - 14:00	ONG 08:00 - 15:30

Figure 26: Even week of the blueprint

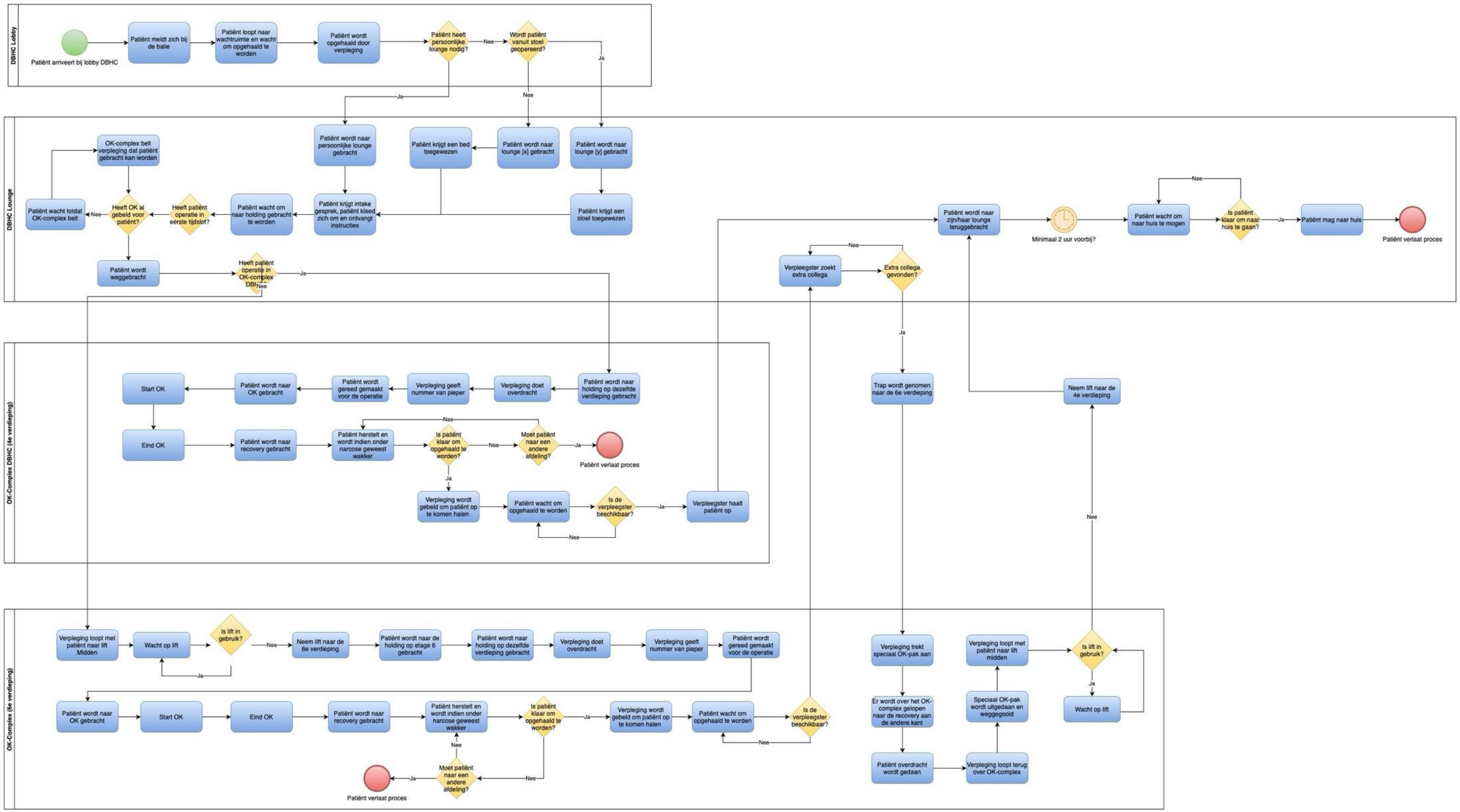


Figure 27: Flowchart diagram of the process of day care surgery

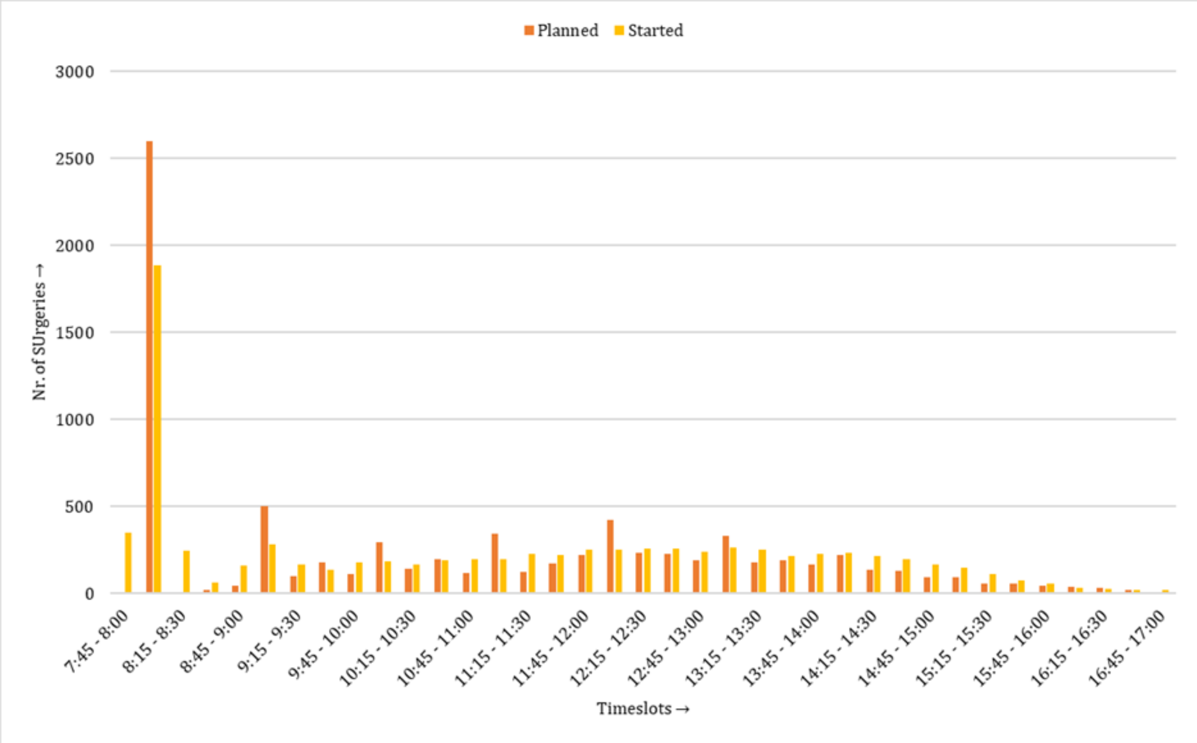


Figure 28: Number of clinical surgeries starting per timeslot in 2019

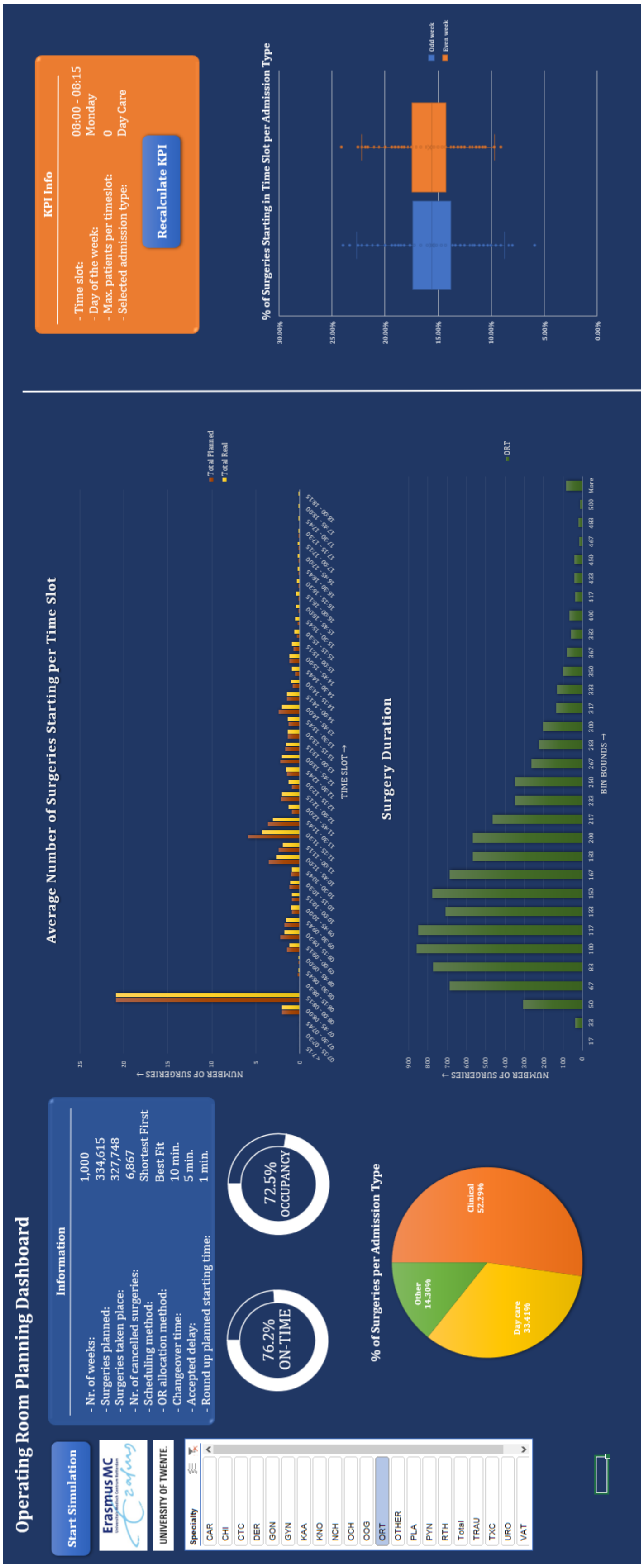
Specialty	1		2		3	
	Amount	Sequence	Amount	Sequence	Amount	Sequence
CAR	36	LLL	32	L	23	LL
CHI	185	L	174	LL	71	LLL
CTC	815	LL	360	L	66	LLL
DER	13	L	3	M	3	MMMSM
GON	31	L	24	LL	13	ML
GYN	29	LL	21	LLL	7	LML
KAA	53	L	39	LLL	31	LL
KNO	87	LL	60	LLL	46	L
NCH	245	LL	136	LLL	88	L
OCH	61	L	48	LL	26	ML
OOG	6	MMMMMM	5	MML	5	MMM
ORT	136	LL	130	LLL	31	L
PLA	104	L	55	LL	43	LLL
PYN	12	SSSSSS	12	SSSSSSS	8	SSSSSSSS
RTH	39	L	12	M	12	MM
TRAU	53	LLL	45	LL	23	L
TXC	85	L	82	LL	48	LLL
URO	67	LL	52	L	25	LLL
VAT	99	LL	26	L	20	LLL
Other	1983	LL	1264	L	779	LLL

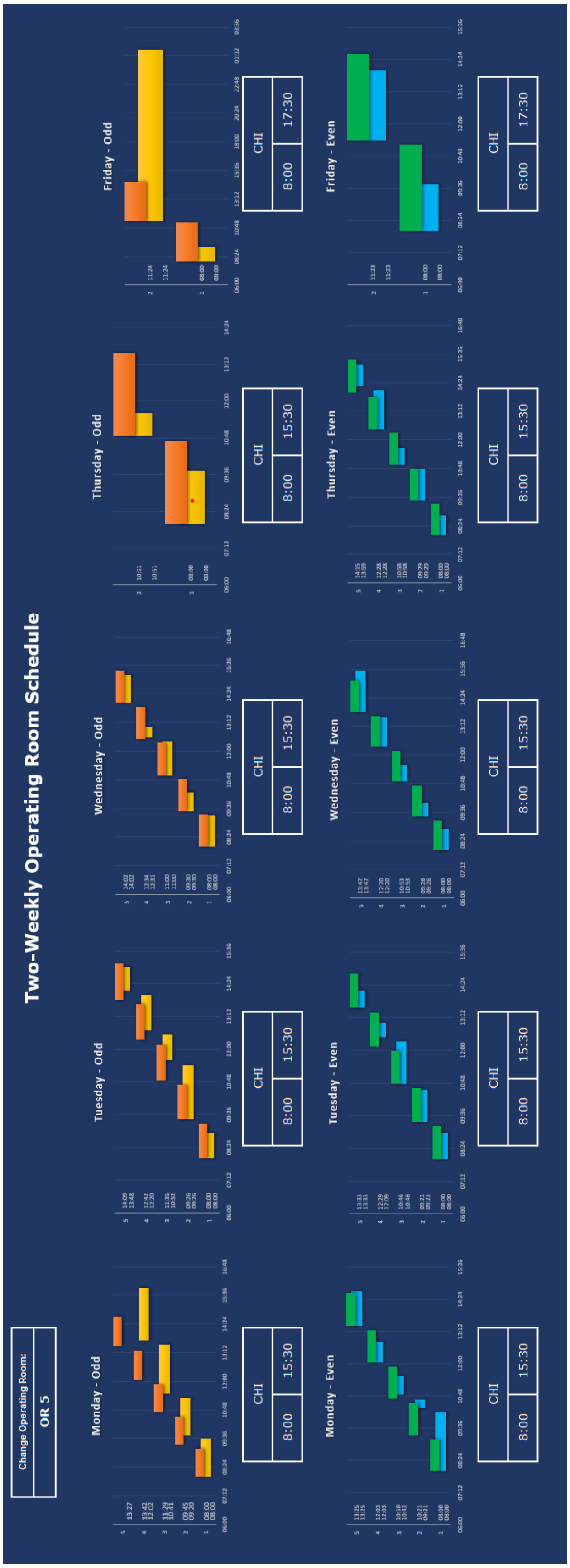
Table 7: Most occurring surgery duration sequences

Specialty	[08:00, 11:00)	[11:00, 14:00)	[14:00, 17:00)	[17:00, 20:00]
CAR	2	4	1	3
CHI	1	2	3	4
CTC	1	2	4	3
DER	1	3	2	Not Planned
GON	1	2	3	Not Planned
GYN	1	2	3	Not Planned
KAA	1	2	3	4
KNO	1	2	3	4
NCH	1	2	4	3
OCH	1	2	3	4
OOG	2	4	3	1
ORT	1	2	3	4
PLA	2	3	4	1
PYN	3	1	2	Not Planned
RTH	1	3	2	Not Planned
TRAU	2	1	4	3
TXC	1	2	3	4
URO	1	3	4	2
VAT	1	2	3	4
Other	1	2	4	3

Table 8: Surgery duration per timeslot

Appendix B: The Operating Room Planning Dashboard





Appendix C: Algorithm VBA Code

```
Option Explicit

Type TSurgery
    SessionNr As Long
    PlannedTime As Long
    Admission As String
    OT As String
    Specialty As String
    PlannedStart As Long
    Duration As Long
    TimeGradation As String
    Result As String
End Type

Type TSessionArray
    SessionNr As Long
    Specialty As String
    TimeGradation As String
    AdmissionResult As String
    Result As String
    NrSurgeries As Long
    Surgery() As TSurgery
End Type

Public SessionArray() As TSessionArray
Public nrSessions As Long
```

```
Sub GetPlanningMethod()

    Dim i, j, k, l As Long
    Dim numSurgeries As Long
    Dim lastRow As Long
    Dim rowCounter As Long
    Dim inputSheet, outputSheet, SurgeryDur As Worksheet
    Dim SessionNr As Long
    Dim shortLength, middleLength, longLength As Long
    Dim dummyL As Long
    Dim dummyS As String
    Dim SpecialtyArray(1 To 20, 1 To 2) As Variant


    Set inputSheet = ThisWorkbook.Sheets("Input")
    Set outputSheet = ThisWorkbook.Sheets("Output2")
    Set SurgeryDur = ThisWorkbook.Sheets("Surgery Duration")
    lastRow = inputSheet.Cells(inputSheet.Rows.Count, "A").End(xlUp).Row

    For i = 1 To 20
        SpecialtyArray(i, 1) = SurgeryDur.Cells(i, 1)
        SpecialtyArray(i, 2) = SurgeryDur.Cells(i, 2)
    Next i

    rowCounter = 2

    For i = 1 To lastRow - 1

        ReDim Preserve SessionArray(1 To i)

        If rowCounter > lastRow Then
            GoTo printLine
        End If

        'Find session number
        SessionNr = inputSheet.Cells(rowCounter, 1).Value

        'Check how much identical session numbers follow
        numSurgeries = 0
        Do Until inputSheet.Cells(rowCounter + numSurgeries, 1) <> SessionNr
            numSurgeries = numSurgeries + 1
        Loop

        SessionArray(i).NrSurgeries = numSurgeries

        'Create array with length j
        ReDim SessionArray(i).Surgery(1 To numSurgeries)
```

```
        For k = 1 To numSurgeries
            SessionArray(i).Surgery(k).SessionNr = inputSheet.Cells(rowCounter - 1 + k, 1).Value
            SessionArray(i).Surgery(k).Admission = inputSheet.Cells(rowCounter - 1 + k, 2).Value
            SessionArray(i).Surgery(k).Specialty = inputSheet.Cells(rowCounter - 1 + k, 4).Value
            SessionArray(i).Surgery(k).PlannedStart = inputSheet.Cells(rowCounter - 1 + k, 5).Value
            SessionArray(i).Surgery(k).Duration = inputSheet.Cells(rowCounter - 1 + k, 6).Value

            'Check specialty and find corresponding average duration of specialty
            For j = 1 To 20
                  If SpecialtyArray(j, 1) = SessionArray(i).Surgery(k).Specialty Then
                    'If shorter than mean give S
                    If SessionArray(i).Surgery(k).Duration <= SpecialtyArray(j, 2) Then
                        SessionArray(i).Surgery(k).TimeGradation = "S"
                        SessionArray(i).Surgery(k).Result = "S" & SessionArray(i).Surgery(k).Admission
                        Exit For
                    'If longer than mean give L
                    ElseIf SessionArray(i).Surgery(k).Duration > SpecialtyArray(j, 2) Then
                        SessionArray(i).Surgery(k).TimeGradation = "L"
                        SessionArray(i).Surgery(k).Result = "L" & SessionArray(i).Surgery(k).Admission
                        Exit For
                    End If
                End If
            Next j
        Next k

        rowCounter = rowCounter + numSurgeries

        'Bubble sort surgeries on start time
        For j = 1 To numSurgeries
            For k = j + 1 To numSurgeries
                If SessionArray(i).Surgery(j).PlannedStart > SessionArray(i).Surgery(k).PlannedStart Then

                    dummyL = SessionArray(i).Surgery(j).PlannedStart
                    SessionArray(i).Surgery(j).PlannedStart = SessionArray(i).Surgery(k).PlannedStart
                    SessionArray(i).Surgery(k).PlannedStart = dummyL

                    dummyL = SessionArray(i).Surgery(j).SessionNr
                    SessionArray(i).Surgery(j).SessionNr = SessionArray(i).Surgery(k).SessionNr
                    SessionArray(i).Surgery(k).SessionNr = dummyL

                    dummyS = SessionArray(i).Surgery(j).Admission
                    SessionArray(i).Surgery(j).Admission = SessionArray(i).Surgery(k).Admission
                    SessionArray(i).Surgery(k).Admission = dummyS

                    dummyS = SessionArray(i).Surgery(j).Specialty
                    SessionArray(i).Surgery(j).Specialty = SessionArray(i).Surgery(k).Specialty
                    SessionArray(i).Surgery(k).Specialty = dummyS

                    dummyL = SessionArray(i).Surgery(j).Duration
                    SessionArray(i).Surgery(j).Duration = SessionArray(i).Surgery(k).Duration
                    SessionArray(i).Surgery(k).Duration = dummyL

                    dummyS = SessionArray(i).Surgery(j).Result
                    SessionArray(i).Surgery(j).Result = SessionArray(i).Surgery(k).Result
                    SessionArray(i).Surgery(k).Result = dummyS

                    dummyS = SessionArray(i).Surgery(j).TimeGradation
                    SessionArray(i).Surgery(j).TimeGradation = SessionArray(i).Surgery(k).TimeGradation
                    SessionArray(i).Surgery(k).TimeGradation = dummyS

                End If
            Next k
        Next j

        SessionArray(i).SessionNr = SessionArray(i).Surgery(1).SessionNr
        SessionArray(i).Specialty = SessionArray(i).Surgery(1).Specialty

        For j = 1 To numSurgeries
            SessionArray(i).TimeGradation = SessionArray(i).TimeGradation & SessionArray(i).Surgery(j).TimeGradation
            SessionArray(i).AdmissionResult = SessionArray(i).AdmissionResult & SessionArray(i).Surgery(j).Admission
            SessionArray(i).Result = SessionArray(i).Result & SessionArray(i).Surgery(j).Result
        Next j

    Next i

printLine:
    nrSessions = i

    Call WriteSessionData

End Sub
```

```
Sub WriteSessionData()

Application.ScreenUpdating = False
Application.Calculation = xlCalculationManual

    Dim i, j As Long
    Dim lastRow As Long
    Dim inputSheet, outputSheet As Worksheet


    Set inputSheet = ThisWorkbook.Sheets("Input")
    Set outputSheet = ThisWorkbook.Sheets("Output2")
    lastRow = inputSheet.Cells(inputSheet.Rows.Count, "A").End(xlUp).Row

    For i = 1 To nrSessions
        outputSheet.Cells(1 + i, 1).Value = SessionArray(i).SessionNr
        outputSheet.Cells(1 + i, 2).Value = SessionArray(i).Specialty
        outputSheet.Cells(1 + i, 3).Value = SessionArray(i).TimeGradation
        outputSheet.Cells(1 + i, 4).Value = SessionArray(i).AdmissionResult
        outputSheet.Cells(1 + i, 5).Value = SessionArray(i).Result

        For j = 1 To SessionArray(i).NrSurgeries
            outputSheet.Cells(1 + i, 5 + j).Value = SessionArray(i).Surgery(j).Result
        Next j
    Next i

Application.ScreenUpdating = True
Application.Calculation = xlCalculationAutomatic


End Sub
```

```
Sub CheckNextLetter()

    Application.ScreenUpdating = False
    Application.Calculation = xlCalculationManual

    Dim inputSheet, outputSheet As Worksheet
    Dim lastRow As Long
    Dim i, j As Long
    Dim myString As String
    Dim myCounter() As Long
    Dim LCounter As Long
    Dim MCounter As Long
    Dim SCounter As Long

    Set inputSheet = ThisWorkbook.Sheets("Sheet4")
    Set outputSheet = ThisWorkbook.Sheets("Output")
    lastRow = inputSheet.Cells(inputSheet.Rows.Count, "C").End(xlUp).Row

    ReDim myCounter(1 To lastRow, 1 To 13)

    For i = 1 To lastRow
        myString = inputSheet.Cells(1 + i, 3).Value

        LCounter = 0
        MCounter = 0
        SCounter = 0
        For j = 1 To Len(myString)
            If Mid(myString, j, 1) = "L" Then
                LCounter = LCounter + 1
                If Mid(myString, j + 1, 1) = "L" Then
                    myCounter(i, 1) = myCounter(i, 1) + 1
                ElseIf Mid(myString, j + 1, 1) = "M" Then
                    myCounter(i, 2) = myCounter(i, 2) + 1
                ElseIf Mid(myString, j + 1, 1) = "S" Then
                    myCounter(i, 3) = myCounter(i, 3) + 1
                End If
            ElseIf Mid(myString, j, 1) = "M" Then
                MCounter = MCounter + 1
                If Mid(myString, j + 1, 1) = "L" Then
                    myCounter(i, 4) = myCounter(i, 4) + 1
                ElseIf Mid(myString, j + 1, 1) = "M" Then
                    myCounter(i, 5) = myCounter(i, 5) + 1
                ElseIf Mid(myString, j + 1, 1) = "S" Then
                    myCounter(i, 6) = myCounter(i, 6) + 1
                End If
            ElseIf Mid(myString, j, 1) = "S" Then
                SCounter = SCounter + 1
                If Mid(myString, j + 1, 1) = "L" Then
                    myCounter(i, 7) = myCounter(i, 7) + 1
                ElseIf Mid(myString, j + 1, 1) = "M" Then
                    myCounter(i, 8) = myCounter(i, 8) + 1
                ElseIf Mid(myString, j + 1, 1) = "S" Then
                    myCounter(i, 9) = myCounter(i, 9) + 1
                End If
            End If
        Next j
        myCounter(i, 11) = LCounter
        myCounter(i, 12) = MCounter
        myCounter(i, 13) = SCounter
    Next i

    For i = 1 To lastRow
        For j = 1 To 13
            inputSheet.Cells(1 + i, 3 + j).Value = myCounter(i, j)
        Next j
    Next i
Application.ScreenUpdating = True
Application.Calculation = xlCalculationAutomatic

End Sub
```

Appendix D: Dashboard VBA Code

```
Option Explicit
Option Base 1

Const NumDaysPerWeek = 5
Const NumORs = 24
Const NumSpecialties = 20

Type TSurgeryType
    Exp As Double
    stdDev As Double
    ExpDiff As Double 'Expected difference in realised time
    stdDevDiff As Double 'Standard deviation difference in
    SurgTypeID As Long
    SpecialtyID As Long
    Percentage As Double '% of all the surgery types of this specialty
    AdmissionType As String
End Type

Type TSurgery
    Exp As Double
    stdDev As Double
    SurgTypeID As Long
    SpecialtyID As Long
    SpecialtyName As String
    AdmissionType As String
    Realization As Double 'Simulated duration
    SchedStartTime As Double 'Scheduled start time
    SchedEndTime As Double 'Scheduled end time
    RealStartTime As Double 'Realized start time
    RealEndTime As Double 'Realized end time
    SurgeryPerformed As Boolean
End Type

Type TORSchedule
    Schedule() As TSurgery
    Cancellations() As TSurgery
    NumSurg As Long
    SessionStart As Double
    SessionEnd As Double
    sessionID As String
    PlannedEndTime As Double 'earliest available time for OR to plan a new surgery
    RealEndTime As Double
    occupancy As Double
End Type

Type TOR
    ORnr As Long
    SpecialtyID As Long
    SpecialtyName As String
    ORschedule As TORSchedule
End Type

Type TDay
    ORs() As TOR
    DayName As String
End Type

Type TSurgeryList
    surg() As TSurgery
    NumSurg As Long
End Type

Type TSpecialty
    SpecialtyID As Long
    SpecialtyName As String
    Percentage As Double
    SurgeryTypes() As TSurgeryType
    Weeks() As TSurgeryList 'contains lists of their patients
End Type

Type TData
    specialty() As TSpecialty
End Type

Public Days() As TDay 'contains schedules for every ORday
Public Data As TData
Public NumWeeks As Long
Public NumDays As Long

'These variables are used for the userform options
Public NrRuns As Long
Public ChangeOverTime As Long
Public AcceptedDelay As Long
Public PatientCancellation As Boolean
Public PlanningSF As Boolean
Public PlanningLF As Boolean
Public PlanningRand As Boolean
Public PatientInfo As Boolean
Public TimeSlotNr As Long
Public PatientsPerTimeslotPreferred As Long
Public MinRoundUpPlanningTime As Long
Public WeekDay As String
Public shSurgeriesStartingPerTimeslot As Worksheet
Public shDashboardMeasurements As Worksheet
Public useSeedValues As Boolean
Public oRSchedulingBestFit As Boolean
Public KPIAdmissionType As String
```

```
Sub ShowStartSimulationForm()
'This sub opens a userform where the user can enter the simulation details
    ufStartSimulation.Show
End Sub
```

```
Sub StartSimulation()

    Application.Calculation = xlCalculationManual
    Application.ScreenUpdating = False
    Application.EnableEvents = False

    'Store the given control variables in the public variables
    Call StoreVariables

    'Load all input data in the memory and generate patients
    Call InitData

    'Schedule the surgery lists
    Call ListScheduling

    'Create the realized schedule
    Call RealizeSchedule(PatientCancellation)

    'Write the scheduled patients in the schedule patients worksheet
    Call WriteScheduledPatients

    'Use the first scheduled surgery as example in dashboard
    Call CreateSchedulingExample

    'Calculate the percentage of patients in a specified time slot in a week
    Call CalculateKPI(TimeSlotNr, WeekDay, PatientsPerTimeslotPreferred)

    'Calculate the number of surgeries starting per timeslot
    Call CalculateSurgeriesStartingPerTimeslot

    'Calculate the number of surgeries that are on time
    Call CalculateSurgeriesOnTime

    'Get occupancy of OR sessions with or without writing down all the information
    Call CalculateOccupancy

    'Calculate the duration of every specialty
    Call CalculateSurgeryDuration

    'Give info that simulation is done
    MsgBox "Simulation is done!", vbInformation

    Application.Calculation = xlCalculationAutomatic
    Application.ScreenUpdating = True
    Application.EnableEvents = True

End Sub
```

```
Public Sub StoreVariables()
'This sub stores the values of the userform in global variables and calculates values for the _
dashboard

    Dim shControls As Worksheet

    Set shControls = ThisWorkbook.Sheets("Controls")
    Set shDashboardMeasurements = ThisWorkbook.Sheets("Dashboard_Measurements")

    With shControls
         NrRuns = .Cells(2, 2)
         ChangeOverTime = .Cells(3, 2)
         AcceptedDelay = .Cells(4, 2)
         PatientCancellation = .Cells(5, 2)
         TimeSlotNr = .Cells(6, 3)
         PatientInfo = .Cells(7, 2)
         PatientsPerTimeslotPreferred = .Cells(8, 2)
         WeekDay = .Cells(9, 2)
         PlanningSF = .Cells(10, 2)
         PlanningLF = .Cells(11, 2)
         PlanningRand = .Cells(12, 2)
         MinRoundUpPlanningTime = .Cells(13, 2)
         useSeedValues = .Cells(14, 2)
         oRSchedulingBestFit = .Cells(15, 2)
         KPIAdmissionType = .Cells(17, 2)
    End With

    If PlanningSF Then
        shDashboardMeasurements.Cells(9, 2).Value = "Shortest First"
    ElseIf PlanningLF Then
        shDashboardMeasurements.Cells(9, 2).Value = "Longest First"
    ElseIf PlanningRand Then
        shDashboardMeasurements.Cells(9, 2).Value = "Random"
    End If

    If oRSchedulingBestFit Then
        shDashboardMeasurements.Cells(10, 2).Value = "Best Fit"
    Else
        shDashboardMeasurements.Cells(10, 2).Value = "First Fit"
    End If

    shDashboardMeasurements.Cells(2, 2).Value = NrRuns * 2
    shDashboardMeasurements.Cells(11, 2).Value = ChangeOverTime & " min."
    shDashboardMeasurements.Cells(12, 2).Value = AcceptedDelay & " min."
    shDashboardMeasurements.Cells(13, 2).Value = MinRoundUpPlanningTime & " min."

End Sub
```

```
Sub InitData()
'Load all input data in the memory and generate patients

    Dim i As Long
    Dim d As Long
    Dim s As Long
    Dim w As Long
    Dim DayOfWeek As Long
    Dim colPatientsPerWeek As Long
    Dim NumSurgeryTypes As Long
    Dim lastCol As Long
    Dim DayWeek As String
    Dim timeSlot As String
    Dim shSurgeryTypes As Worksheet
    Dim shPatientsPerWeek As Worksheet
    Dim shScheduledPatients As Worksheet
    Dim shBluePrint As Worksheet

    'Define worksheet variables
    Set shSurgeryTypes = ThisWorkbook.Sheets("SurgeryTypes")
    Set shScheduledPatients = ThisWorkbook.Sheets("Scheduled_Patients")
    Set shPatientsPerWeek = ThisWorkbook.Sheets("Generated_Patients_per_Week")
    Set shDashboardMeasurements = ThisWorkbook.Sheets("Dashboard_Measurements")
    Set shBluePrint = ThisWorkbook.Sheets("Schedule")

    'Check if we need to perform the simulation with the same random values
    If useSeedValues Then
        'Initialize rnd function with new seed value to make sure the same random numbers are used
        Rnd (-1)
        Randomize 10
    End If

    'Calculate the number of weeks and number of days
    NumWeeks = NrRuns * 2
    NumDays = NumWeeks * NumDaysPerWeek

    'Initialize specialties
    ReDim Data.specialty(1 To NumSpecialties)

    'Initialize the weekly surgery lists for every specialty
    For i = 1 To NumSpecialties
        ReDim Data.specialty(i).Weeks(1 To NumWeeks) As TSurgeryList
    Next i

    'Initialize the number of days
    ReDim Days(1 To NumDays)

    'Initialize the data for every day
    For d = 1 To NumDays
        'Determine day of week and the corresponding name
        DayOfWeek = d - (((d - 1) \ 5) * 5)
        If DayOfWeek = 1 Then
            DayWeek = "Monday"
        ElseIf DayOfWeek = 2 Then
            DayWeek = "Tuesday"
        ElseIf DayOfWeek = 3 Then
            DayWeek = "Wednesday"
        ElseIf DayOfWeek = 4 Then
            DayWeek = "Thursday"
        ElseIf DayOfWeek = 5 Then
            DayWeek = "Friday"
        End If
        Days(d).DayName = DayWeek

        'Define the number of ORs for every day
        ReDim Days(d).ORs(1 To NumORs)

        'Give every OR their daily specifications
        For i = 1 To NumORs
            'Give OR number to daily OR
            Days(d).ORs(i).ORnr = i
            'Get session information in case of an odd week
            If Not (1 + ((d - 1) \ 5)) Mod 2 = 0 Then
                'Use left schedule in schedule tab to obtain specialty numbers
                Days(d).ORs(i).SpecialtyID = shBluePrint.Cells(1 + i, DayOfWeek * 3).Value
                'Find session start & end
                Days(d).ORs(i).ORschedule.SessionStart = shBluePrint.Cells(1 + i, 1 + DayOfWeek * 3).Value
                Days(d).ORs(i).ORschedule.SessionEnd = shBluePrint.Cells(1 + i, 2 + DayOfWeek * 3).Value
                Days(d).ORs(i).ORschedule.sessionID = "SO" & 10 * (d & i)
                'No surgeries are planned in the OR yet
                Days(d).ORs(i).ORschedule.PlannedEndTime = Days(d).ORs(i).ORschedule.SessionStart
            'Get session information in case of an even week
            ElseIf (1 + ((d - 1) \ 5)) Mod 2 = 0 Then 'Even weeks
                'Use right schedule in schedule tab to obtain specialty numbers
                Days(d).ORs(i).SpecialtyID = shBluePrint.Cells(1 + i, 17 + DayOfWeek * 3).Value
                'Find session start & end
                Days(d).ORs(i).ORschedule.SessionStart = shBluePrint.Cells(1 + i, 18 + DayOfWeek * 3).Value
                Days(d).ORs(i).ORschedule.SessionEnd = shBluePrint.Cells(1 + i, 19 + DayOfWeek * 3).Value
                Days(d).ORs(i).ORschedule.sessionID = "SE" & 10 * (d & i)
                'No surgeries are planned in the OR yet
                Days(d).ORs(i).ORschedule.PlannedEndTime = Days(d).ORs(i).ORschedule.SessionStart
            End If
        Next i
    Next d
```

```
    'Initialize the data for every specialty
    For s = 1 To NumSpecialties
        Data.specialty(s).SpecialtyID = s
        Data.specialty(s).SpecialtyName = Sheets("Specialties").Cells(s + 1, 2)
        Data.specialty(s).Percentage = Sheets("Specialties").Cells(s + 1, 3)

        'Initialize the counters for the specialties and number of surgery types
        i = 2
        NumSurgeryTypes = 0

        'Loop through specialty list
        Do While Not IsEmpty(shSurgeryTypes.Cells(i, 1).Value)
            'Add surgery type to surgery type list of this specialty
            If shSurgeryTypes.Cells(i, 1) = s Then
                NumSurgeryTypes = NumSurgeryTypes + 1
                ReDim Preserve Data.specialty(s).SurgeryTypes(1 To NumSurgeryTypes)
                'Init surgery type data per specialty
                With Data.specialty(s).SurgeryTypes(NumSurgeryTypes)
                    .SpecialtyID = s
                    .SurgTypeID = 1000 * s + NumSurgeryTypes 'unique ID
                    .AdmissionType = shSurgeryTypes.Cells(i, 7)
                    .Exp = shSurgeryTypes.Cells(i, 5)
                    .stdDev = shSurgeryTypes.Cells(i, 6)
                    .Percentage = shSurgeryTypes.Cells(i, 4) 'Relative frequency
                End With
            End If
            i = i + 1
        Loop
    Next s

    'Generate patients per specialty per week
    For w = 1 To NumWeeks
        'In even weeks take column 6 for mean patients per week
        If w Mod 2 = 0 Then
            colPatientsPerWeek = 6
        'In odd weeks take column 4 for mean patients per week
        Else
            colPatientsPerWeek = 4
        End If
        'Generate the patients for every specialty s in week w
        For s = 1 To NumSpecialties
            Call GeneratePatients(w, s, Round(shPatientsPerWeek.Cells(3 + s, colPatientsPerWeek).Value, 0))
        Next s
    Next w

End Sub
```

```
Sub GeneratePatients(w As Long, s As Long, Amount As Long)
'Generates a number of patients for specialty s in week w with the right properties

    Dim p As Long
    Dim NumSurgeries As Long
    Dim st As Long
    Dim randomnr As Double
    Dim Total As Double
    Dim mean As Double
    Dim stdDev As Double
    Dim ScaledMean As Double
    Dim ScaledStdDev As Double
    Dim realizationTime As Double

    'Generate "Amount" of patients
    For p = 1 To Amount

         'Draw a random surgery type "st" by using the random function of Excel
         Total = 0
         st = 0
         randomnr = Rnd()
         'Let Total be equal or less than the drawn random number since the random function _
         such that a surgery type is drawn even when the random number is equal to zero
         Do While Total <= randomnr
            st = st + 1
            Total = Total + Data.specialty(s).SurgeryTypes(st).Percentage
         Loop

         'Increase the counter for number of surgeries
         NumSurgeries = NumSurgeries + 1
         'Add one surgery to surgery list for specialty s in week w
         ReDim Preserve Data.specialty(s).Weeks(w).surg(1 To NumSurgeries)

         'Declare mean and standard deviation variables
         mean = Data.specialty(s).SurgeryTypes(st).Exp
         stdDev = Data.specialty(s).SurgeryTypes(st).stdDev

         'Create scaled mean and standard deviation
         ScaledMean = Application.WorksheetFunction.Ln(mean ^ 2 / Sqr(mean ^ 2 + stdDev ^ 2))
         ScaledStdDev = Sqr(Application.WorksheetFunction.Ln((mean ^ 2 + stdDev ^ 2) / mean ^ 2))

         With Data.specialty(s).Weeks(w).surg(NumSurgeries)
            'Create the expected length of the surgery based on the surgery type
            .Exp = mean + CLng((15 - (-15 + 1) * Rnd + (-15)))
            .stdDev = stdDev 'Standard deviation surgery length
            .SurgTypeID = Data.specialty(s).SurgeryTypes(st).SurgTypeID 'Surgery type ID
            .SpecialtyID = s 'Specialty ID
            .SpecialtyName = Data.specialty(s).SpecialtyName
            .AdmissionType = Data.specialty(s).SurgeryTypes(st).AdmissionType
            'Set number of patients per week
            Data.specialty(s).Weeks(w).NumSurg = NumSurgeries
            'Use the scaled mean and scaled standard deviation to draw random numbers from a log-normal distribution
            .Realization = Round(Exp(Application.WorksheetFunction.NormInv(Rnd(), ScaledMean, ScaledStdDev)), 0)
         End With
    Next p

End Sub
```

```
Sub ListScheduling()
'Schedules the patients in ORs

    Dim w As Long
    Dim s As Long
    Dim p As Long
    Dim DayNr As Long
    Dim ORnr As Long
    Dim numPatients As Long 'Counter for nr of patients per OR per day
    Dim AvailableOR As Collection

    'Loop through all weeks
    For w = 1 To NumWeeks
        'Loop through all specialties
        For s = 1 To NumSpecialties

            'Sort patients in a convenient sequence
            If PlanningSF Then
                Call BubbleSortSurgeryListOnDuration(w, s, True)
            ElseIf PlanningLF Then
                Call BubbleSortSurgeryListOnDuration(w, s, False)
            ElseIf PlanningRand Then
                Call RandomSortSurgeryList(w, s)
            End If

            'Loop through all patients for this week
            For p = 1 To Data.specialty(s).Weeks(w).NumSurg

                'Plan each patient to the first fit or best fit OR
                If oRSchedulingBestFit Then
                    Set AvailableOR = FindORBestFit(w, s, p)
                Else
                    Set AvailableOR = FindORFirstFit(w, s, p)
                End If

                DayNr = AvailableOR.Item(1)
                ORnr = AvailableOR.Item(2)

                'If there is an available OR, plan surgery in OR
                If Not DayNr = 0 Or Not ORnr = 0 Then
                    Days(DayNr).ORs(ORnr).ORschedule.NumSurg = Days(DayNr).ORs(ORnr).ORschedule.NumSurg + 1
                    numPatients = Days(DayNr).ORs(ORnr).ORschedule.NumSurg

                    'Insert surgery data
                    ReDim Preserve Days(DayNr).ORs(ORnr).ORschedule.Schedule(1 To numPatients)

                    With Days(DayNr).ORs(ORnr).ORschedule.Schedule(numPatients)
                        'Insert the general data per patient in the schedule
                        .Exp = Data.specialty(s).Weeks(w).surg(p).Exp
                        .SpecialtyID = s
                        .SpecialtyName = Data.specialty(s).Weeks(w).surg(p).SpecialtyName
                        .stdDev = Data.specialty(s).Weeks(w).surg(p).stdDev
                        .SurgTypeID = Data.specialty(s).Weeks(w).surg(p).SurgTypeID
                        .AdmissionType = Data.specialty(s).Weeks(w).surg(p).AdmissionType

                        'Determine the scheduled starting time and round up to the selected amount of minutes
                        .SchedStartTime = Application.WorksheetFunction.RoundUp(Days(DayNr).ORs(ORnr).ORschedule. _
                        PlannedEndTime / MinRoundUpPlanningTime, 0) * MinRoundUpPlanningTime

                        'Determine the scheduled end time and realization of the surgery
                        .SchedEndTime = Days(DayNr).ORs(ORnr).ORschedule.PlannedEndTime + Round(Data.specialty(s).Weeks(w) _
                        .surg(p).Exp, 0)
                        .Realization = Data.specialty(s).Weeks(w).surg(p).Realization

                        'Add surgery to OR schedule including changeover time
                        Days(DayNr).ORs(ORnr).ORschedule.PlannedEndTime = .SchedStartTime + Round(Data.specialty(s).Weeks(w) _
                        .surg(p).Exp, 0) + ChangeOverTime
                    End With
                End If
            Next p
        Next s
    Next w

End Sub
```

```
Sub BubbleSortSurgeryListOnDuration(w As Long, s As Long, shortFirst As Boolean)
'Sorts patients based on expected surgery length in week w for specialty s

    Dim i As Long
    Dim j As Long
    Dim numPatients As Long
    Dim DummyL As Long
    Dim DummyD As Double

    numPatients = Data.specialty(s).Weeks(w).NumSurg

    'Loop through all patients
    For i = 1 To numPatients
        'Loop through all patients we are going to compare
        For j = i To numPatients
            'When sort method is shortest first
            If shortFirst = True Then
                'If patient i+1 has a shorter expected surgery duration then swap
                If Data.specialty(s).Weeks(w).surg(i).Exp > Data.specialty(s).Weeks(w).surg(j).Exp Then 'swap
                    Call SwapPatients(w, s, i, j)
                End If
            'When sort method is longest first
            ElseIf shortFirst = False Then
                'If patient i+1 has a shorter expected surgery duration then swap
                If Data.specialty(s).Weeks(w).surg(i).Exp < Data.specialty(s).Weeks(w).surg(j).Exp Then 'swap
                    Call SwapPatients(w, s, i, j)
                End If
            End If
        Next j
    Next i
End Sub
```

```vb
Sub RandomSortSurgeryList(w As Long, s As Long)
'Sorts patients per week in a random order in week w for specialty s

    Dim i As Long
    Dim j As Long
    Dim numPatients As Long

    'Get the number of patients
    numPatients = Data.specialty(s).Weeks(w).NumSurg

    'Check all the patients of the surgery list
    For i = 1 To numPatients

        'Select a random other patient of the surgery list
        j = CLng(((numPatients - i) * Rnd) + i)

        'Swap the patients if they are not the same
        If i <> j Then
            Call SwapPatients(w, s, i, j)
        End If
    Next i

End Sub
```

```vb
Sub SwapPatients(week As Long, specialty As Long, surgery1 As Long, surgery2 As Long)

    Dim DummyL As Long
    Dim DummyD As Double
    Dim DummyS As String

    'Swap admission type
    DummyS = Data.specialty(specialty).Weeks(week).surg(surgery1).AdmissionType
    Data.specialty(specialty).Weeks(week).surg(surgery1).AdmissionType = Data.specialty(specialty).Weeks(week). _
    surg(surgery2).AdmissionType
    Data.specialty(specialty).Weeks(week).surg(surgery2).AdmissionType = DummyS

    'Swap expected duration
    DummyD = Data.specialty(specialty).Weeks(week).surg(surgery1).Exp
    Data.specialty(specialty).Weeks(week).surg(surgery1).Exp = Data.specialty(specialty).Weeks(week).surg(surgery2).Exp
    Data.specialty(specialty).Weeks(week).surg(surgery2).Exp = DummyD

    'Swap realization
    DummyL = Data.specialty(specialty).Weeks(week).surg(surgery1).Realization
    Data.specialty(specialty).Weeks(week).surg(surgery1).Realization = Data.specialty(specialty).Weeks(week). _
    surg(surgery2).Realization
    Data.specialty(specialty).Weeks(week).surg(surgery2).Realization = DummyL

    'Swap specialty ID
    DummyL = Data.specialty(specialty).Weeks(week).surg(surgery1).SpecialtyID
    Data.specialty(specialty).Weeks(week).surg(surgery1).SpecialtyID = Data.specialty(specialty).Weeks(week). _
    surg(surgery2).SpecialtyID
    Data.specialty(specialty).Weeks(week).surg(surgery2).SpecialtyID = DummyL

    'Swap specialty name
    DummyS = Data.specialty(specialty).Weeks(week).surg(surgery1).SpecialtyName
    Data.specialty(specialty).Weeks(week).surg(surgery1).SpecialtyName = Data.specialty(specialty).Weeks(week). _
    surg(surgery2).SpecialtyName
    Data.specialty(specialty).Weeks(week).surg(surgery2).SpecialtyName = DummyS

    'Swap standard deviation
    DummyD = Data.specialty(specialty).Weeks(week).surg(surgery1).stdDev
    Data.specialty(specialty).Weeks(week).surg(surgery1).stdDev = Data.specialty(specialty).Weeks(week).surg(surgery2).stdDev
    Data.specialty(specialty).Weeks(week).surg(surgery2).stdDev = DummyD

    'Swap surgery type ID
    DummyL = Data.specialty(specialty).Weeks(week).surg(surgery1).SurgTypeID
    Data.specialty(specialty).Weeks(week).surg(surgery1).SurgTypeID = Data.specialty(specialty).Weeks(week). _
    surg(surgery2).SurgTypeID
    Data.specialty(specialty).Weeks(week).surg(surgery2).SurgTypeID = DummyL

End Sub
```

```
Function FindORBestFit(w As Long, s As Long, p As Long) As Collection
'Returns the OR that has the smallest free partition available that meets the requirements

    Dim d As Long
    Dim o As Long
    Dim bestFitDay As Long
    Dim bestFitOR As Long
    Dim bestFitTime As Double
    Dim var As Collection
    Set var = New Collection

    'We initialize the best fit time with a large number such that a first available OR is selected anyhow
    bestFitTime = 10000
    'Initialize counters
    bestFitDay = 0
    bestFitOR = 0

    'Check every OR on every day of the week
    For d = w * 5 - 4 To w * 5
        For o = 1 To NumORs
            'Check if the OR has the required specialty planned on this day
            If Days(d).ORs(o).SpecialtyID = s Then

                'Check if the expected time of the surgery fits in the time left for the session of the OR _
                since we assume we won't plan the surgery's end time after the session end time
                If (Days(d).ORs(o).ORschedule.SessionEnd - Days(d).ORs(o).ORschedule.PlannedEndTime) >= Data. _
                specialty(s).Weeks(w).surg(p).Exp Then

                    'Check if the time left for the session is the smallest until now since we want to chose the OR that _
                    has the best time left to fit this surgery
                    If (Days(d).ORs(o).ORschedule.SessionEnd - Days(d).ORs(o).ORschedule.PlannedEndTime) < bestFitTime Then
                        'If this is the case, set this OR as best fit
                        bestFitDay = d
                        bestFitOR = o
                        bestFitTime = Days(d).ORs(o).ORschedule.SessionEnd - Days(d).ORs(o).ORschedule.PlannedEndTime
                    End If
                End If
            End If
        Next o
    Next d

    'Add the earliest day and OR to a collection and let the function return the collection
    var.Add bestFitDay
    var.Add bestFitOR
    Set FindORBestFit = var

End Function
```

```
Function FindORFirstFit(w As Long, s As Long, p As Long) As Collection
'This function returns the earliest available OR on day d of week w for patient p with specialization s

    Dim d As Long
    Dim o As Long
    Dim firstFitDay As Long
    Dim firstFitOR As Long
    Dim leastSurgeries As Long
    Dim firstFitTime As Double
    Dim var As Collection
    Set var = New Collection

    'We initialize the least number of surgeries and earliest time with a large number such that a first available OR is _
    selected anyhow
    leastSurgeries = 10000
    firstFitTime = 10000

    'Initialize counters
    firstFitDay = 0
    firstFitOR = 0

    'Check every OR on every day of the week
    For d = w * 5 - 4 To w * 5
        For o = 1 To NumORs
            'Check if the OR has the required specialty planned on this day
            If Days(d).ORs(o).SpecialtyID = s Then
                'Check if the expected time of the surgery fits in the time left for the session of the OR _
                since we assume we won't plan the surgery's end time after the session end time
                If (Days(d).ORs(o).ORschedule.SessionEnd - Days(d).ORs(o).ORschedule.PlannedEndTime) >= Data.specialty(s) _
                .Weeks(w).surg(p).Exp Then
                    'We want to find the OR with the least number of surgeries planned and the earliest starting time _
                    so we check if this OR has an equal or less amount of surgeries planned than the current earliest found OR
                    If Days(d).ORs(o).ORschedule.NumSurg <= leastSurgeries Then
                        'If this OR has an earlier starting time than the current earliest found OR, indicate that this OR is _
                        the current earliest found OR
                        If Days(d).ORs(o).ORschedule.PlannedEndTime < firstFitTime Then
                            firstFitDay = d
                            firstFitOR = o
                            firstFitTime = Days(d).ORs(o).ORschedule.PlannedEndTime
                        End If
                    End If
                End If
            End If
        Next o
    Next d

    'Add the earliest day and OR to a collection and let the function return the collection
    var.Add firstFitDay
    var.Add firstFitOR
    Set FindORFirstFit = var

End Function
```

```
Sub RealizeSchedule(cancelStartAfterSession As Boolean)

    Dim d As Long
    Dim o As Long
    Dim p As Long
    Dim bound As Long
    Dim PatientCancelled As Boolean

    'Loop through all days
    For d = 1 To NumDays
        'Loop through all ORs
        For o = 1 To NumORs
            'Loop through all patients
            For p = 1 To Days(d).ORs(o).ORschedule.NumSurg
                PatientCancelled = False
                'Check if surgeries starting after the session ends should be cancelled
                If cancelStartAfterSession = True Then
                    With Days(d).ORs(o).ORschedule
                    'If these surgeries have to be cancelled, give no realization, starting and end time to this surgery _
                    and indicate that the patient is cancelled
                        If .RealEndTime >= .SessionEnd Then
                            .Schedule(p).Realization = 0
                            .Schedule(p).RealStartTime = 0
                            .Schedule(p).RealEndTime = 0
                            .Schedule(p).SurgeryPerformed = False
                            PatientCancelled = True
                        End If
                    End With
                End If

                'If the patient is not cancelled place the right realization info in the OR-schedule
                If Not PatientCancelled Then
                    'The first surgery is always on time
                    If p = 1 Then
                        Days(d).ORs(o).ORschedule.Schedule(p).RealStartTime = Days(d).ORs(o).ORschedule.Schedule(p) _
                        .SchedStartTime
                    Else
                        'The surgery starts at the end time of the previous surgery unless the scheduled starting time is _
                        later, since we assume a surgery cannot take place earlier than its appointment time
                        If Days(d).ORs(o).ORschedule.Schedule(p - 1).RealEndTime < Days(d).ORs(o).ORschedule.Schedule(p) _
                        .SchedStartTime Then
                            Days(d).ORs(o).ORschedule.Schedule(p).RealStartTime = Days(d).ORs(o).ORschedule.Schedule(p) _
                            .SchedStartTime
                        Else
                            Days(d).ORs(o).ORschedule.Schedule(p).RealStartTime = Days(d).ORs(o).ORschedule.Schedule(p - 1) _
                            .RealEndTime
                        End If
                    End If

                    'Realize the end times of the surgery
                    If p = Days(d).ORs(o).ORschedule.NumSurg Then
                        'The last surgery has no changeover time
                        Days(d).ORs(o).ORschedule.Schedule(p).RealEndTime = Days(d).ORs(o).ORschedule.Schedule(p) _
                        .RealStartTime + _
                            Round(Days(d).ORs(o).ORschedule.Schedule(p).Realization, 0)
                        'Update the end time of the session
                        Days(d).ORs(o).ORschedule.RealEndTime = Days(d).ORs(o).ORschedule.Schedule(p).RealEndTime
                    Else
                        'The end time of the other surgeries is the start time plus the realised time plus the changeover time
                        Days(d).ORs(o).ORschedule.Schedule(p).RealEndTime = Days(d).ORs(o).ORschedule.Schedule(p) _
                        .RealStartTime + _
                            Round(Days(d).ORs(o).ORschedule.Schedule(p).Realization, 0) + ChangeOverTime
                        'Update the end time of the session
                        Days(d).ORs(o).ORschedule.RealEndTime = Days(d).ORs(o).ORschedule.Schedule(p).RealEndTime
                    End If

                    'Indicate that the surgery has been performed
                    Days(d).ORs(o).ORschedule.Schedule(p).SurgeryPerformed = True
                End If
PatientCancelled:
            Next p
        Next o
    Next d

End Sub
```

```
Sub WriteScheduledPatients()

    Dim d As Long
    Dim o As Long
    Dim p As Long
    Dim r As Long
    Dim lastRow As Long
    Dim SpecialtyName As String
    Dim shScheduledPatients As Worksheet

    Set shScheduledPatients = ThisWorkbook.Sheets("Scheduled_Patients")

    'Make sheet empty to write new patient data
    With shScheduledPatients
       lastRow = .Cells(.Rows.Count, "R").End(xlUp).Row
        If lastRow > 1 Then
            .Rows(2 & ":" & lastRow).Delete
        End If
    End With

    'Check if patient data has to written out
    If PatientInfo Then
        r = 2
        'For all days, ORs on these days and number of patients in these ORs write data in scheduledpatients sheet
        For d = 1 To NumDays
            For o = 1 To NumORs
                For p = 1 To Days(d).ORs(o).ORschedule.NumSurg

                    shScheduledPatients.Cells(r, 1).Value = 1 + ((d - 1) \ 5) 'Week number
                    shScheduledPatients.Cells(r, 2).Value = Days(d).DayName
                    shScheduledPatients.Cells(r, 3).Value = o
                    shScheduledPatients.Cells(r, 4).Value = Days(d).ORs(o).ORschedule.sessionID
                    shScheduledPatients.Cells(r, 5).Value = Days(d).ORs(o).ORschedule.SessionStart
                    shScheduledPatients.Cells(r, 6).Value = Days(d).ORs(o).ORschedule.SessionEnd
                    shScheduledPatients.Cells(r, 8).Value = Days(d).ORs(o).ORschedule.Schedule(p).SurgTypeID
                    shScheduledPatients.Cells(r, 10).Value = Days(d).ORs(o).ORschedule.Schedule(p).Exp
                    shScheduledPatients.Cells(r, 11).Value = Days(d).ORs(o).ORschedule.Schedule(p).stdDev
                    shScheduledPatients.Cells(r, 12).Value = 10 * (d & o & p) 'Surgery ID
                    shScheduledPatients.Cells(r, 13).Value = Days(d).ORs(o).ORschedule.Schedule(p).SchedStartTime
                    shScheduledPatients.Cells(r, 14).Value = Days(d).ORs(o).ORschedule.Schedule(p).SchedEndTime
                    shScheduledPatients.Cells(r, 15) = Days(d).ORs(o).ORschedule.Schedule(p).RealStartTime
                    shScheduledPatients.Cells(r, 16) = Days(d).ORs(o).ORschedule.Schedule(p).RealEndTime
                    shScheduledPatients.Cells(r, 9).Value = Days(d).ORs(o).ORschedule.Schedule(p).AdmissionType
                    shScheduledPatients.Cells(r, 7).Value = Days(d).ORs(o).ORschedule.Schedule(p).SpecialtyName
                    shScheduledPatients.Cells(r, 17) = Days(d).ORs(o).ORschedule.Schedule(p).Realization
                    r = r + 1
                Next p
            Next o
        Next d
    End If

End Sub
```

```
Sub CalculateKPI(timeSlot As Long, KPIDay As String, MaxPatients As Long)
'This sub writes the percentage of surgeries that start in a specific time slot after a run

    Dim d As Long
    Dim o As Long
    Dim p As Long
    Dim nrSurgeries As Long
    Dim totalSurgeriesDay As Long
    Dim weekNr As Long
    Dim lastRow As Long
    Dim lowerBound As Long
    Dim upperBound As Long
    Dim KPIDayNr As Long
    Dim requirementCheck As Boolean
    Dim shOutput As Worksheet

    'Put worksheets in variables
    Set shOutput = ThisWorkbook.Sheets("Percentage_Surgeries_Timeslot")
    Set shSurgeriesStartingPerTimeslot = ThisWorkbook.Sheets("Surgeries_Starting_per_Timeslot")

    'Make the KPI output sheet empty to write new patient data
    With shOutput
        lastRow = .Cells(.Rows.Count, "A").End(xlUp).Row
        If lastRow > 1 Then
            .Range("A2:D" & lastRow).ClearContents
            .Range("F2:I" & lastRow).ClearContents
        End If
    End With

    'Initialize the counters for surgeries on this day and surgeries in specified timeslot
    totalSurgeriesDay = 0
    nrSurgeries = 0

    'Give every day a day number
    If KPIDay = "Monday" Then
        KPIDayNr = 1
    ElseIf KPIDay = "Tuesday" Then
        KPIDayNr = 2
    ElseIf KPIDay = "Wednesday" Then
        KPIDayNr = 3
    ElseIf KPIDay = "Thursday" Then
        KPIDayNr = 4
    ElseIf KPIDay = "Friday" Then
        KPIDayNr = 5
    End If

    'Set the upper and lower bound of the time slot and initialize the week number with 1
    lowerBound = shSurgeriesStartingPerTimeslot.Cells(1 + timeSlot, 3).Value * 1440
    upperBound = shSurgeriesStartingPerTimeslot.Cells(1 + timeSlot, 4).Value * 1440
    weekNr = 1
```

```
    'Check all days
    For d = 1 To NumDays
        'Check if this is the right day to measure the KPI, making use of the floor function
        If (d - (((d - 1) \ NumDaysPerWeek) * NumDaysPerWeek)) = KPIDayNr Then
            'Check all patients of all ORs on this day
            For o = 1 To NumORs
                For p = 1 To Days(d).ORs(o).ORschedule.NumSurg

                    'Initialize the requirement check to false
                    requirementCheck = False

                    'Check if we need to show the KPI for a specific admission type or for all the surgeries
                    If KPIAdmissionType = "All" Then
                        requirementCheck = True
                    'If the surgeries need to be checked on admission type, make sure the selected surgery has the required _
                    surgery type
                    ElseIf KPIAdmissionType = "Clinical" And Days(d).ORs(o).ORschedule.Schedule(p).AdmissionType = "K" Then
                        requirementCheck = True
                    ElseIf KPIAdmissionType = "Day Care" And Days(d).ORs(o).ORschedule.Schedule(p).AdmissionType = "D" Then
                        requirementCheck = True
                    ElseIf KPIAdmissionType = "Other" And Days(d).ORs(o).ORschedule.Schedule(p).AdmissionType = "OTHER" Then
                        requirementCheck = True
                    End If

                    'Check if this patient has the right admission type and if the surgery takes place
                    If requirementCheck And Days(d).ORs(o).ORschedule.Schedule(p).Realization <> 0 Then
                        'Increase the total number of surgeries of this admission type on this day
                        totalSurgeriesDay = totalSurgeriesDay + 1
                        'Check if the starting time of the surgery falls in the required timeslot _
                        'and if the surgery does actually take place
                        If Days(d).ORs(o).ORschedule.Schedule(p).RealStartTime >= lowerBound And _
                            Days(d).ORs(o).ORschedule.Schedule(p).RealStartTime <= upperBound Then
                            'Increase the counter for the number of surgeries
                            nrSurgeries = nrSurgeries + 1
                        End If
                    End If
                Next p
            Next o
        End If

        'If this is the last day of the week write the week number and the counters _
        to the output sheet and set the counters to zero
        If (1 + ((d + 1 - 1) \ NumDaysPerWeek)) <> weekNr Then 'The week number of the next day _
                                                 is not equal to the week number of this day
            'Only count the number of surgeries that exceed the maximum number _
            'of surgeries starting in a timeslot
            nrSurgeries = nrSurgeries - MaxPatients

            'Let number of surgeries be nonnegative
            If nrSurgeries < 0 Then
                nrSurgeries = 0
            End If

            'Write data of odd weeks in the left column
            If Not (1 + ((d - 1) \ NumDaysPerWeek)) Mod 2 = 0 Then
                With shOutput
                    lastRow = .Cells(.Rows.Count, "A").End(xlUp).Row
                    .Cells(lastRow + 1, 1) = weekNr
                    .Cells(lastRow + 1, 2) = nrSurgeries
                    .Cells(lastRow + 1, 3) = totalSurgeriesDay

                    'Make sure we do not divide by zero
                    If totalSurgeriesDay = 0 Then
                        .Cells(lastRow + 1, 4) = 0
                    Else
                        .Cells(lastRow + 1, 4) = nrSurgeries / totalSurgeriesDay
                    End If
                End With
            'Write data of even weeks in the right column
            ElseIf (1 + ((d - 1) \ NumDaysPerWeek)) Mod 2 = 0 Then
                With shOutput
                    lastRow = .Cells(.Rows.Count, "F").End(xlUp).Row
                    .Cells(lastRow + 1, 6).Value = weekNr
                    .Cells(lastRow + 1, 7) = nrSurgeries
                    .Cells(lastRow + 1, 8) = totalSurgeriesDay

                    'Make sure we do not divide by zero
                    If totalSurgeriesDay = 0 Then
                        .Cells(lastRow + 1, 9).Value = 0
                    Else
                        .Cells(lastRow + 1, 9).Value = nrSurgeries / totalSurgeriesDay
                    End If
                End With
            End If

            totalSurgeriesDay = 0
            nrSurgeries = 0
            weekNr = weekNr + 1
        End If
    Next d
End Sub
```

```
Sub RecalculateKPI()
'This sub opens a userform to recalculate the percentage surgeries starting per timeslot

    Dim i As Long
    Dim lastRow As Long

    'Show the recalculate KPI form
    ufRecalculateKPI.Show

    'Check all cells with control names to store the right values in the right variables
    With ThisWorkbook.Sheets("Controls")
        For i = 1 To 30
            If .Cells(i, 1).Value = "coboTimeSlotNr" Then
                TimeSlotNr = .Cells(i, 3).Value
            ElseIf .Cells(i, 1).Value = "coboWeekDay" Then
                WeekDay = .Cells(i, 2).Value
            ElseIf .Cells(i, 1).Value = "tbPatientsPerTimeslotAllowed" Then
                PatientsPerTimeslotPreferred = .Cells(i, 2).Value
            ElseIf .Cells(i, 1).Value = "coboKPIAdmissionType" Then
                KPIAdmissionType = .Cells(i, 2).Value
            End If
        Next i
    End With

    'Calculate the percentage of patients in a specified time slot in a week
    Call CalculateKPI(TimeSlotNr, WeekDay, PatientsPerTimeslotPreferred)

End Sub
```

```
Sub CreateSchedulingExample()

    Application.ScreenUpdating = False
    Application.EnableEvents = False
    Application.Calculation = xlCalculationManual

    Dim i As Long
    Dim j As Long
    Dim lastRow As Long
    Dim operatingRoom As Long
    Dim operatingRoomName As String
    Dim rg As Range
    Dim rg1 As Range
    Dim tbl As ListObject
    Dim shCreateSchedulingExample As Worksheet
    Dim shScheduleVisualization As Worksheet
    Dim shControls As Worksheet

    Set shCreateSchedulingExample = ThisWorkbook.Sheets("Create_Scheduling_Example")
    Set shScheduleVisualization = ThisWorkbook.Sheets("Schedule_Visualization")
    Set shControls = ThisWorkbook.Sheets("Controls")

    'Empty all the tables for the schedules
    For Each tbl In shCreateSchedulingExample.ListObjects
        If Not tbl.DataBodyRange Is Nothing Then
            tbl.DataBodyRange.Delete
        End If
    Next tbl

    'Clear all specialty data from the visualization sheet for the odd week
    Set rg = shScheduleVisualization.Range("D23:AG23")
    For Each rg1 In rg
        If rg1.MergeCells Then rg1.MergeArea.ClearContents
    Next

    'Clear all specialty data from the visualization sheet for the even week
    Set rg = shScheduleVisualization.Range("D43:AG43")
    For Each rg1 In rg
        If rg1.MergeCells Then rg1.MergeArea.ClearContents
    Next

    'Clear all time data from the visualization sheet for both weeks
    shScheduleVisualization.Range("D27:AG27").ClearContents
    shScheduleVisualization.Range("D27:AG27").ClearContents

    'Get the name of the OR we have to show
    operatingRoomName = shScheduleVisualization.Cells(4, 3).Value

    'Get the corresponding number of the OR
    For i = 1 To 24
        If shControls.Cells(i, 7).Value = operatingRoomName Then
            operatingRoom = shControls.Cells(i, 6).Value
        End If
    Next i

    'Initialize the days
    j = 1

    'Fill all the tables for the schedules
    Do Until j = 11
        With Days(j).ORs(operatingRoom).ORschedule
            'If there is a specialty planned write values
            If Not Days(j).ORs(operatingRoom).SpecialtyID = 0 Then
                For i = 1 To Days(j).ORs(operatingRoom).ORschedule.NumSurg
                    shCreateSchedulingExample.Cells(1 + i, 1 + 6 * (j - 1)).Value = .Schedule(i).SchedStartTime / (24 * 60)
                    shCreateSchedulingExample.Cells(1 + i, 2 + 6 * (j - 1)).Value = .Schedule(i).Exp / (24 * 60)
                    shCreateSchedulingExample.Cells(1 + i, 4 + 6 * (j - 1)).Value = .Schedule(i).RealStartTime / (24 * 60)
                    shCreateSchedulingExample.Cells(1 + i, 5 + 6 * (j - 1)).Value = .Schedule(i).Realization / (24 * 60)
                Next i
            End If
```

```
            'Write the specialty names below the schedules
            If j < 6 Then
                'If there is a specialty planned write values
                If Not Days(j).ORs(operatingRoom).SpecialtyID = 0 Then
                    'Odd week
                    shScheduleVisualization.Cells(23, -3 + 7 * j).Value = .Schedule(1).SpecialtyName
                    shScheduleVisualization.Cells(25, -3 + 7 * j).Value = .SessionStart / (24 * 60)
                    shScheduleVisualization.Cells(25, -2 + 7 * j).Value = .SessionEnd / (24 * 60)
                End If
            Else
                'If there is a specialty planned write values
                If Not Days(j).ORs(operatingRoom).SpecialtyID = 0 Then
                    'Even week
                    shScheduleVisualization.Cells(43, -3 + 7 * (j - 5)).Value = .Schedule(1).SpecialtyName
                    shScheduleVisualization.Cells(45, -3 + 7 * (j - 5)).Value = .SessionStart / (24 * 60)
                    shScheduleVisualization.Cells(45, -2 + 7 * (j - 5)).Value = .SessionEnd / (24 * 60)
                End If
            End If
        End With
        j = j + 1
    Loop

    Application.ScreenUpdating = True
    Application.EnableEvents = True
    Application.Calculation = xlCalculationAutomatic

End Sub
```

```
Sub CalculateSurgeriesStartingPerTimeslot()
'This sub calculates the number of surgeries starting per timeslot without writing patient data

    Dim d As Long
    Dim o As Long
    Dim p As Long
    Dim i As Long
    Dim j As Long
    Dim nrSurgeriesPlanned As Long
    Dim nrSurgeriesTakePlace As Long
    Dim nrSurgeriesDayCare As Long
    Dim nrSurgeriesClinical As Long
    Dim nrSurgeriesOther As Long
    Dim plannedSlotFound As Boolean
    Dim realSlotFound As Boolean

    'Matrix to count starting times with time slot boundaries in the first two columns
    Dim surgeriesStartingPerTimeslot(1 To 53, 0 To 9) As Long
    Dim tblSurgeriesStartingPerTimeslot As ListObject

    'Assign sheet and table to variables
    Set shDashboardMeasurements = ThisWorkbook.Sheets("Dashboard_Measurements")
    Set shSurgeriesStartingPerTimeslot = ThisWorkbook.Sheets("Surgeries_Starting_per_Timeslot")
    Set tblSurgeriesStartingPerTimeslot = shSurgeriesStartingPerTimeslot.ListObjects("tblSurgeriesStartingPerTimeslot")

    'Place the time slot boundaries in the first two columns of the counting matrix
    For i = 1 To 53
        surgeriesStartingPerTimeslot(i, 0) = shSurgeriesStartingPerTimeslot.Cells(1 + i, 14).Value
        surgeriesStartingPerTimeslot(i, 1) = shSurgeriesStartingPerTimeslot.Cells(1 + i, 15).Value
    Next i

    'Clear the surgery start counters
    For i = 5 To 12
        tblSurgeriesStartingPerTimeslot.ListColumns(i).DataBodyRange.Clear
    Next i

    'Check all days, ORs on days, surgeries in ORs
    For d = 1 To NumDays
        For o = 1 To NumORs
            For p = 1 To Days(d).ORs(o).ORschedule.NumSurg

                'Initialize the timeslot and counter variables for every patient
                plannedSlotFound = False
                realSlotFound = False

                With Days(d).ORs(o).ORschedule.Schedule(p)
                    'Find the timeslot of the planned and real starting time of this surgery
                    For i = 1 To 53
                        'Check if the surgery is scheduled, the scheduled starting time takes later place than the lower bound _
                        of the time slot and takes earlier place than the upper bound of the time slot
                        If .SchedStartTime <> 0 And .SchedStartTime >= surgeriesStartingPerTimeslot(i, 0) And _
                        .SchedStartTime <= surgeriesStartingPerTimeslot(i, 1) Then

                            'Increase the total number of planned surgeries starting in this time slot and indicate the timeslot
                            is found
                            surgeriesStartingPerTimeslot(i, 5) = surgeriesStartingPerTimeslot(i, 5) + 1
                            nrSurgeriesPlanned = nrSurgeriesPlanned + 1
                            plannedSlotFound = True

                            'Check the admission type and add the surgery to the counter of the corresponding admission type
                            If .AdmissionType = "K" Then
                                'If this admission type is clinical increase the counter
                                surgeriesStartingPerTimeslot(i, 2) = surgeriesStartingPerTimeslot(i, 2) + 1
                            ElseIf .AdmissionType = "D" Then
                                'If this admission type is day care increase the counter
                                surgeriesStartingPerTimeslot(i, 3) = surgeriesStartingPerTimeslot(i, 3) + 1
                            ElseIf .AdmissionType = "OTHER" Then
                                'If this admission type is other increase the counter
                                surgeriesStartingPerTimeslot(i, 4) = surgeriesStartingPerTimeslot(i, 4) + 1
                            End If
                        End If

                        'Check if the surgery takes place, the real starting time takes later place than the lower bound _
                        of the time slot and takes earlier place than the upper bound of the time slot
                        If .RealStartTime <> 0 And .RealStartTime >= surgeriesStartingPerTimeslot(i, 0) And _
                        .RealStartTime <= surgeriesStartingPerTimeslot(i, 1) Then
```

```vba
                        'Increase the total number of real surgeries starting in this time slot and indicate the timeslot _
                        is found
                        surgeriesStartingPerTimeslot(i, 9) = surgeriesStartingPerTimeslot(i, 9) + 1
                        nrSurgeriesTakePlace = nrSurgeriesTakePlace + 1
                        realSlotFound = True

                        'Check the admission type and add the surgery to the counter of the corresponding admission type
                        If .AdmissionType = "K" Then
                            'If this admission type is clinical increase the counter
                            surgeriesStartingPerTimeslot(i, 6) = surgeriesStartingPerTimeslot(i, 6) + 1
                            nrSurgeriesClinical = nrSurgeriesClinical + 1
                        ElseIf .AdmissionType = "D" Then
                            'If this admission type is day care increase the counter
                            surgeriesStartingPerTimeslot(i, 7) = surgeriesStartingPerTimeslot(i, 7) + 1
                            nrSurgeriesDayCare = nrSurgeriesDayCare + 1
                        ElseIf .AdmissionType = "OTHER" Then
                            'If this admission type is day care increase the counter
                            surgeriesStartingPerTimeslot(i, 8) = surgeriesStartingPerTimeslot(i, 8) + 1
                            nrSurgeriesOther = nrSurgeriesOther + 1
                        End If
                    End If

                    'If the the planned timeslot and real timeslot are found, do not look further otherwise go to the next _
                    timeslot
                    If plannedSlotFound And realSlotFound Then
                        Exit For
                    End If
                Next i

            End With
        Next p
    Next o
Next d

'Write matrix in table
For i = 1 To 53
    For j = 1 To 8
        tblSurgeriesStartingPerTimeslot.ListColumns(4 + j).DataBodyRange(i, 1).Value = surgeriesStartingPerTimeslot _
        (i, 1 + j) / NumDays
    Next j
Next i

'Write number of planned and cancelled surgeries in dashboard measurements sheet
shDashboardMeasurements.Cells(3, 2).Value = nrSurgeriesPlanned
shDashboardMeasurements.Cells(8, 2).Value = nrSurgeriesPlanned - nrSurgeriesTakePlace
shDashboardMeasurements.Cells(14, 2).Value = nrSurgeriesClinical
shDashboardMeasurements.Cells(15, 2).Value = nrSurgeriesDayCare
shDashboardMeasurements.Cells(16, 2).Value = nrSurgeriesOther
shDashboardMeasurements.Cells(17, 2).Value = nrSurgeriesTakePlace

End Sub
```

```vba
Sub CalculateSurgeriesOnTime()
'This sub calculates the number of surgeries per specialty that are on time by making use of a counter matrix

    Dim d As Long
    Dim o As Long
    Dim p As Long
    Dim i As Long
    Dim surgeryOnTimeMatrix(1 To NumSpecialties, 1 To 2) As Long
    Dim shSurgeriesOnTime As Worksheet
    Dim tblSurgeriesOnTime As ListObject

    'Assign sheet and table to variables
    Set shSurgeriesOnTime = ThisWorkbook.Sheets("Surgeries_On_Time")
    Set tblSurgeriesOnTime = shSurgeriesOnTime.ListObjects("tblSurgeriesOnTime")

    'Delete counters from table
    tblSurgeriesOnTime.ListColumns(2).DataBodyRange.ClearContents
    tblSurgeriesOnTime.ListColumns(3).DataBodyRange.ClearContents

    'Check all the surgeries per OR on every day
    For d = 1 To NumDays
        For o = 1 To NumORs
            For p = 1 To Days(d).ORs(o).ORschedule.NumSurg
                With Days(d).ORs(o).ORschedule.Schedule(p)
                    'If the surgery takes place and earlier than the planned starting time including the _
                    accepted delay  the surgery is on time
                    If .RealStartTime <> 0 And .RealStartTime <= .SchedStartTime + AcceptedDelay Then
                        'Increase on time and total value when surgery is on time
                        surgeryOnTimeMatrix(.SpecialtyID, 1) = surgeryOnTimeMatrix(.SpecialtyID, 1) + 1
                        surgeryOnTimeMatrix(.SpecialtyID, 2) = surgeryOnTimeMatrix(.SpecialtyID, 2) + 1
                    'In all the other cases where the surgery takes place, the surgery is too late
                    ElseIf .RealStartTime <> 0 Then
                        'Increase total value if surgery started too late
                        surgeryOnTimeMatrix(.SpecialtyID, 2) = surgeryOnTimeMatrix(.SpecialtyID, 2) + 1
                    End If
                End With
            Next p
        Next o
    Next d

    'Write matrix in table
    For i = 1 To NumSpecialties
        tblSurgeriesOnTime.ListColumns(2).DataBodyRange(i, 1).Value = surgeryOnTimeMatrix(i, 1)
        tblSurgeriesOnTime.ListColumns(3).DataBodyRange(i, 1).Value = surgeryOnTimeMatrix(i, 2)
    Next i

End Sub
```

```
Sub CalculateOccupancy()

    Dim d As Long
    Dim o As Long
    Dim p As Long
    Dim occupancy As Long
    Dim totalOccupancy As Long
    Dim totalSessionDuration As Long

    'Name sheets
    Set shDashboardMeasurements = ThisWorkbook.Sheets("Dashboard_Measurements")

    'Determine session information for all sessions on every day and in every OR
    For d = 1 To NumDays
        For o = 1 To NumORs
            'Initialize the occupancy of the OR-session
            occupancy = 0

            'Consider the OR-schedule for every OR on every day
            With Days(d).ORs(o).ORschedule
                'Write total OR time
                For p = 1 To .NumSurg
                    occupancy = occupancy + .Schedule(p).Realization
                Next p

                'Construct occupancy parameters
                .occupancy = occupancy
                totalOccupancy = totalOccupancy + occupancy
                totalSessionDuration = totalSessionDuration + .SessionEnd - .SessionStart
            End With
        Next o
    Next d

    'Write occupancy KPI in dashboard measurements sheet
    shDashboardMeasurements.Cells(7, 2).Value = totalOccupancy / totalSessionDuration

End Sub
```

```
Sub CalculateSurgeryDuration()
'This sub fills a table with surgery duration bins to create a historgram of surgery duration per specialty

    Dim d As Long
    Dim o As Long
    Dim p As Long
    Dim i As Long
    Dim j As Long
    Dim surgeryDurationMatrix(0 To NumSpecialties + 1, 0 To 31) As Long
    Dim nextPatient As Boolean
    Dim tblSurgeryDuration As ListObject
    Dim shSurgeryDuration As Worksheet

    Set shSurgeryDuration = ThisWorkbook.Sheets("Surgery_Duration")
    Set tblSurgeryDuration = shSurgeryDuration.ListObjects("tblSurgeryDuration")

    'Initialize next patient indicator
    nextPatient = False

    'Empty the table
    For i = 2 To 31
        tblSurgeryDuration.ListColumns(i).DataBodyRange.Clear
    Next i

    'Write the bin boundaries down in the matrix
    For i = 1 To 30
        surgeryDurationMatrix(0, i) = tblSurgeryDuration.HeaderRowRange(1, 1 + i).Value
    Next i

    'Consider all surgeries in every OR on every day
    For d = 1 To NumDays
        For o = 1 To NumORs
            For p = 1 To Days(d).ORs(o).ORschedule.NumSurg
                With Days(d).ORs(o).ORschedule.Schedule(p)
                    'Consider all bins
                    For j = 1 To 31
                        'Check the lower bound
                        If .Realization > surgeryDurationMatrix(0, j - 1) Then
                            'If the right bin is not found yet and this is the last bin, we don't need an upper bound
                            If j = 31 Then
                                surgeryDurationMatrix(.SpecialtyID, j) = 1 + surgeryDurationMatrix(.SpecialtyID, j)
                                surgeryDurationMatrix(21, j) = 1 + surgeryDurationMatrix(21, j)
                                nextPatient = True
                                Exit For
                            'Check the upper bound
                            ElseIf .Realization <= surgeryDurationMatrix(0, j) Then
                                surgeryDurationMatrix(.SpecialtyID, j) = 1 + surgeryDurationMatrix(.SpecialtyID, j)
                                surgeryDurationMatrix(21, j) = 1 + surgeryDurationMatrix(21, j)
                                nextPatient = True
                                Exit For
                            End If
                        End If
                    Next j
                    'Check if we have to consider the next specialty
                    If nextPatient Then
                        'Move on to the next patient
                        Exit For
                    End If
                End With
                nextPatient = False
            Next p
        Next o
    Next d

    'Write matrix in sheet
    For i = 1 To NumSpecialties + 1
        For j = 1 To 31
            shSurgeryDuration.Cells(2 + i, 2 + j).Value = surgeryDurationMatrix(i, j)
        Next j
    Next i

End Sub
```