Technische Universiteit
Eindhoven
University of Technology

/ Department of
Mechanical Engineering

Manufacturing Networks

# Efficiency of the Emergency Department of the Catharina Hospital Eindhoven

J.J.D. Timmermans

Where innovation starts

Efficiency of the Emergency Department of the Catharina Hospital Eindhoven

J.J.D. Timmermans (0615795)

MN-420704

Master Thesis

Supervisor TU/e: prof.dr.ir. I.J.B.F. Adan
Advisor TU/e: dr. ir. A.A.J. Lefeber
Advisors CZE: ir. M. Wolleswinkel-Schriek
drs. A. v.d. Waarsenburg

Eindhoven University of Technology
Department of Mechanical Engineering
Manufacturing Networks Group

Eindhoven, November 28, 2012

MASTER PROJECT (INDIVIDUAL)

EINDHOVEN UNIVERSITY OF TECHNOLOGY
Department of Mechanical Engineering
Manufacturing Networks Group

May 2012

| | |
|---|---|
| Student | J.J.D. Timmermans |
| Supervisor | Prof.dr.ir. I.J.B.F. Adan |
| Advisors | Dr.ir. A.A.J. Lefeber<br>ir. M. Wolleswinkel-Schriek (CZE)<br>A. van de Waarsenburg (CZE) |
| Start | Januari 2012 |
| Finish | October 2012 |
| Title | Efficiency of the Emergency Department of the Catharina Hospital in Eindhoven |

Subject

The Catharina Hospital is the largest hospital in the area of Eindhoven and has a capacity of almost 700 beds. The Emergency Department (ED) is renovated in Januari 2010 and it became part of the Emergency Post. An Emergency Post (EP) has been a collaboration between the General Practitioners (GP), the pharmacy and the ED of a hospital. Patients can visit the EP for emergency care and medication. This collaboration provides the proper care from the right health care provider on one location. Health care costs are increasing and thus also the need increases to use the resources more efficiently. Clearly, a decrease in waiting time will also improve patients experience.

Assignment

The first part of the assignment is to build a simulation model of the ED. This model will be used to investigate the capacity level needed to deliver the health care within a standard time. The capacity consists of (ED-)physicians, (ED-)medical interns, (ED-)nurses and treatment rooms. Using this model, it should be possible to address questions such as:

- What capacity is minimal needed on a Monday? And how does this differ in comparison to a Sunday?
- How much does the waiting time decrease if the availability of a physicians increases?

Next, an aggregate model will be build to gain more insight in the efficiency of the ED at a single glance. This information can be used to compare different days or to compare the efficiency of this ED with an ED of another hospital.

Prof.dr.ir. I.J.B.F. Adan Dr.ir. A.A.J. Lefeber ir. M. Wolleswinkel-Schriek A. van de Waarsenburg

Manufacturing Networks Department of Mechanical Engineering

# Preface

With this thesis, I conclude my Master in Mechanical Engineering at Eindhoven University of Technology. As part of my Master, I have been to Auckland, New Zealand in the autumn of 2011 for an internship on the subject of operation research in health care systems. The trip itself was a great experience, but I also enjoyed working on the project. Therefore, it is not a coincidence that the subject of this thesis is within the same research area.

First, I would like to thank professor Adan and Erjen Lefeber for their support during my project. Their comments and remarks during our weekly meetings helped me to structure my thoughts and to improve my work. I would also like to thank professor Rooda and Albert Hofkamp for their support and prompt assistance on Chi 3.

Next, I would like to thank my coaches from the Catharina Hospital in Eindhoven, Annet van de Waarsenburg and Mireille van Wolleswinkel. My thanks also goes to Maaike Maas, Etienne Houben, Frans Francken and Tim Brouwer. They have guided me to make me understand the hospital environment, taught me the ins and outs of the Emergency Department and provided me with the necessary information.

Moreover, I would like to thank my fellow students in the System Engineering Lab for their input on my project and for the necessary relaxation during breaks. Also, thanks to my friends for their laughs and support during evenings and weekends.

A special thanks goes out to my parents, grandparents and sisters. Their enthusiasm and support have inspired and motivated me during my entire study career to achieve my goals. As well I would like to thank my girlfriend Melanie for her great support!

Jordi Timmermans
Eindhoven, November 28, 2012

# Summary

Due to rising health care costs, the health care sector is required to work more efficient and to better utilise their resources. Therefore, the LEAN principles are introduced in health care. This also holds for Emergency Department (ED) of the Catharina Hospital in Eindhoven (CZE).

This research is conduced to provide more insight into the current efficiency of the ED, to answer capacity questions and to find and evaluate options to reduce waiting time. Waiting time reduction can increase patient experience and can improve the quality of the health care.

On average, the patient waiting time does not exceed the target maximal waiting time. However, this time is exceeded for a large group of patients due to the variability at the ED. The variability is caused by variation in patient arrivals, by the different types of patients and by the type of treatment.

Capacity issues questions that arise are: 'What capacity is at least required on a typical Monday such that the number of patients that exceed the target time is relatively small?' and 'How does the waiting time change if patients arriving by ambulance are treated with the same priority as other patients?'

To answer these question, a software package has been developed. This package includes a simulation model of the ED, a tool to generate simulation input files and tools to analyse historical data and simulation output.

The simulation model includes the arrival, triage, waiting room and the treatment process. The waiting room process also includes the patient dispatching policy to determine which patient is treated first. A data analyses has been conducted to find the important elements of the ED that determine the processes and that case the variabilities. Next, the model has been verified and validated, it represents the reality relatively close.

Together with the tools for analysis, the simulation model can be used to search for performance improvements. Several performance indicators can be used such as waiting time per patient or utilisation of resources. In this report, improvement opportunities are tested for waiting time reduction on Mondays, one of the most crowded days of the week. The results show that a reduction of the treatment time has a large positive influence on the waiting time. Especially patients with normal urgency benefit from this reduction.

Next, several scenarios can be investigated using the simulation model. Amongst others, the consequence of an increased number of patient arrivals is examined in this report. This increase can be caused by e.g. a (temporary) closure of a neighboring ED. Especially for patient with normal urgency, the waiting time will increase much.

The software package proved to be very useful in providing information on the efficiency of the ED. The simulation model can be used to simulate and evaluate scenarios before trailing them in the real world.

# Samenvatting

De kosten in de zorgsector stijgen waardoor de vraag naar efficiëntere zorg toeneemt. Het LEAN werken is daarin een veel gehoorde term. Door verspilling en overschot te beperken is het mogelijk om met dezelfde middelen meer bereiken. Dit speelt ook bij de Spoedeisende Hulp (SEH) van het Catharina Ziekenhuis in Eindhoven (CZE).

Dit onderzoek is opgezet om inzicht te geven in de huidige efficiëntie van de SEH en om capaciteitsvraagstukken te beantwoorden. Het verkorten van de wachttijden voor patiënten is ook een doel. Door kortere wachttijden is het mogelijk om de kwaliteit van de zorg te verbeteren. Ook zal de patiëntbeleving verbeteren.

Bij de SEH zijn verschillende richtwachttijden vastgesteld. Voor de gemiddelde patiënt worden deze richtlijnen gehaald maar door de vele variabiliteit zijn er veel relatief veel patiënten die langer moeten wachten dan de richttijd. Bij variabiliteit kan o.a. gedacht worden aan verschillen in aankomstintensiteit, soort patiënten en soort behandeling.

Bij capaciteitsvraagstukken kan gedacht worden aan vragen als: 'Welke bezetting is op een gemiddelde maandag nodig zodat maar een relatief klein percentage van de patiënten de maximaal vastgestelde wachttijd overschrijdt?' en 'Wat is de invloed op de wachttijd als patiënten aangekomen met de ambulance met gelijke prioriteit worden behandeld als de overige patiënten?'.

Om antwoorden te geven op deze vraagstukken is een softwarepakket ontwikkeld. Tot dit pakket behoren het simulatiemodel van de SEH en hulpprogramma's om invoerbestanden voor het model te genereren en om de historische data en de simulatie resultaten te analyseren.

Factoren als het triage-, behandel- en wachtkamerproces zijn meegenomen in het detail simulatiemodel. In het wachtkamerproces is ook de rol van de oudste van dienst meegenomen. Deze hoofdverpleegkundige bepaalt met de artsen in welke volgorde de patiënten behandeld worden. Het model is gevalideerd en geverifieerd, de simulatieresultaten geven een relatief goede benadering van de werkelijke situatie.

Het model is samen met de hulpprogrammas gebruikt voor analyse van de huidige situatie. In dit report is uitgewerkt hoe een analyse gedaan kan worden voor de verkorting van de wachttijden op de maandag, een van de drukste dagen van de week. Daaruit is gebleken dat vooral het verkorten van de behandeltijden een grote invloed heeft op de wachttijd. Deze invloed is vooral te zien bij patiënten met klachten van gemiddelde urgentie.

Daarnaast is het mogelijk verschillende scenarios te simuleren. Er wordt in dit report o.a. gekeken naar het effect van een toename van patiënten door bijvoorbeeld een sluiting van een naburige SEH. Een toename van het aantal patiënten zorgt hierbij ook vooral voor een toename van de wachttijden voor patiënten met gemiddelde urgentie.

Het softwarepakket met simulatiemodel is dus erg bruikbaar om inzicht te geven in de efficiëntie van de afdeling. Ook is het mogelijk verbetervoorstellen te simuleren en evalueren voordat deze in werkelijkheid worden toegepast.

# Contents

# Chapter 1

# Introduction

This project is a collaboration between Eindhoven University of Technology and the Catharina Hospital in Eindhoven (CZE). This chapter starts with a brief introduction of the Catharina Hospital in Eindhoven. The second section gives an introduction of the Emergency Department. Next, the motivation and objective of this research are given, followed by a review of related work. This chapter ends with an outline of the following chapters of this report.


Figure 1.1: Integrated Emergency Post (CZE).

## 1.1 Catharina Hospital Eindhoven

The Catharina Hospital dates from the first half of the 19th century and was initiated by the nuns of the St. Catherine parish of Eindhoven. Over the next 130 years, the hospital used the name ‘Rooms Katholieke Binnenziekenhuis’. In the early sixties of the last century, the hospital became too large to fit in their accommodation located in the city center of Eindhoven. Therefore, the construction of a new hospital in Woensel, one of the suburbs of Eindhoven, started in 1968. The new Catharina Hospital opened in September 1973. Nowadays, the hospital has almost 700 beds, over 3400 employees and annually almost a half million patient visits [5]. This study focusses on the efficiency of the Emergency Department, one of the 39 (sub-)specialities in the hospital.

## 1.2 Emergency Department

The Emergency Department (ED) of the Catharina Hospital, Figure 1.1, is a specialistic and multidisciplinary department that operates 24/7. The goal of the ED is to deliver high quality critical, emergent and urgent care in a professional and safe environment. The ED is officially second care, which means that patients should arrive by referral. But in practice, still many patients are self-referrals. To cope with this problem, an Integrated Emergency Post (IEP) is set-up to function as a gatekeeper to redirect the patients, see Figure 1.2.

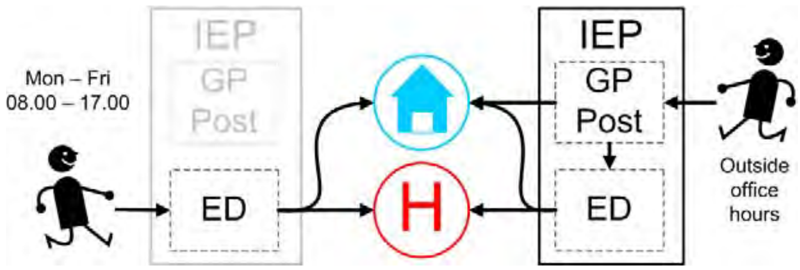


Figure 1.2: Catharina Hospital Eindhoven - Integrated Emergency Post.

The IEP operates outside regular office hours and is a partnership between the ED, the General Practitioners Post (GPP) and the pharmacy. These three groups are all located in the hospital and together they can provide the proper care from the right health care provider in one location. Amongst others, also the St. Anna Hospital in Geldrop and the Elkerliek hospital in Helmond have IEPs. Outside office hours, the GP at the GPP will decide whether or not to refer the patient to the ED.

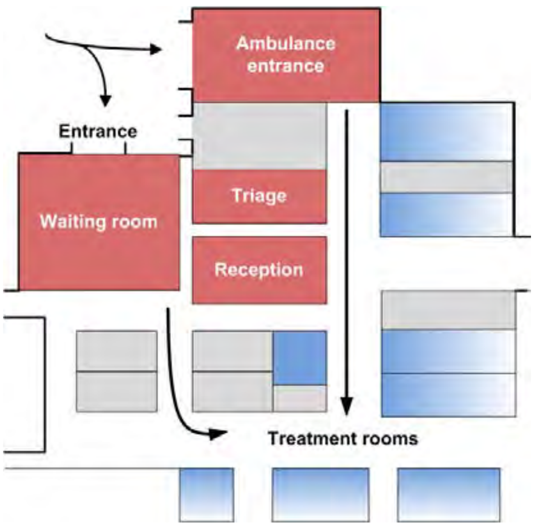


Figure 1.3: Catharina Hospital Eindhoven - Partial map ED.

After discussing the IEP, now we focus on the ED itself. In 2011, over 34.000 patients have visited the ED of the CZE for emergency care and medication. Arriving patients have to register at the front desk and can take place in the waiting room, see Figure 1.3. The triage system is used as a waiting room management system. The triage nurse examines the patient complaints and determines their urgency in order to provide health to the patient which needs treatment first. When capacity is available, the patient is accompanied by a nurse to the waiting room. After a certain amount of time, the doctor will see the patient and start the treatment, if necessary. In some cases, a blood test or x-ray scan is needed to determine the cause of the symptoms. When the treatment is finished, the patient will either go home or will be transferred to the ward for further treatment.

## 1.3 Motivation

A paramedic, Don Lundy, once said: "Waiting is good. It means you're not going to die. The person you need to feel sorry for is the one who gets rushed into the ED and treated first." Indeed, he is correct. In most cases, patients with less urgent complaints are the ones that have to wait the longest. Especially when they arrive during the rush hours in the afternoon. Figure 1.4 shows the waiting time for that group of patients. The increasing demand and scarcity of recourses result in longer waiting times. Therefore, the probability increases that patients have to wait too long before proper care is provided.

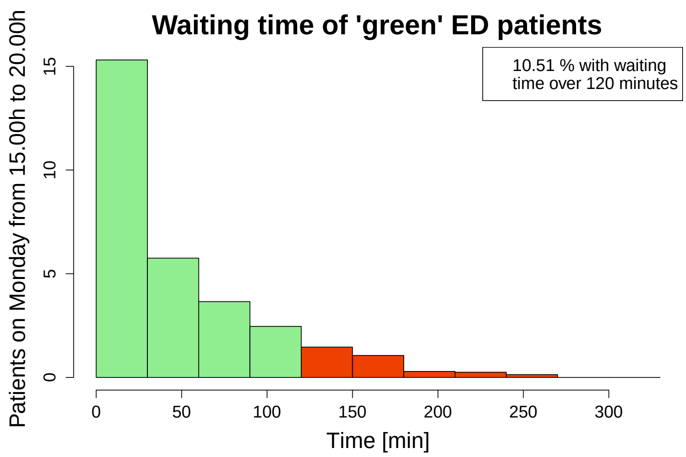


Figure 1.4: Waiting time for 'green' patients at the CZE ED on Mondays between 15:00h and 20:00h in 2011.

This project focusses on the efficiency of the ED of the CZE. The population is aging, health care costs are increasing and thus the need increases to use the resources more efficiently. Clearly, a decrease in waiting time will also improve patients experience and increase the quality of care provided.

## 1.4 Objective

The objective of this project is to give insight in the current efficiency of the ED and supply answers to ‘what-if’ scenarios. Therefore, a simulation model of the ED will be build. This model will be used to investigate the capacity level needed to deliver the health care services within a standard time. The capacity consists of (ED-)physicians, (ED-)medical interns, (ED-)nurses and treatment rooms. Using this model, it should also be possible to address questions such as:

- What capacity is at least required on a typical Monday?
- How much does the waiting time decrease if the number of nurses increases?
- How does the waiting time change if patients arriving by ambulance are treated with the same priority as other patients?

Next, an aggregate model is build to gain more insight in the efficiency of the ED at a single glance. This information can be used to compare different days or to compare the efficiency of this ED with an ED of another hospital.

## 1.5 Related Work

This section starts with an introduction of engineering techniques for health care systems. Next, related work is described that aims to improve the efficiency of Emergency Departments using LEAN tools and simulation models. An extended survey on patient arrival and capacity issues is presented. This sections ends with two related projects carried out for the ED of the CZE.

### 1.5.1 Health Care Systems Engineering

Efficiency of production plants in industrial companies is a well studied subject. The techniques developed in those studies are increasingly applied in health care systems, partly due to the increasing health care costs. More than 12% of the Gross National Product of the Netherlands [10] is spent on health care.

Fowler et al. [7] divides health care systems engineering into the following research areas: Health Care Operations Management (HCOM), Socio-Technical Systems Analysis, Health Care Quality Engineering, Health care Informatics, Medical Decision Making, and Health Care Public Policy. In this project, HCOM is the most applicable one. The objectives of HCOM [7] are: *"The improvement in the efficiency, effectiveness, quality, value, and timeliness of the operations and processes of the system. This means that the primary focus of this area is on the core processes and operations of the health care delivery system"*. For Emergency Departments, HCOM can provide answers to questions such as:

- What is the best staffing level for the ED, for each day of the week?
- How can waiting times in the ED be reduced?

The HCOM research area covers a wide range of topics. One project in this area is the patient transit project. In a hospital, orderlies transfer patient for example from the treatment area to the ward. To optimize this process, Timmermans [19] describes a simulation model to experiment with different orderly dispatch policies and to assess waiting time for transport, before introducing them in the real world.

The dissertation of Zonderveld [22] is another example of a HCOM project. The presented models allow for a quantification of consequences of capacity distribution decisions. Also, a methodology is presented to develop appointment schedules for outpatient clinics with scheduled and unscheduled patients. A cyclic appointment schedule for scheduled patients can be generated given the arrival rates of unscheduled patients.

### 1.5.2 Increase efficiency of the Emergency Department

In Crane [4], applied management science and ED experience are combined to create models of how to improve emergency department operations. To increase efficiency, the LEAN principles of Toyota are integrated in the ED. As mentioned before, it is key to utilize capacity in an efficient way, see Figure 1.5. Because the arrival rate differs during a day, week or month, a smart use of capacity is asked.

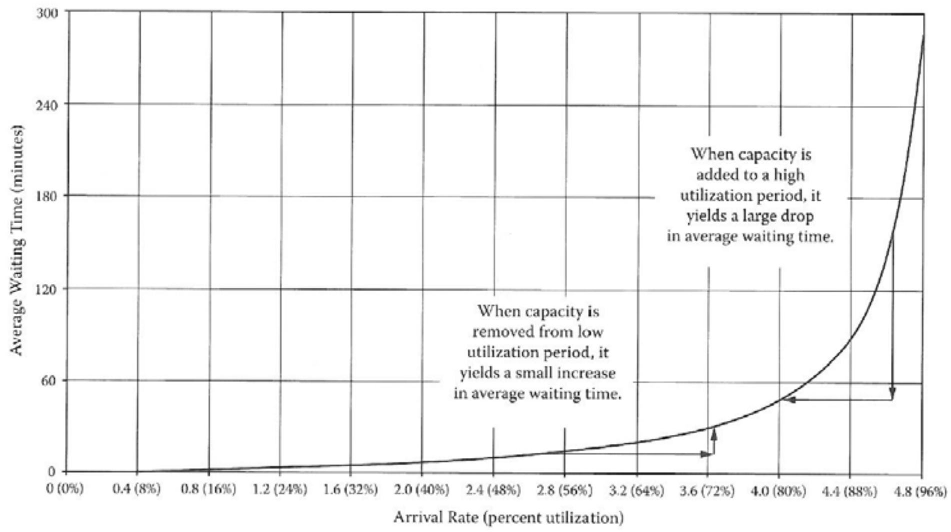


Figure 1.5: An example of an waiting time curve and the conceptual effects of aligning capacity with demand.

The LEAN concept is also used in the CZE to improve operational processes [21]. The aim is to eliminate unnecessary operations to archive better performance using existing resources.

The methods of Crane [4] are also used at CC Zorgadviseurs [12], a consultancy office in health care. Their goal is to improve an ED by reducing errors in processes and optimizing the flow. Queueing theory, capacity analyses and simulation models are used to support those goals.

Visser [20] developed a conceptual model for an IEP in Almelo. The conceptual model is mostly based on assumptions, based on the experience of staff members and based on historical data. Next, the conceptual model is one to one translated into a simulation model. The model is built to test organizational interventions and to analyse the results. The objective of the study of Grummels [11] is to investigate the applicability of simulation optimization for resource allocation in hospitals. Therefore, a theoretical discrete-event simulation model of an ED is created to apply the techniques. Grummels [11] concludes that the response surface method approach, using a rational function regression model, is applicable for optimization on resource allocation problems in hospitals regarding the patient flow.

Data mining techniques are used by Ceglowski et al. [3] to calculate the cumulative weighted impact on different patient types. This results in an overview of ED workload and it identifies critical interactions between the ED and hospital wards. The patient types are generated according to their urgency, disposal and treatment. In this study, 99% of the patients can be categorized in 161 of the 401 possible combinations of urgency, disposal and treatment. These categories are used in the simulation model. Instead of following the physical movement of patients, the simulation tracks the state of ED treatment sites as being 'occupied' or 'free'. Waiting time will occur when all treatment resources are occupied.

### 1.5.3 Patient arrival

Patient arrivals can be divided into unscheduled and scheduled visits. However, the majority of patient of the ED arrive on referral, the exact arrival time is unknown. Thus, patient arrivals for an ED are in general unscheduled. But at the IEP, 'the gatekeeper' can schedule patients, with normal priority, at a later stage.

According to Alexopoulos [2], in previous practice, unscheduled patient arrivals were often assumed to follow an ordinary Poisson process, i.e. the arrivals are independent and identically distributed. But this approach is not generally valid as arrival rates typically vary over time, therefore Alexopoulos [2] proposes a nonhomogeneous Poisson process to model the patient arrivals. The correctness of the patient arrival rate is important to supply the simulation model with useful data.

### 1.5.4 Capacity

The most fundamental measure of hospital capacity is the number of inpatient beds, according to Green [9]. Target occupancy levels are traditionally based on the bed capacity. Most commonly used is the 85% occupancy. Units that face high costs, such as the intensive care unit, are often running at a higher occupancy level. Furthermore, Green [9] states that hospitals still use the number of beds as primary measure however literature provides queueing and optimization models to support their planning. Due to increasing pressure to be more cost efficient, targets above 90% are set without addressing the issues of bottlenecks and congestion.

Also personnel, in particular nurses, is a major component of the capacity. In hospitals, nursing units are often assigned by a specified ratio of patients to nurses. This ratio differs from 1 to 1 at intensive care units to 8 to 1 at normal treatment wards. However, these standards are often exceeded due to the variability of patient numbers over time that leads to inadequate planning. Optimization models can be created to include this effect in planning but according to Green [9], sufficient and useful data is often lacking.

### 1.5.5 Emergency Department CZE

In 2010, two operation research projects where carried out in a collaboration between the TU/e and the CZE. Rijk et al. [15] conducted a statistical analyses on the processing times to point out factors that increase the length of stay (LoS) at the ED at the CZE. Those factors are, among others:

- Waiting for a nurse from the ward to pick up the patient for hospitalisation;
- Patient that have complex complaints who need additional examination.
- Many patients present that need care from one particular specialism.

As mentioned before, not all patients can go home after treatment at the ED and have to stay for hospitalisation. Using historical data, Tacken [16] developed a tool to predict that number of patients. This tool can supply useful information to schedulers of hospital beds, especially when a temporary hold on new intakes is declared. In 2009, this occurred 39 times [16].

## 1.6 Outline

The Emergency Department is described in more detail in Chapter 2. The second part of that chapter analysis the historical data. Chapter 3 explains the simulation model in detail and states the underlying assumptions. Also, the validation of the model is described. The bottlenecks in the current situation are indicated in Chapter 4. Next, in Chapter 5, several scenarios are discussed. Finally, the conclusions and recommendations are given in Chapter 6.

# Chapter 2

# Emergency Department

This chapter explains the ED in more detail. The process of the ED is described in Section 2.1. The patient flow through the ED is described and several possible bottlenecks, identified by ED-employees themselves, are highlighted. The historical data is analysed in Section 2.2. In more detail is looked at several aspect of the data, including the fluctuation of the arrival rate, the type of patients and the treatment times. The section ends with an analysis of the historic waiting times.

## 2.1 System description

This section gives a more extended description of the ED of the CZE. First, the objective of an ED is stated. Next, the patient flow and also the role of physicians and nurses is described.

### 2.1.1 Objective ED

Usually, an ED is found in a hospital on the first floor and has its own entrance. An emergency department aims to provide patients with high quality emergency care as efficiently as possible. Due to the unplanned nature of patient arrivals, the ED must provide treatment for a broad spectrum of illnesses and injuries. Amongst others, patients can suffer from a cardiac arrest, hearth attack, acute exacerbations of respiratory diseases or from trauma injuries such as a broken bone, cuts or internal bleedings caused by a car accident or accidents at home. Obviously, the urgency of these health problems varies from life-threatening to not urgent. The department operates 24 hours a day and staffing levels are typically lower at night due to the lower arrival rate of patients.

### 2.1.2 Patient flow

The patient flow is described using the map of the CZE emergency department, see Figure 2.1, and the patient flowchart, shown in Figure 2.2. Prior to the actual arrival, for most referred patients, it is already know that they will arrive in the near future. The paramedic or general partitioner contacts the senior nurse in order to keep him/her up-to-date. The senior nurse adds a new entry to the electronic hospital information system (EZIS). New patients normally arrive by own transportation or by ambulance. Both patient flows register at the reception. At that time, patients are also logged in on EZIS. This means that the patient is physically present at the ED.

After registration, the patient will take place in the waiting room. Generally, patients go in order of arrival to the triage room to undergo triage. When finished, the patient goes back to the waiting room. If a nurse is free and a treatment room is available, the nurse picks up the patient with the highest priority to accompany him/her to the treatment room.

Paramedics, transporting patients by ambulance, have to wait at the reception before the patient can be dropped off at a treatment room. The maximal target waiting time is 15 minutes for these patients. This patient flow is not drawn in Figure 2.2. In the current situation, patients arriving by ambulance are served prior to patients in the waiting room.

When the patient has arrived in the treatment room, the treatment process starts and the nurse ensures that the patient is installed properly in the emergency room. In some cases, the nurse already starts up a few small examinations such as taking a blood sample. Next, a physician visits the patient for a first evaluation of the complaints, in most cases done by the medical intern. After consultation with a medical specialist, it is decided which extra examinations are needed, e.g. an x-ray. When the tests are finished and the results are reviewed, the physician determines what treatment is needed to cure the patient. If the physician is uncertain about the complaints and how to treat, then a medical specialist of another speciality is paged to examine and treat the patient. During the treatment, the responsible nurse keeps monitoring and nursing the patient when needed.

When the treatment is finished, several things can occur. A patient can go home and the nurse can schedule a follow-up appointment at the general partitioner or at the policlinic. Another option is that the patient has to stay for hospitalization or is transported to another hospital. In those cases, the patient can only leave if a nurse from the ward or a paramedic has arrived to pick up the patient. During the delay that occurs, the treatment room stays occupied and is therefore not available for a new patient.

### 2.1.3 Triage

When multiple patients require emergency care simultaneously but when sufficient capacity is lacking, a choice has to be made to determine which patient gets priority. The system that determines which patient needs treatment first is called the triage. At the CZE, the Dutch Triage Standard is used. This system uses four urgency levels, see Table 2.1. Each level has its own acceptable waiting time. Acute 'red' patients, e.g. patients that suffer from a hearth attack, need to see a physician within 10 minutes. But when a patient has

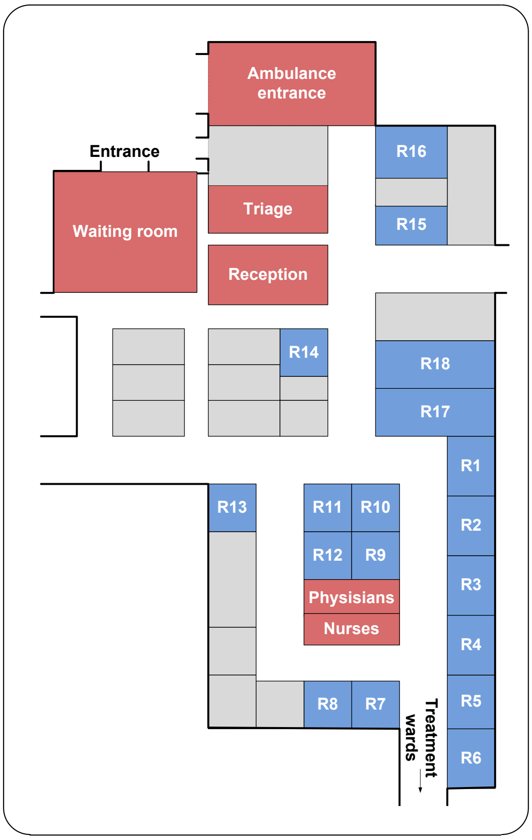


Figure 2.1: Map of the Emergency Department of the CZE.

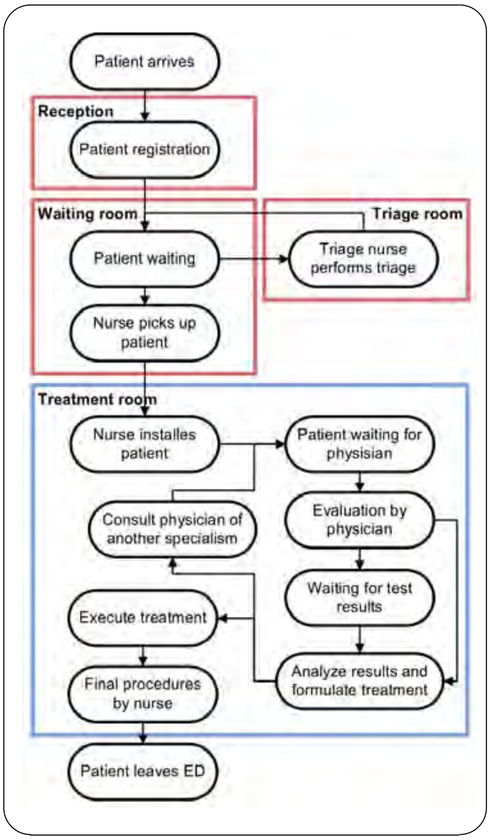


Figure 2.2: Schematic view of the patient flow in an ED.

a broken collarbone, he usually gets assigned to the green level. This injury is less urgent and it is therefore justified to treat other patients first. During (extended) office hours, the triage is preformed by a dedicated triage nurse. The senior nurse is responsible for the triage process outside office hours. The maximum target time of the triage is 10 minutes.

| Urgency level | Time [min] |
|---|---|
| Acute | 10 |
| Urgent | 60 |
| Standard | 120 |
| Not urgent | 360 |

Table 2.1: Maximal acceptable waiting time.

Next, the Dutch Triage Standard also helps to determine the appropriate health care provider. Desired effects of this method are adequate use of resources, improvement of patient logistics and clearer communication. At the CZE, 'red' patients are treated first. When no more 'red' patients are present in the waiting room, the 'yellow' patients are first in line to be treated. Next, 'green' patients are queued, followed by 'blue' patients. In some cases, an exception is made. For example, several 'green' patients, that have a waiting time that is significantly larger than the acceptable waiting time, get priority above newly arriving 'yellow' patients. Another exception is made when the treatment rooms are filled with many patients of one particular specialism. Then, the choice can be made to first treat a patient that needs care from a different capacity group. The health staff at the CZE agreed to triage all patients, even when the waiting room is empty. In most cases when the patient can be treated immediately, the triage takes place at the treatment room instead of the triage room.

### 2.1.4 Nurses

At the ED, a nurse can have several tasks. Besides the nurse assigned for the triage process, there are typically 3 to 5 emergency nurses that nurse and escort patients. One nurse is assigned as senior nurse. About 50% of the time, the senior nurse still performs nursing tasks. Next, (s)he takes care of the operational logistic part of the ED. The senior nurse assigns nurses to patients and regulates the patient flow from the waiting room to the treatment room. Due to the large variety of patient complaints and tasks that have to be performed, the nurse to patient ratio varies from 2:1 to 1:4. The 2:1 ratio usually occurs for the first 15 to 45 minutes of the treatment of a red patient. After that, the patient requires the same amount of capacity as other patients.

### 2.1.5 Physicians

An ED is often compared with a pigeon house, physicians from different specialism come and go to treat patients. The top 7 medical specialities are; general surgery, internal medicine, cardiology, orthopedics, pediatrics, lung specialism and neurology. On average, neurology receives each day about 4 patients but the number of patients per day that is assigned for general surgery often exceeds 30.

Besides the medical specialists, also emergency physicians operate at the ED. These physicians are a relatively new phenomena in the Netherlands. In contrast to most other physicians, the emergency physician is fully assigned to the ED and has no activities in other departments of the hospital.

### 2.1.6 Bottlenecks

As mentioned before, one of the goals of this project is to identify bottlenecks. To conclude this section, several possible bottlenecks, that are already identified by ED-employees, are listed below in random order;

- When a patient has to stay for hospitalization, a nurse from the wards has to pick up the patient. The maximal target time is 15 minutes but this time is often exceeded.
- The medical specialists are not fully dedicated to the ED. When a patient arrives, physicians cannot always go immediately to the ED because of their other activities

such as patient or colleague meetings and the daily round at the treatment ward. This can cause substantial delay.

- In the current situation, most patients stay in the treatment room while waiting for test results. Meanwhile, the room is not available for other patients.
- If a physician is not able to cure the patient, then a physician from another specialism is consulted. For this second consult, the treatment loop starts again.

## 2.2 Data analysis

As mentioned in Section 2.1, the electronic hospital information system EZIS is used by the ED. Mainly to record the patient flow and to view and modify patient records. The main screen of the ED section of EZIS shows all 18 treatment rooms and the details of their current occupation. It also shows the expected patients and the patients waiting in the waiting room. In this section, data from EZIS is used to perform a numerical analysis of the department in order to get an impression of the important factors at the ED.

### 2.2.1 Origin of patients

The Catharina hospital is located in the north of Eindhoven and Figure 2.3 shows that most of the 34.000 patients that visited the ED in 2011, live in urban areas relatively close to the hospital.

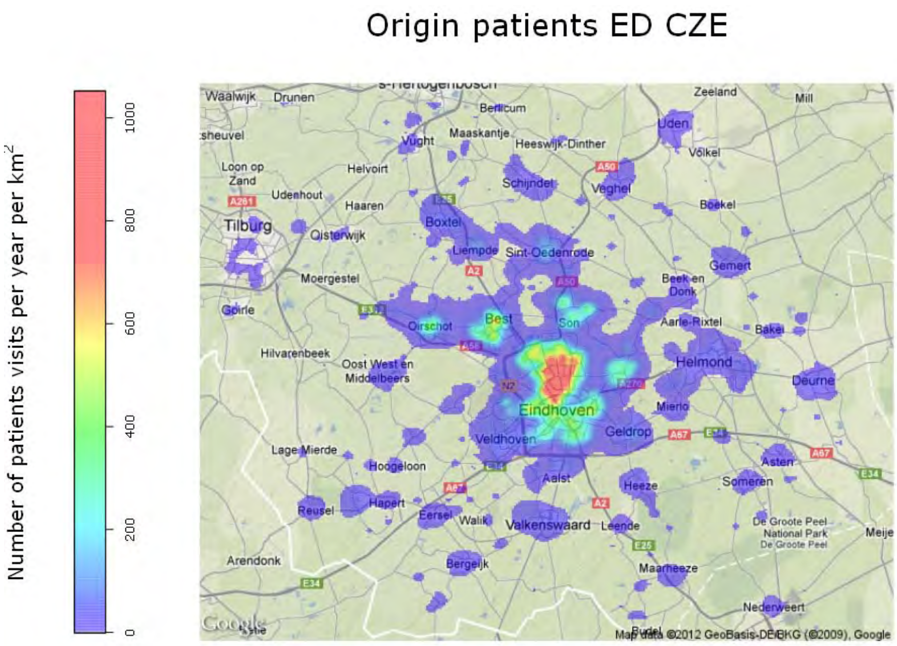


Figure 2.3: Origin of patients of the ED in 2011.

### 2.2.2 Patient arrival process

The patient arrival rate varies over time. Obviously, at night less patients arrive than during office hours. But there is also a significant difference in arrival rate between the different weekdays, see Figure 2.4. Monday and Friday are the most busy days in terms of arrival rate. The lowest arrival rates occur during the weekend days. The rates on Saturday and Sunday as well as the rates on Tuesday to Thursday are corresponding.

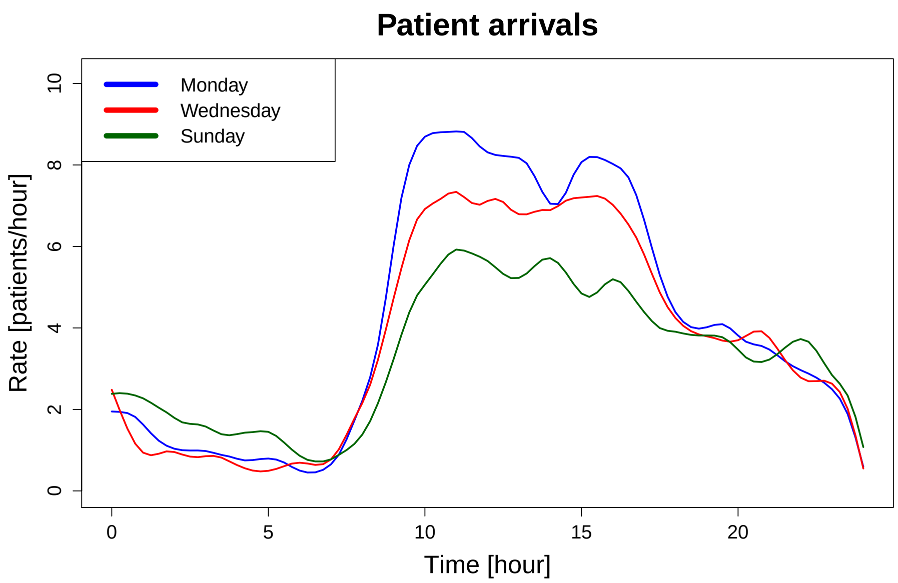


Figure 2.4: Patient arrival rates on an average Monday, Wednesday and Sunday in 2011.

As mentioned in Section 1.5, Alexopoulos [2] proposes a nonhomogeneous Poisson process to model the patient arrivals. This approach is also applicable in this situation because the average arrivals per hour differs over time and for each hour, the arrivals are Poisson distributed. For each weekday and each hour of the day, a different arrival rate can be used in the simulation model.

### 2.2.3 Diversity of patients

As input for the ED simulation model, not only the number of arriving patients has a large influence, but also the patient details are important. The first attribute of a patient that is discussed is the medical speciality. By assigning a specialism to each patient, it is possible to address capacity issues per medical speciality. Also, treatment times are dependent on the specialism, see Section 2.2.5.

In 2011, over 20 different medical specialities were consulted. In our simulation model, only the 11 most visited specialities are included. The ones that are left out have, on average, less than one patient visit per day. The included specialities are surgery, internal medicine, cardiology, orthopedics, pediatrics, lung diseases, neurology, urology, gynecology, plastic surgery and geriatrics.

Figure 2.5 shows the distribution of specialities from Monday to Sunday. Due to agreements between the surgeons and orthopedists, on Tuesdays and Fridays to Sundays more patients are seen by the surgeon. On Monday, Wednesday and Thursday, those patients are seen by the orthopedist. The result of this agreement is the most notable effect in this figure.

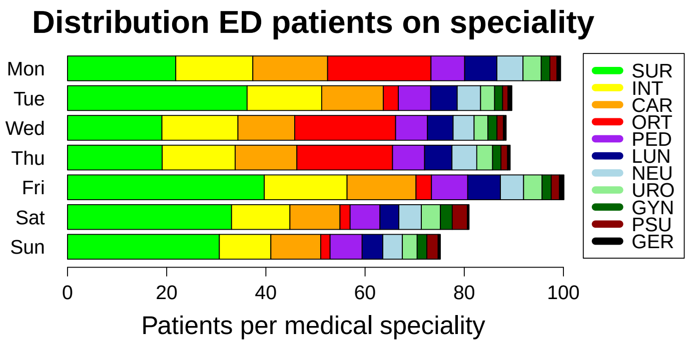


Figure 2.5: Patients per specialism on the different weekdays.

As mentioned before, the triage system is used as a waiting room control system. Therefore, the percentage of respectively red, yellow, green and blue patients have to be included in the simulation model. This percentage varies over the different medical specialities, see Figure 2.6.

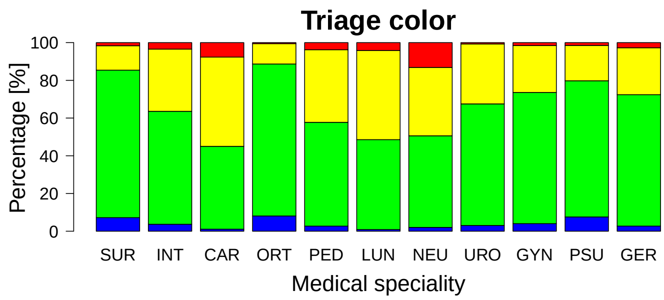


Figure 2.6: Triage color distribution per speciality.

In 2011, 12% of the patients arrived by ambulance. As mentioned in Section 2.1.2, these patients are served first. The other patients arrive by own transportation. After treatment, 31% of the patients is hospitalized.

Also the age of the patient is an attribute that is taken into account. The patients are divided in six age groups, see Table 2.2. Generally, the treatment time increases with the age of the patient.

| Age range | Number of patients | Average treatment time [min] |
|---|---|---|
| 0 to 3 | 1983 | 92 |
| 4 to 16 | 3552 | 82 |
| 17 to 35 | 7006 | 96 |
| 36 to 55 | 7482 | 111 |
| 56 to 75 | 8630 | 124 |
| 76 + | 6171 | 142 |

Table 2.2: Patients per age group.

### 2.2.4 Type of physician

As mention before, the attending physician can either be a medical specialist or an emergency physician. Basically, emergency physicians are always present at the ED. They do not have any commitments to visit other departments. On the other side, the medical specialists do have that commitment. By definition, their home base is not the ED. They still have to visit their patients in the wards and are therefore not always immediately available at the ED. The speciality internal medicine is the only exception.

A medical intern is stationed at the ED on weekdays during office hours. The data on availability is used to create rosters as input for the simulation model. Information on how the roster is created can be found in Chapter 3.

Next to the presence of the physicians, also the choice for the attending physician is important input for the simulation model. Generally, patients that attend the specialty Orthopaedics are seen by the emergency physician in more than 85% of the cases. However, 85% patients for cardiology and gynecology are seen by their own medical specialist.

The group of the top 11 most visited specialties is extended by splitting the patient group that attends the medical speciality General Surgery. The criteria to split this group is whether or not the complaints are caused by trauma or injuries. The reason for this split is that patients with trauma are more often seen by the emergency physicians and patients without trauma consult the surgeon.

When a physician cannot find the cause of the health issue or when a physician identifies that another medical specialist has a better background to treat it, a second consult is requested. The average treatment time for patients that get a second consult is 184 minutes which is 72% higher than the treatment time of other patients.

The probability that a second consult is requested is dependent on medical speciality and the triage color. Next, also based on historical data, a matrix is created to determine which medical specialist performs the second consult if the first consult is carried out by another particular specialist.

### 2.2.5 Distribution of treatment time

The treatment time is used as input for the ED simulation model. As mentioned in Section 2.2.4, the treatment time increases significantly if a second consult is conducted. Figure 2.7 shows the distribution of the historic times. The gamma distribution is found the most suitable to fit the historical data. A gamma distribution is a two-parameter family of continuous probability distributions and is characterized by a shape and scale parameter.

Besides the second consult, also the medical speciality, the triage color, the age, the type of attending physician (ED-physician or other medical specialist) and the number of patients that the physician treats at that moment (PIP) are factors that have a significant influence. Table 2.2.5 shows for all factors and categories the mean and variance of the treatment, if all patient treatment times are only split on one factor.

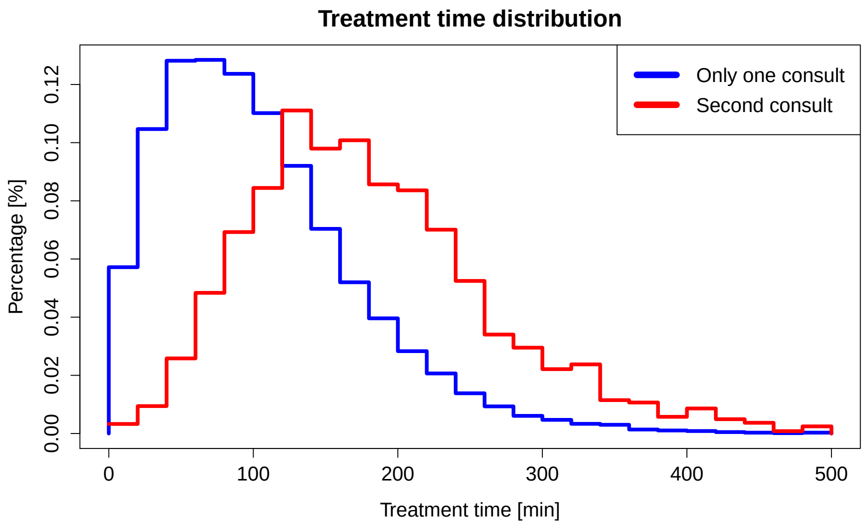


Figure 2.7: Distribution of the historical treatment time. The treatment for patients which have only one consult takes less time than the treatment for patients that need a second consult.

### 2.2.6 Waiting time

In the ED simulation model, the inputs are the patient profile, the treatment time and the patient pathway. The waiting time, one of the outputs of the model, is discussed in this subsection. The distribution of waiting time on a Monday is shown is Figure 2.8. For the majority of patients, the waiting time is less than one hour.

As described in Section 2.1.3, each urgency level has its own maximal target times for the waiting time. On average, these target are easily met, see Table 2.4. But, as discussed in Section 1.3, green patients arriving during peak hours have to wait a relatively long time. During this hours, these patients are more likely to exceed the maximum target time. Figure 2.9 shows that this also holds for ‘yellow’ patients.

| classification | | mean | variance |
|---|---|---|---|
| Second consult | No | 107 | 5874 |
| | Yes | 184 | 7927 |
| Type Physician | ED | 98 | 5180 |
| | other | 133 | 5842 |
| Specialism | SURt | 153 | 4606 |
| | SURn | 90 | 5447 |
| | INT | 160 | 6946 |
| | CAR | 115 | 7245 |
| | ORT | 83 | 4014 |
| | PED | 105 | 4830 |
| | LUN | 157 | 4767 |
| | NEU | 127 | 5576 |
| | URO | 116 | 4561 |
| | GYN | 114 | 7522 |
| | PSU | 75 | 4310 |
| | GER | 141 | 4409 |

| classification | | mean | variance |
|---|---|---|---|
| Triage | Red | 126 | 5035 |
| | Yellow | 141 | 6271 |
| | Green | 106 | 5679 |
| | Blue | 67 | 4278 |
| Age | 0-3 | 93 | 4340 |
| | 4-16 | 82 | 4282 |
| | 17-35 | 96 | 5819 |
| | 36-55 | 112 | 6478 |
| | 56-75 | 123 | 6303 |
| | 76+ | 142 | 6891 |
| PIP | 0 | 116 | 7816 |
| | 1 | 117 | 5834 |
| | 2 | 118 | 6238 |
| | 3 | 112 | 5785 |
| | 4 | 103 | 5407 |
| | 5+ | 100 | 4919 |

Table 2.3: The mean and variance of the historical treatment times if all patient treatment times are only split on one category.

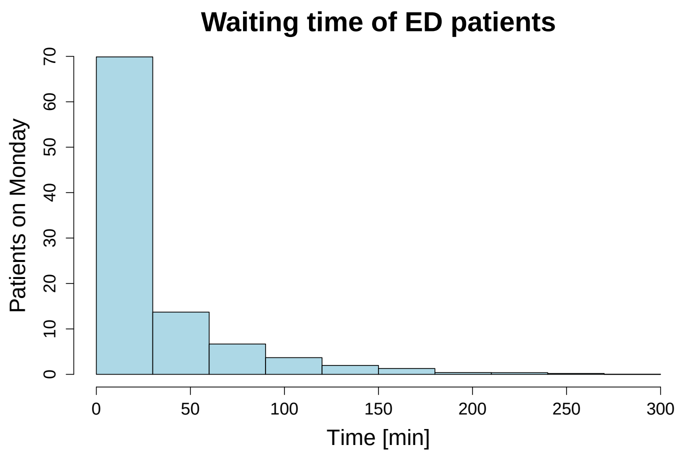


Figure 2.8: Waiting time for patients at the CZE ED on Mondays in 2011.

| Arrived by | Ambulance | Own transportation |
|---|---|---|
| Red | 4 m 57 s | 9 m 20 s |
| Yellow | 6 m 13 s | 15 m 15 s |
| Green | 7 m 22 s | 27 m 28 s |
| Blue | 12 m 43 s | 26 m 6 s |

Table 2.4: Average waiting time per triage color.

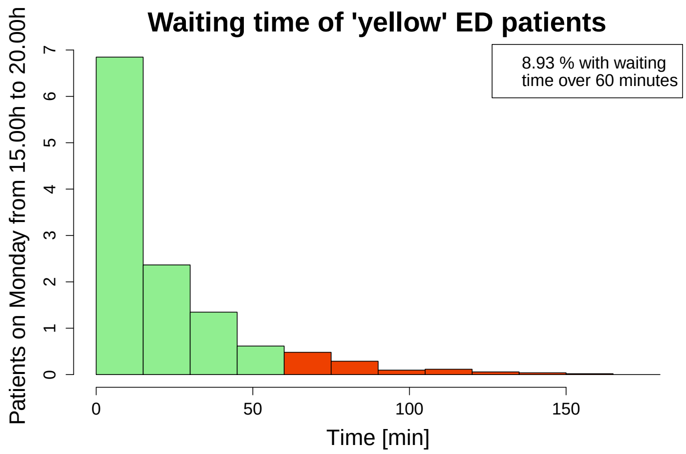


Figure 2.9: Waiting time for 'yellow' patients on Mondays between 15:00h and 20:00h in 2011.

### 2.2.7 Conclusion

In Section 2.1, the current way of working of the ED is described. This section shows an overview of the data analysis that is conducted. It highlights important aspects such as the attributes of patients and the distribution of treatment times. In Chapter 3, the simulation model is explained.

# Chapter 3

# Modelling ED

This chapter starts by discussing the software package that is developed. Besides the simulation model, the software package also includes a tool to analyse the historical and simulation data. Next, the modelling choices and model assumptions are discussed. In Section 3.3, the components of the simulation model are discussed in detail. This chapter ends with the validation of the simulation model.

## 3.1 Software package

The software package exists of three parts, see Figure 3.1. The first program that is required is Microsoft Excel 2007 [13]. An Excel file contains the 2011 historical data and is able to generate all the input files that are used by the other tools.

The simulation model is build in Chi 3.0 [1]. This simulation tool is developed by the Manufacturing Networks Group at the TU/e. Chi 3.0 is a plugin for the Java IDE Eclipse Juno with Modeling Tools. The Modeling Tools require the Java Development Kit [14], which is also included in the software package.

Third, the statistical package R, in combination with R-studio [6], is used to evaluate the historical and simulation data. More information on the use of the software package can be found in the tool manual [18].

## 3.2 Modelling choices

In this section, the modelling choices and assumptions are discussed. First, the way of modelling the arrival of patients is explained. Next, the assumptions regarding the nurse and physician capacity are discussed. Also, the waiting room management choices are explained. The section ends by explaining the assumptions made for the treatment process.

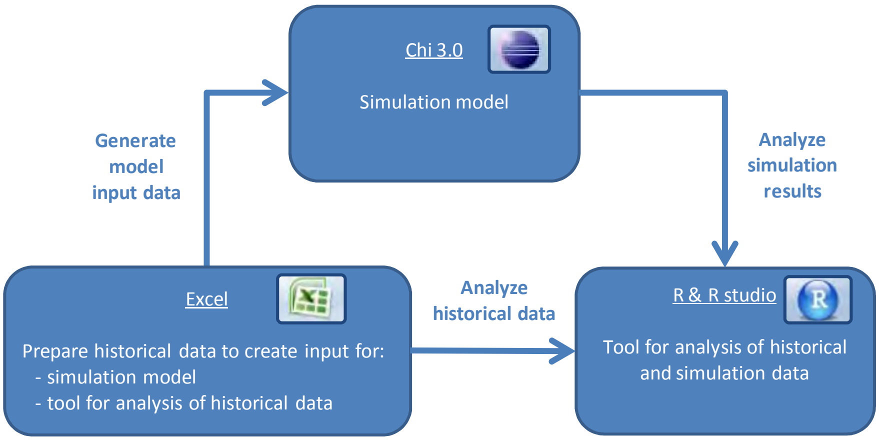


Figure 3.1: Software used and linked.

### 3.2.1 Arrival process

As mentioned in Subsection 2.2.2, the arrival rate is different for each day of the week and the attending physician also differs per weekday, see Figure 2.5. Therefore, the choice is made to generate separate simulation input files for the different simulation days and to simulate them separately. Each simulation day will start with an empty system.

For each hour of the day, the rate at which patients arrive can be different but constant during that hour. The time between two arrivals is assumed to be exponentially distributed, i.e. arrivals occur according to a Poisson process. If the time of arrival exceeds the current hour, then no more patients are created until the next hour begins. In the simulation at the boundaries of hours, the arrival rate of the next hour is used to determine the time of the next arrival.

The attributes of the patients arriving by ambulance are different from the attributes of the other patients. Also, their patient care path is different. The diversity of patients has been discussed in Subsection 2.2.3. If a new patient arrives, the medical speciality that is chosen depends on whether the patient has arrived by ambulance or not. Next, the triage color is sampled and is dependent on the specialism and also on the type of arrival. The third attribute that is picked is the age group, depending on specialism. Next, the required capacity is determined by assigning tokens, see Section 3.2.2. Last, depending on specialism and triage color, the choice is made whether or not a patient needs a second consult. The physician that performs the second consult depends on the physician who conducted the first consult. The distributions used to determine the attributes of a patient are generated using the historical data.

After registration, the patient enters the waiting room. Next, the senior nurse has to take note of the patient before taking action. Due to this delay, it is possible that a patient has to wait, although the required capacity is already available. The delay is modelled as an exponential delay. For red patients, the need for more urgent care results in faster response time. Therefore, their modelled delay is smaller than the delay of other patients.

### 3.2.2 Modelling resource capacity

A token system is used to model the capacity of the physicians, nurses and triage nurses. To start the treatment, a patient claims a combination of tokens representing the resources that are needed. Four nurse tokens are used to represent one nurse because (s)he can treat a maximum of four patients at the same time. So each patient needs one nurse token. Moreover, the triage nurse, senior nurse and physician are modelled as respectively one, two and two or three tokens. So, the simulation uses for example 20 nurse tokens to simulate five nurses.

Each patient demands one triage nurse, one nurse and one physician token. An exception is made for red patients. They demand more intensive care for the first 15 to 30 minutes of their treatment. A triangle distribution is used to sample the length of the intensive treatment. Therefore, all red patients claim more tokens for the first part of the treatment.

As mentioned before, the arrival rate is different for each hour of each day of the week. Moreover, there is also a different working roster for each weekday. One of the simulation input files contains the schedules. It specifies the new total capacity for each time the roster changes. For example, the staffing levels are lower at night. Next, also the transfer from night to morning shift is taken into account by decreasing the available capacity between 7:30h and 9:00h.

### 3.2.3 Waiting room management

The waiting room management system is discussed in this subsection. In the ED model, the waiting room is used to keep track of the available capacity. The waiting room process will be notified if a roster update occurs or when capacity is claimed or released by a patient. Using this information, the waiting room process can perform the managing role of the senior nurse. It determines when the triage or treatment of a patient can start.

The triage process will serve red and yellow patients in order of arrival. So, the triage process treats red and yellow patients with priority over green and blue patients. The treatment dispatch policy is more complex. Generally, the patients are dispatched according to 1) their triage color and 2) their time of arrival. Red patients have the highest priority and blue patients have low priority. The patients are dispatched as soon as the required capacity is available for treatment.

In some cases, an exception is made. If there is not enough available capacity to serve the most urgent and longest waiting patient. So the queue of waiting patients will be scanned from head to tail until the first patients is found for which capacity is available. For example, if a room becomes available and already three treatment rooms are filled with patients that want to see the cardiologist, an orthopedic patient is preferred above a cardiologic patient.

Also, due to their urgency, red patients are always served as fast as possible. Therefore, in the simulation model, the capacity restrictions are not applicable to red patients. Next, the model includes the aim of the senior nurse to keep one treatment room unoccupied in case of emergency. Only a red patient can enter the last unoccupied room.

### 3.2.4 Triage and treatment

The triage is represented as a simple delay. This delay is sampled from a gamma distribution. Historical triage times are used to determine the parameters of the distribution. A triage can start is the triage room and triage nurse are available. When the triage is completed, the waiting room process is informed that the triage nurse and triage room are available again.

The treatment rooms are modelled individually and each room can be occupied by one patient. The treatment itself is modelled as a time delay for the patient, an occupation of nursing and physician tokens and a occupation of the treatment room. The number of available tokens is decreased for the entire treatment time. How the physician or nurse exactly organizes their time between patients that are simultaneously present at the ED is not modelled. So, the number and duration of actual visits during the treatment is not taken into account.

The combination of the factors shown in Table 2.2.5 leads to 6912 different treatment group. As mentioned before, the ED is visited by 34.000 patients in 2011. That gives on average 5 patients per group and makes it unreliable to fit gamma distributions for these groups. To cope with this problem, the data mining technique of recursive partitioning is used. Groups that have equal treatment times are not split. The package 'rpart' [17] is used by the software program R [8] to generate a decision tree.

Figure 3.2 shows the first part of this tree. In each decision step, the group is split into two groups by the factor that has the most influence on the mean treatment time. The 'n' in an end node denotes the number of patients that fit into that group. The 't' is the average treatment time. This tree divides the patients into 37 groups. For each group, a gamma distribution has been fitted on the treatment times. The entire tree can be found in Appendix A.

For each of the 6912 mostly bad conditioned groups, the decision tree end node is determined. The calculated mean and variance of the historical treatment times of all patients in each node are coupled to their corresponding original group. As a result, many of the 6912 groups use the same mean and variance but no group is bad conditioned.

So, the treatment time is sampled from a gamma distribution. The treatment time is dependent on the medical specialism, triage color, age group, patients in process, type of physician and whether or not the patient needs a second consult. These factors determine which parameter values are used to sample the patient treatment time.

Another option was a more aggregated implementation of the treatment times by using just one mean and one variance for all patients. But that implementation disables the opportunity to trial scenarios such as 'What influence has an average 15 minutes decrease of the treatment time of cardiology patients on the waiting time?'.

As mentioned in Section 3.2.2, red patients need more capacity for the first part of their treatment. If this first part is finished, the capacity is reduced to normal level, i.e. to one nurse and one physician token. Next, for patients that need a second consult, the physician capacity is changed when the treatment is halfway.

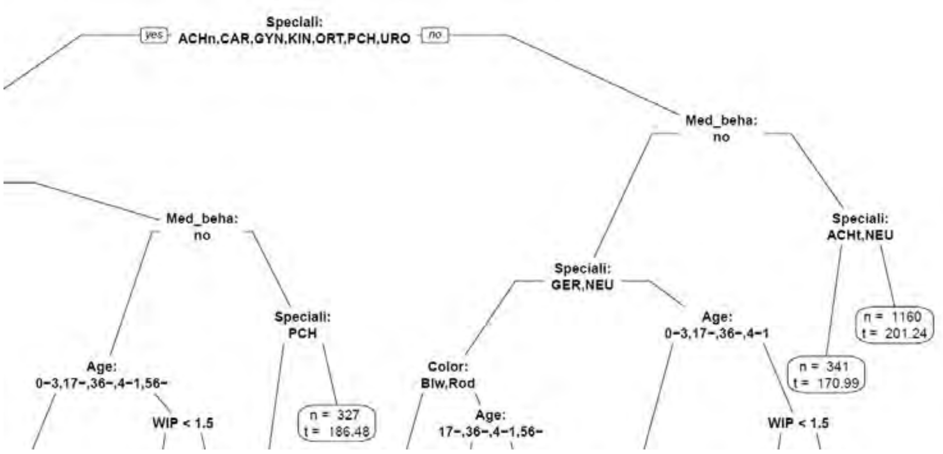


Figure 3.2: Part of the treatment time decision tree.

## 3.3 Simulation model

This section starts with an global description of the simulation model. In the following subsections, the individual parts are discussed in more detail.

### 3.3.1 Model setup

The model is able to simulate multiple days in one run, see Figure 3.3, each day being a specific day of the week, e.g. all Mondays. This is needed due to the stochastic nature of this problem. Some days are more busy than others, for example in terms of arrival rate or intensity of care. The process *SimOneDay* represents one simulation day and when a day is finished, the next day is started with an empty ED.

The simulation initialisation parameters can be adjusted using the input files generated in Excel. Using these parameter, it is possible to change the number of simulation days, the simulated day of the week, the number of arrivals per patient group, the treatment time per patient group, the number of treatment rooms and whether or not to give priority to ambulance arrivals.

Next, the process *SimOneDay* is discussed in more detail. This process consists of the generator $G$, the arrival process delay $D$, the waiting room $W$, the triage process $T$, the treatment rooms $R$ and the exit $E$, see Figure 3.4. In this figure, the schematic representation of the ED CZE simulation model is given.

The generators $G^{\mathrm{k}}$ represents the arrival process and creates the individual patients including their attributes with a certain arrival rate. The first generater creates patients arriving by an ambulance and the second generates patients arriving by own transforation. A new patient is sent via $D$ to the waiting room $W$. The delay process $D$ represents the time needed to process a new arrival.

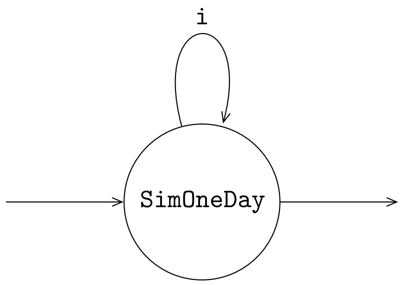


Figure 3.3: Top level of the ED simulation model.

The waiting room process keeps track of all waiting patients and of the availability of the recourses. It uses this information to determine when and which patient will go to the triage process or to the treatment room first. The triage process $T$ receives patients from the waiting room and sends the patients back after a certain amount of time, required for performing the triage. Process $DT$ is used to sample the triage time.

If the treatment can start, the patient is sent to one of the treatment rooms $R^{\mathrm{n}}$. The update process $U$ is used to report staffing changes to process $W$. For example, the staffing levels at night are usually lower than during daytime. The waiting room also receives information from $T$ and $R$ about their availability.

Treatment room $R$ receives a patient and requires nursing and physician capacity. The process $DT$ is also used to sample the treatment time. During or after this delay, the capacity is released again and process $R$ informs the waiting room. When the treatment is finished, the patient goes to the exit $E$. This represents the departure of a patient. The patient either goes home or is hospitalized. The exit process sends information to the print process $P$ which takes care of the simulation output. Next, the print process also signals when the system is empty and a new simulation day can start.

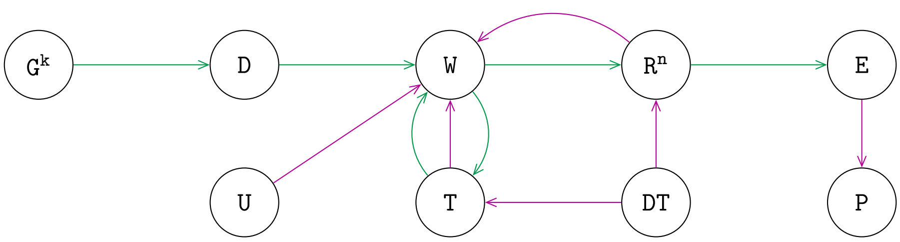


Figure 3.4: SimOneDay — ED model CZE. Patients are transferred using the green channels, other information uses the purple channels.

### 3.3.2 Constants and Types

Before the individual processes are explained, first the constants and data types are discussed. The ED simulation model has four global constants, see Listing 3.1. The line numbers in this and other listings refer to the line numbers in the complete Chi file.

```
1 const real maxreal = 9e9,
2       int  AR      = 24,   # different arrival rates
3       int  DS      = 400,  # distribution intervals
4       int  DC      = 12;   # different specialties
```

Listing 3.1: Chi 3 listing of model constants.

Next, the constants are declared, see Listing 3.2. The type *staff* consists of several fields for capacity of the triage nurse, the nurse, the ED-physician and a list for the other medical specialists.

Next, the *inter* type has fields for a begin and end time. This type is used in the *timelist*. The *timelist* saves the simulation day and the start and end times for the triage, the treatment and the total stay at the ED.

The *patient* type contains fields for the medical speciality, the triage color, the age group, the arrival by ambulance, the treatment room, the staff needed, the patients in process at the beginning of the treatment and the *timelist*. The *stlvl* type is used to send the data of the availability of the staff for a certain day and time to the print process.

The types *roster*, *siminits*, *arrival* and *t_treat* are used to respectively save the daily staffing schedule, the initial simulation parameters, the arrival rate and the treatment time parameters. The *strm* type is used to update the waiting room process when capacity is released. If the boolean *rm* is true, also the treatment room becomes available again.

```
6  type patient   = tuple(int spec, color, age, amb, room, cons2;
7                          staff st, st2, wip;
8                          timelist t),
9       staff     = tuple(int  t, n, s;
10                         list(DC) int p),
11      inter     = tuple(real b, e),
12      timelist  = tuple(int  simday;
13                         inter total, triage, treat),
14      roster    = tuple(list real t;
15                         list staff st),
16      stlvl     = tuple(int day;
17                         real ttime;
18                         staff st),
19      siminits  = tuple(int  weekday, simdays, NR, NRR, amb;
20                         real ArrAll; list(DC) real ArrSpec; list(4) real ArrCol;
21                         list(6) real ArrAge; list(2) real ArrPhy, Arr2con;
22                         int TmtAll; real TmtPer; list(DC) int TmtSpec; list(4) int
                               TmtCol;
23                         list(6) int TmtAge; list(2) int TmtPhy, Tmt2con),
24      arrival   = tuple(list(2)  list(AR) real rate;
25                         list(2)  list(DS) int spec;
26                         list(2)  list(DC) list(DS) int col;
27                         list(DC) list(DS) int age;
28                         list(DC) real phys;
29                         list(DC) list(4)  real cons2;
30                         list(DC) list(DS) int cons2by),
31      t_treat   = tuple(list(DC) list(4) list(6) list(9) list(2) list(2) real a;
32                    list(DC) list(4) list(6) list(9) list(2) list(2) real b),
33      strm      = tuple(bool rm;
34                         staff st);
```

Listing 3.2: Chi 3 listing of the data types.

### 3.3.3 Model

In Section 3.3.1 we explained that the model simulates one specific day of the week multiple times and due to the stochastic behaviour, each day is different. The Chi 3 listings for the model $M$ are given in Listing 3.3. In line 37 to 40, functions are called that read and process the input files. Next, the two files to save the simulation output are opened. The variable *si.simdays* denotes the number of days that is simulated in one run. If *SimOneDay* is started, the processes shown in Figure 3.4 are started. One exception, processes $U$ and $DT$ are initiated in respectively the waiting room process and the triage and treatment process.

```
36 model M():
37    siminits   si = read_inits("simulation_init.txt");
38    arrival   arr = read_arr_input(si);
39    roster    ros = read_schedule_input(si.weekday);
40    t_treat   tmt = read_treatment_input("treatment.txt");
41    file    f_pat = open("resultpatient.txt", "w"),
42             f_st = open("resultstaff.txt", "w");
43
44    for i in range(si.simdays):
45       run SimOneDay(arr, ros, f_pat, f_st, si, tmt, i)
46    end
47
48   close(f_pat);
49   close(f_st)
50 end
```

Listing 3.3: Chi 3 listings of the ED simulation model.

Listing 3.4 shows the Chi 3 listing of the *SimOneDay* process. Two generators are started, one for the ambulance arrivals and one to simulate the arrivals of the other patients, which are referred by *j*. Next, also *si.NR* treatment rooms are created.

```
proc SimOneDay(arrival arr; roster ros; file f_pat, f_st;
             siminits si; t_treat tmt; int i):
   chan patient a, c, d, pe;
   list(2) chan patient b;
   chan stlvl pst;
   chan void pa, e;
   chan strm sa;
   chan real dt;

   run
      unwind j in range(2):
         G(a, i, j, arr, pa, time, si)
      end,
      D(a,d, e),
      W(d, b, pst, sa, e, ros, i, si),
      T(b[0], a, sa, e, dt),
      unwind k in range(si.NR):
         R(b[1], c, sa, e, dt, tmt, si, k)
      end,
      E(c, pe, e),
      P(pa, pe, pst, e, f_pat, f_st, i, time, si.NR)
end
```

Listing 3.4: Chi 3 listings of the 'SimOneDay' process.

### 3.3.4 Read input files

The listings of the functions that read the input files can be found in Appendix B starting from line 69 until 296. The first function is *siminits*. This function is used to read the simulation parameters such as the number of simulation days, the number of treatment rooms and the factor to increase or decrease the treatment time to trial different scenarios.

The function *roster* reads the roster input file and generates the daily schedule for this simulation day. For each moment in time that the roster changes, the input file contains the maximal number of tokens available for each capacity group. The arrival rate is read in the function *arrival*. The distribution of the arrival rate for patients arriving by ambulance is separated from the arrival of other patients. Also the distributions of the patient attributes are read in this function. The input file is constructed according to the results found in Section 2.2.

Last, the treatment time distribution is imported by the function *t_treat*. The file is constructed using the decision tree from Section 2.2.5. Each patient group is coupled to an end node of the tree and the mean and variance of that node are assigned to sample the treatment time for that group. More information on these files is given in the software package manual [18].

### 3.3.5 Generator process

The Chi 3 listing for the generator process $G$ is given in Listing 3.5. As mentioned in Section 3.3.3, the generator is started twice. From line 339 to 346, the distributions are created to assign the attributes to a simulated patient. The variable $t$ denotes the time until the next arrival. This time is sampled using an exponential distribution and the arrival rate of the current hour. The generator generates new patients for 24 simulation hours.

A new patient is created when the time until the next arrival is passed. The function *CreatePatient* is used to create a new patient and samples the necessary patient attributes such as the specialism, triage color and age group. The distributions are saved in a cumulative format such that the attribute can be determined by sampling an integer value between 0 and *DS*. An example is shown in Figure 3.5, the triage color for this patient becomes green.

In the *CreatePatient* function, also the start time is added to the patient information. Next, the capacity needed by the patient is determined. If the triage color is red, the patient demands the first part of the treatment more intensive care. Therefore, the patient claims more capacity. Also, the type of physician is determined. If the sample $dp$ does not exceed a certain threshold, then the patient treated by a medical specialist. Otherwise, the ED-physician performs the treatment.

When also these final values are determined, the patient is ready to be sent to process $D$ via channel $a$. The generator also informs the print process $P$ that a new patient has entered the system. The process ends by determining the delay to the next arrival. If the next arrival time exceeds the current hour, the process waits until this hour is finished and takes

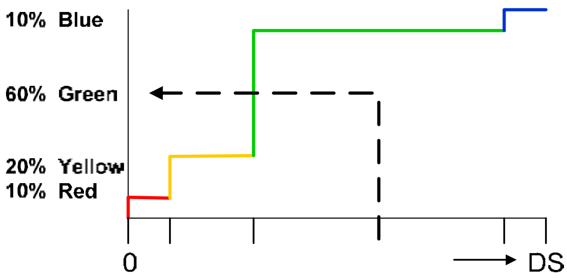


Figure 3.5: Sample patient attributes.

a new sample using the arrival rate of the following hour.

```
proc G(chan! patient a; int i, amb; arrival arr;
       chan! void pa; real tstart; siminits si):
   patient     x;
   int         n;
   real        t, h = 60.0;
   dist real   darr  = exponential(1.0);
   dist real   phys  = uniform(0.0, 1.0),
               cons2 = uniform(0.0, 1.0);
   dist int    age   = uniform(0, DS),
               spec  = uniform(0, DS),
               color = uniform(0, DS),
               co2by = uniform(0, DS),
               dplus = uniform(2, 6);

   t = sample darr * h / arr.rate[amb][0];

   while ( time - tstart + t ) < ( h * 24 ):
      delay t;

      x = CreatePatient(arr, sample spec, sample color, sample age, sample dplus,
                        amb, sample co2by, i, sample phys, sample cons2, time, si)
                            ;
      a!x;
      pa!;

      n = floor((time - tstart) / h);
      t = sample darr * h / arr.rate[amb][n];
      while floor( ( time - tstart + t ) / h ) > n and n < 23:
         n = n + 1;
         t = sample darr * h / arr.rate[amb][n];
         delay tstart + h * n - time;
      end
   end
end
```
Listing 3.5: Chi 3 listings of the generator process.

### 3.3.6 Delay process

The Chi 3 listing of the process $D$ is given in Listing 3.6. This process represents the delay to initiate the care pathway. A timer is set for arriving patients. If this timer is finished, the patients is sent to the waiting room. For red patients, the delay is two times smaller. If the simulation day is finished, a void is send via channel $e$ to terminate the process.

```
proc D(chan patient a,d; chan void e):
   real m = 8.0;
   dist real dtransp = exponential(m);
   list patient xs;
   patient x;
   list timer ts;
   bool go = true;
   real dt;

   while go:
      select
         a?x:
            xs = xs + [x];
            dt = sample dtransp;
            if x.color == 0:
              dt = dt / 2
            end
            ts = ts + [timer(dt)]
      alt
         unwind i in range(size(xs)):
            size(xs) > 0 and ready(ts[i]), d!xs[i]:
               xs = xs - [xs[i]];
               ts = ts - [ts[i]]
         end
      alt
         e?:
            go = false
      end
   end
end
```

Listing 3.6: Chi 3 listings of the generator process.

### 3.3.7 Waiting room process

First, the roster update process is discussed, see Listing 3.7. This process is initiated by process $W$. The relative roster modification is sent to the waiting room process each time a schedule update occurs.

```
proc U(chan strm sa; roster ros):
   for i in range(1,size(ros.t)):
      delay ros.t[i] - ros.t[i-1];
      sa!(false, ros.st[i])
   end
end
```

Listing 3.7: Chi 3 listings of the roster update process.

The Chi 3 listing of the waiting room process is shown in Listing 3.8. The process starts by declaring some variables. First, $dc$ denotes which capacity groups have to be taken into account and whether it is an addition of capacity or a substraction. From left to right, the capacity groups are triage nurse, nurse, ED-physician and medical specialists.

Next, $xss$ represents 12 parallel waiting lines. The last four lists are reserved for patients that arrived by ambulance, divided by the four different triage colors. Other lists contain patients that arrived by own transportation. The first four are used for patients who have not been triaged, the second four lists contain patients after triage. The variable *cnts* keeps track of the number of patients in each list.

The two lists of two integers $k$ contain the waiting list and position number of the patient which is ready for respectively triage or treatment. To determine these patients, the functions *dispatch_triage* and *dispatch_treatment* are used. Beside the information of the waiting patients and of the capacity type *dc*, also the current available capacity is needed. Moreover, to determine the dispatch for treatment, the number of available treatment rooms *ra* and the number of rooms reserved for red patients *rr* are needed.

For triage, the first patient that is selected is the first arrived red or yellow patient. If none, the first arrived green or blue patients is selected. The function ends by checking if the needed capacity is also available using function *st_check*. If the capacity is not available or if there is no patient selected at all, the number 12 is returned.

First, the *dispatch_treatment* selects waiting patients arrived by ambulance. If there are none, the first arrived patient with the highest urgency is selected. Again, function *st_check* is used to determine if the capacity is available for the selected patient. If not, the function checks if capacity is available for the second patient of the same list. If none of the patients can claim enough capacity or if the number of available rooms is smaller than *rr*, the function returns the number 12. An exception is made for red patients.

After determining the patients to be dispatched, priority is given for triage above treatment in line 557 to 559 if the same patient is selected. The current number of patients in the treatment rooms is saved in *wip*. The waiting room process keeps track of this information and links it to the dispatched patient for treatment. This information is needed to determine the treatment time.

Next, five alternatives can be selected. The first option is to receive a patient via channel *a* from processes *D* and *T*. The patient is added at the proper position of *xss* and *cnts* is increased. Second and third, a patient can be sent for triage or treatment. The needed capacity is claimed using function *st_update*. This function uses *dc* to determine which capacity groups are claimed or become available again. Also, the patient is removed from the list *xss* and from *cnts*. Finally, the print process gets informed of the decrease of capacity.

Next, also a staff update can occur. The waiting room can receive information via channel *sa* from the triage process, the treatment room process and from proc *U*. The update is processed and the new current available capacity is sent to the print process.

If a treatment room becomes available, also variables *ra* and *wip* are updated. The last alternative is to receive a void signal to terminate this process at the end of the simulated day.

```
proc W(chan patient a; list(2) chan patient b; chan stlvl p;
       chan strm sa; chan void e; roster ros; int day; siminits si):
    list(3) list(4) int dc = [[ 1, 0, 0, 0], [ 0, 1, 1, 1], [-1,-1,-1,-1]];
    list(12) list patient xss;
    list(12) int cnts;
    list(2) list(2) int k;
    patient x;
    staff wip, ast = ros.st[0];
    bool go = true;
    int idx, rdx, ra = si.NR, rr = si.NRR;
    strm strmx;

    start U(sa,ros);
    p!(day, time, ast);

    while go:

        k[0] = dispatch_triage(xss[ :4], cnts[ :4], ast, dc[0]);
        k[1] = dispatch_treatment(xss, cnts,   ast, dc[1], ra, rr);
        if k[0] == k[1]:
            k[1][0] = 12
        end
        if k[1][0] < 12:
            xss[k[1][0]][k[1][1]].wip = wip
        end

        select
            a?x:
                idx = x.color;
                if x.amb == 1:
                    idx = idx + 8;
                elif x.t.triage.b != 0.0:
                    idx = idx + 4;
                end
                xss[idx]  = xss[idx] + [x];
                cnts[idx] = cnts[idx] + 1;
        alt
            unwind i in range(2):
                k[i][0] < 12, b[i]!xss[k[i][0]][k[i][1]]:
                    idx       = k[i][0];
                    rdx       = k[i][1];
                    x         = xss[idx][rdx];
                    ast       = st_update(ast, x.st, dc[i]);
                    xss[idx]  = xss[idx] - [xss[idx][rdx]];
                    cnts[idx] = cnts[idx] - 1;
                    if i == 1:
                        ra = ra - 1;
                        wip = modWIP(wip, x.st, 1)
                    end
                    p!(day, time, ast)
            end
        alt
            sa?strmx:
                ast = st_update(ast, strmx.st, dc[2]);
                p!(day, time, ast);
                if strmx.rm:
                    ra  = ra + 1;
                    wip = modWIP(wip, strmx.st, -1)
                end
        alt
            e?:
                go = false
        end
    end
end
```

Listing 3.8: Chi 3 listings of the waiting room process.

### 3.3.8 Triage process

As mentioned in Section 3.3.3, process $DT$ is used to sample the triage time. Listing 3.9 shows the Chi 3 listing of this process. The process input variables are the two distribution parameters and the send channel *dt*. The distribution is initiated and a sample is sent.

```
proc DT(chan real dt; real a, b):
   dist real dtreat  = gamma(a, b);
   dt!(sample dtreat)
end
```

Listing 3.9: Chi 3 listings of the process to sample triage or treatment times.

The triage process $T$ is shown in Listing 3.10. A patient is received via channel *a*. The triage start time is added to the patient details and process $DT$ is started to gain the triage time. The process is delayed with this time and the end time is added to the patient details. The triage is finished, the released capacity is calculated and send via channel *sa* to the waiting room. Next, also the patient is sent back to the waiting room. Likewise as in the waiting room process, the triage process can also receive the end of the day signal from $P$.

```
proc T(chan patient a, b; chan strm sa; chan void e; chan real dt):
   staff st, st_empty;
   real ttriage;
   patient x;
   bool go = true;

   while go:
      select a?x:
         x.t.triage.b = time;
         start dT(dt, 3.0, 3.5);
         dt?ttriage;
         delay ttriage;
         x.t.triage.e = time;
         st = st_update(st_empty, x.st, [-1,0,0,0]);
         sa!(false,st);
         b!x
      alt
         e?:
            go = false
      end
   end
end
```

Listing 3.10: Chi 3 listings of the triage process.

### 3.3.9 Treatment room process

The listing of the treatment room process is given in Listing 3.11. When a patient enters, the room number and the time of the start of the treatment are logged. The type of physician and the number of patients currently under treatment at that physician are saved in respectively *doctype* and *wip*. Next, the treatment time parameters can be determined. These parameters are used by processes $dT$ to determine the treatment time. Function *adapttmt* is used to adapt the treatment time if different scenarios are conducted. For a normal simulation, this function returns the same value.

Next, three timers are set. Timer *tt* denotes the end of the treatment, if *tr* finishes, the intensive treatment for red patients ends and is set to normal occupation. Timer *t2* is used for patients that need a second consult. When this timer is finished, the physician is switched. If the latter two timers are not applicable for a patient, the used capacity is not changed by the for loop from line 680 to 697.

If the treatment time is passed, the end time is logged to the patient data and the patient is sent to the exit process *E*. Also, the waiting room process is informed of the nurse, physician and treatment room capacity that becomes available. The treatment room process is now ready to receive a new patient.

Beside process *D*, *W* and *T*, the treatment room processes are also terminated after receiving a void signal from the print process *P*.

```
642 proc R(chan patient a, b; chan strm sa; chan void e; chan real dt;
643        t_treat tmt; siminits si; int i):
644     dist real dhigh    = triangle(15.0, 20.0,  30.0);
645     dist real dtmt     = uniform(0.0, 1.0);
646
647     staff st_empty, st_back, st_pat, st_temp;
648     real  ttreat, ta, tb;
649     int   doctype, wip, n, k;
650     bool  t2b, trb, go = true;
651     patient x;
652     timer tt,tr,t2;
653
654     while go:
655        select a?x:
656           x.room       = i;
657           x.t.treat.b = time;
658           st_pat = x.st;
659
660           if x.st.s > 0:
661              doctype            = 0;
662              wip                = min(7,x.wip.s)
663           else:
664              doctype            = 1;
665              wip                = min(7,x.wip.p[x.spec])
666           end
667           ta      = tmt.a[x.spec][x.color][x.age][wip][doctype][x.cons2];
668           tb      = tmt.b[x.spec][x.color][x.age][wip][doctype][x.cons2];
669           start dT(dt, ta, tb);
670           dt?ttreat;
671           ttreat = adapttmt(x, si, ttreat, sample dtmt, doctype);
672
673           tt = timer(ttreat);
674           t2 = timer(x.cons2 * real(t2) * 0.5);  t2b = true;
675           tr = timer( min( sample dhigh, real(t2) ) ); trb = true;
676
677           delay min(real(t2),real(tr));
678           for j in range(2):
679              select ready(t2) and t2b:
680                 k = max( max(x.st2.p), x.st2.s );
681                 n = max( max(st_pat.p), st_pat.s );
682                 st_temp = st_update(st_empty, st_pat, [0, 0, -k, -k]);
683                 st_back = st_update(st_temp, x.st2, [0, 0, n, n]);
684                 sa!(false, st_back);
685                 st_pat  = st_update(st_pat, st_back, [0, 0, 1, 1]);
686                 t2b     = false
687              alt ready(tr) and trb:
688                 st_temp    = modWIP(st_empty, st_pat, -1);
689                 st_back    = st_update(st_temp, st_pat, [0, -1, -1, -1]);
690                 sa!(false, st_back);
691                 st_pat = modWIP(st_empty, st_pat, 1);
692                 trb     = false
693              end;
694              delay max(real(t2),real(tr))
```

```
            end

            delay real(tt);
            x.t.treat.e = time;
            sa!(true, st_pat);
            b!x;
        alt
            e?:
                go = false
        end
    end
end
```

Listing 3.11: Chi 3 listings of the treatment room process.

### 3.3.10 Exit process

The listings of the exit process $E$ are shown in Listing 3.12. This process can receive a patient from the treatment room, adds the end time and sends the patient to the print process $P$. Again, also this process can receive the end of the day signal from $P$.

```
proc E(chan? patient a; chan! patient p; chan void e):
    patient x;
    bool go = true;

    while go:
        select a?x:
            x.t.total.e = time;
            p!x;
        alt
            e?:
                go = false
        end
    end
end
```

Listing 3.12: Chi 3 listings of the ED simulation model.

### 3.3.11 Print process

The listings of the print process can be found in Appendix B in line 723 to 797. This process receives patient and staffing information from process G, W and E. This simulation data is written to the output files ‘resultpatient.txt’ and ‘resultstaff.txt’. The first file contains the attributes and time stamps for each patient. Every five simulation minutes, the available staffing levels are saved in the second file.

Next, the print process keeps also track of the total number of patients in the system. If a day is finished and the system is empty, the print process uses channel $e$ to terminate processes W, T and R. After this is done, a new simulation day can start. If all simulation days are finished, the output files are closed and the total simulation terminates.

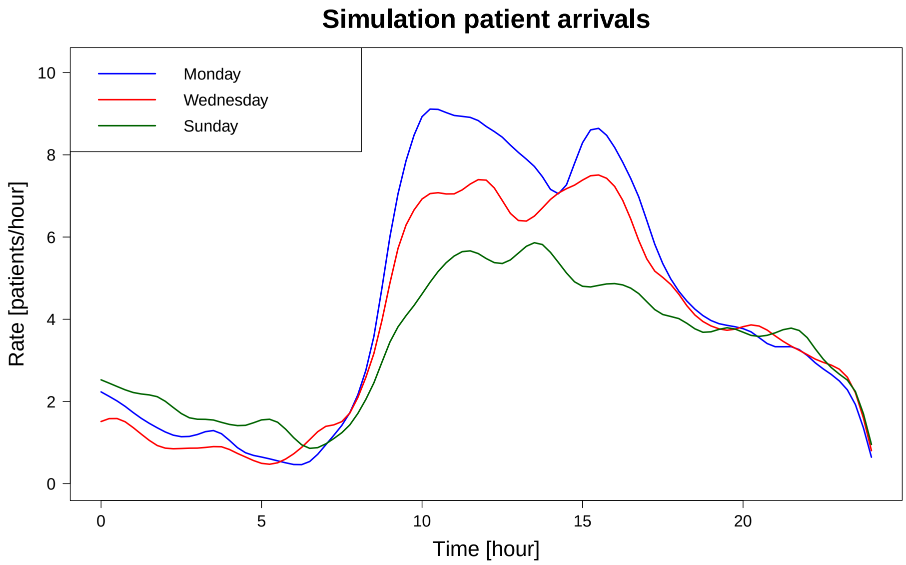


Figure 3.6: Patient arrival rates on an average Monday, Wednesday and Sunday from simulation data.

## 3.4 Model verification

This section describes the model verification. Model verification is used to check whether the simulation model is developed correctly. The first step in model verification is to debug the simulation model. This is done in multiple iterations during the development of the model. The current version runs without coding warnings or errors.

The next step of the model verification is to check whether the model generates the correct number of patients. Figure 2.4 shows the 2011 historical arrival rate. The same plot is created from simulation data, see Figure 3.6. This plot shows the average arrival rate of 100 simulated days for a Monday, Wednesday and Sunday. The historical average is based on 52 days. The two plots are similar which means that the arrival rate is modelled correctly. Note that the results of each simulation are not exactly the same due to stochastic elements.

Likewise as in Table 2.4, the average waiting time for red patients is smaller than the average waiting time for yellow patients. This also holds for yellow and green patients and for patients arrived by ambulance compared to patients arrived by own transportation.

The historical treatment time is used as a simulation model input, as mentioned in Section 2.2.5. Therefore, the treatment time for simulated patients should be equal to the treatment time of historical patients. The average treatment time for simulated patients is 184 and 107 minutes, respectively for patients with and without a second consult. In Appendix A, the average treatment times are compared for all end nodes of the decision tree. These treatment times match and therefore the conclusion can be made that the treatment times are modelled correctly.

Table 3.1: Tested ED-physician schedule.

| Time [h] | ED |
|---|---|
| 00:00 | 14 |
| 07:30 | 10 |
| 08:50 | 18 |
| 22:15 | 14 |

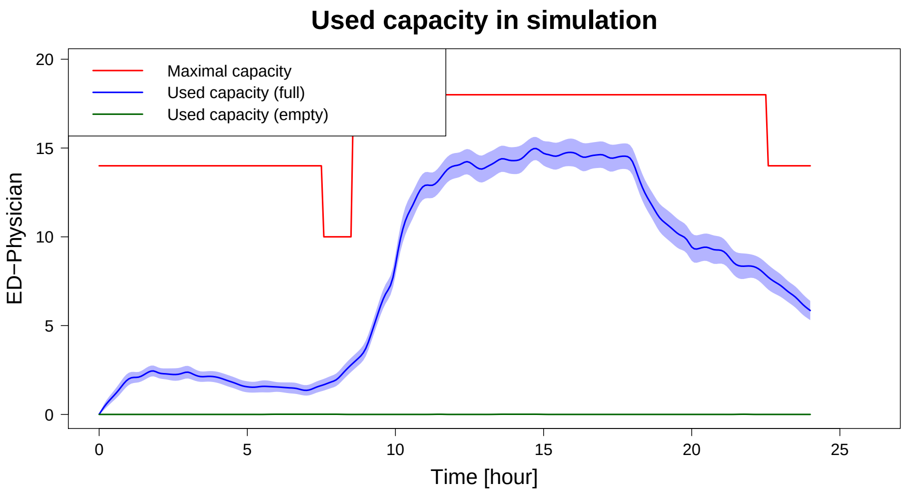


Figure 3.7: Simulated usage of ED-physician capacity.

The last part of this validation is to check if the working schedules are implemented correctly. Two simulation runs are performed, one with and one without patients. In both cases, the run consisted out of 100 simulation days and the roster, shown in Table 3.1, was used the ED-physicians. The average capacity use can be seen in Figure 3.7. The red line shows the maximal capacity. The green line shows that if no patients arrive, all ED-physician capacity is available. In case patients do arrive, also expected results are shown by the blue line. The colored band represents the 95% confidence interval.

From the analysis above can be concluded that the simulation model works correctly. The patient arrivals and treatment times are included in a proper way and the simulation runs without errors.

## 3.5 Model validation

Besides the model verification, also a model validation is conducted. The model validation checks if the simulation model represents the reality relatively close. This is done by comparing the output with the historical data and by team discussion.

The simulation structure and the results are discussed in several team discussion. Hospi-

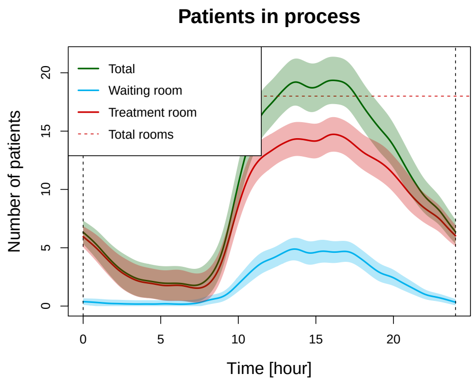


Figure 3.8: Historical occupation of patients on a Monday

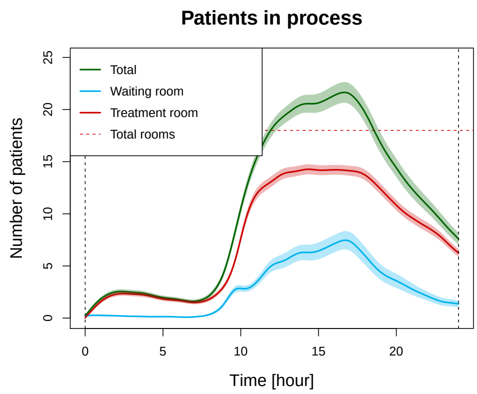


Figure 3.9: Patient occupation for a simulated Monday.

tal managers, the head of the ED, ED-physicians and senior nurses are included in these discussions. During these meetings, the tool for analysis of historical and simulation data was used. This tool is developed to increase the ease of analysing the data. For more information, see the software package manual [18].

As a result of these discussions, most assumptions were confirmed but the meetings have also lead to new insights. For example, a separate stream is created for patients arrived by ambulance. Also, these patients get a higher priority while waiting. Another remark during the discussions was that not all treatment rooms are used equally. Some rooms are more suitable for gynaecology or otolaryngology patients and other rooms are more often used for small traumas. However, this is not taken into account in the simulation model. See Chapter 6 for more recommendations for future work.

In the next part, the historical data and simulation output are compared. As mentioned in Subsection 2.2.2, Monday and Friday are the most busy days. Therefore, the results of the simulated Mondays are used in this section to show the match between historical data and the simulation. This section ends by showing some comparisons for a Sunday. All colored bands shown in the figures are the 95% confidence intervals.

In Figure 3.8 and Figure 3.9, the historical and simulated occupations are given for patients that are present in the waiting room, present in the treatment rooms and for the total number of patients at the ED. The results for the first hours of the day are different because the simulated ED begins each day empty.

Next, starting from 9 o'clock, the waiting room fills to approximately five patients in the afternoon. Both figures show that the waiting room is empty at the end of the day and that there are on average around 7 patients present in the treatment rooms.

Next, in Figure 3.10 and Figure 3.11, the usage of the ED-physician is shown for a historical and simulated Monday. In Figure 3.11, also the ED-physician roster is shown. Again, the first part differs but the other part of the day gives a good match. Deviations can be caused by red patients. As mentioned in Section 3.2.2, red patients claim more tokens for the first

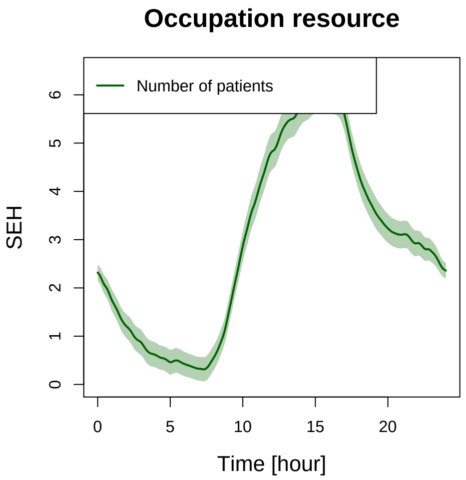


Figure 3.10: Historical use of the ED-physician on a Monday in 2011.

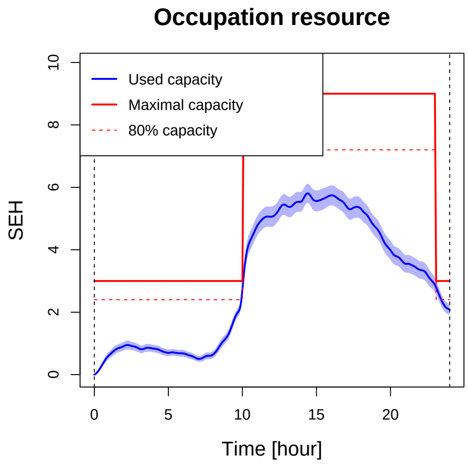


Figure 3.11: Simulated use of the ED-physician on a Monday.

part of the treatment.

Next, the waiting times are discussed. As can be seen in Figure 3.12 and Figure 3.13, the distributions of the waiting times for yellow patients give a relatively close match. Similar results can be obtained by comparing other patient categories.

In Figure 3.14, the cycle time factor is plotted. This factor is the total time a patient is present at the ED divided by the treatment time. If the factor is close to one, the patients have to wait relatively short compared to their treatment time.

Next, in Figure 3.15 and Figure 3.16 the match for the historical and simulated arrivals and departures on a Sunday is shown, the confidence intervals overlap.

In Figure 3.17, the cycle time factor is shown for a Sunday. Again, the simulation output matches the historical data. As expected, this cycle time factor is smaller than for patients arriving on Monday.

This chapter ends by concluding that the simulation model represents the reality relatively close for the different types of patients and for the different days of the week. In Chapter 4 and Chapter 5, the simulation model is used to analyse the 2011 situation and to trial scenarios.

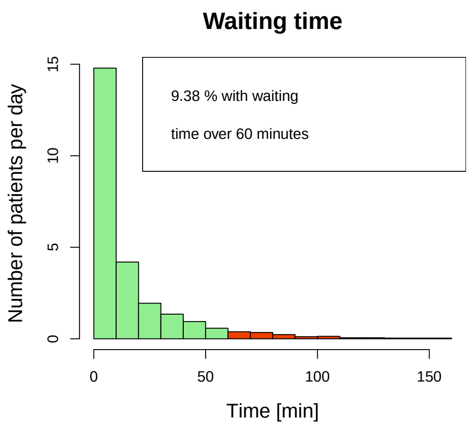


Figure 3.12: Historical waiting time for yellow patients arrived on Mondays.

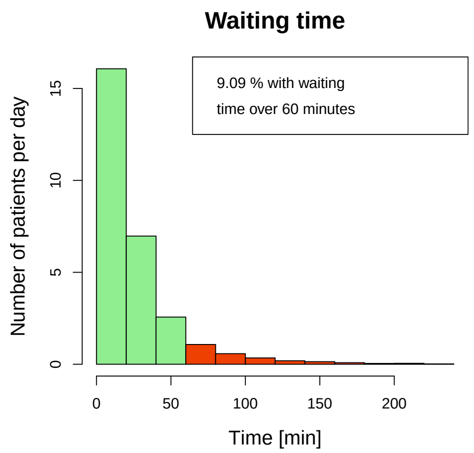


Figure 3.13: The simulated waiting time for yellow patients.

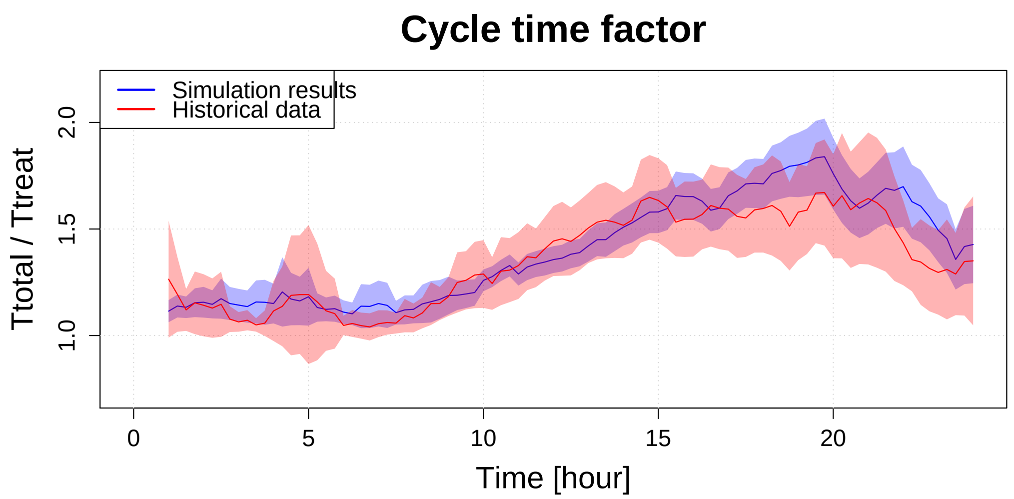


Figure 3.14: Cycle time factor for historical and simulated patients on Monday. The colored band represents the 95% confidence interval. The patients are located according to the time their treatment starts.

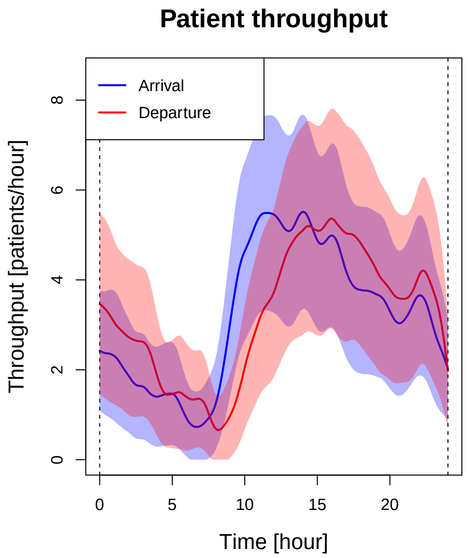


Figure 3.15: Historical arrival and departure rates for a Sunday in 2011.

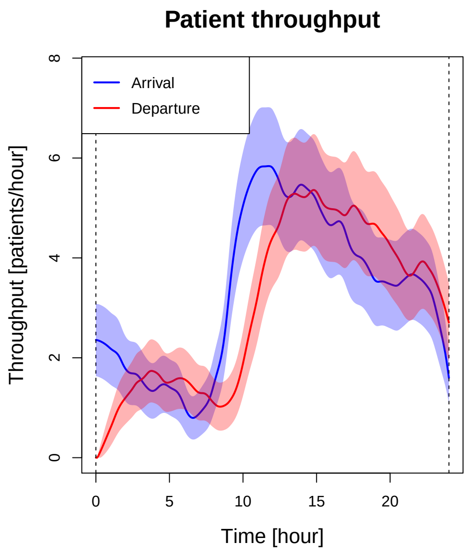


Figure 3.16: The arrival and departure rates derived from output of a simulated Monday.

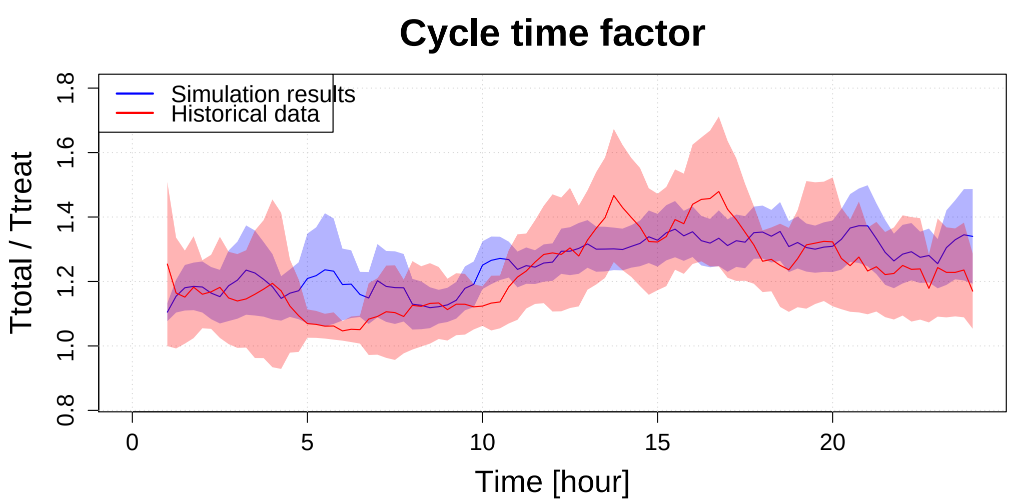


Figure 3.17: Cycle time factor for historical and simulated patients on Sunday.

# Chapter 4

# Performance improvement

Chapter 3 describes the model assumptions and explains how the simulation model is developed. In the first section of this chapter, the ED simulation model is used to analyse the 2011 situation. In the following sections, the simulation model is used to investigate a number of improvement opportunities. In the last section, several of these opportunities are combined to benefit from the mutual contributions.

## 4.1 Analysis of situation in 2011

As described in Section 3.1, a tool for analysis of simulation output is developed in R. The output is processed and adaptive plots are created using the R package *playwith*, see Figure 4.1. The plots show the arrival and departure rate of patients, the number of patients present at the department, the occupation of resources, and histograms of the waiting and total time. Due to the adaptability, a subset of patients can be selected. The patients can be filtered by triage color, medical speciality and by time interval. More information on this tool can be found in the manual [18].

The tool is used to conduct the analysis in this chapter. Using this tool, a large number of improvement goals can be evaluated. For example goals to reduce waiting times, reduce the number of patients waiting or to improve the utilisation of treatment rooms or nursing capacity. For a clear and structured approach, one goal is chosen for improvement in this chapter. In line with the motivation of this project, see Section 1.3, the aim of this analysis is to reduce the percentage of yellow and green patients, present on a Monday between 10:00h and 20:00h, that exceed maximal target waiting times, see Table 2.1. This time window is chosen because it is one of the most crowded times at the ED.

Before improvement opportunities are investigated, first the original situation is simulated. A simulation is conducted for 500 simulated Mondays. The results of the analysis are shown in Figure 4.2. One can see that over 18% of the yellow patients exceed the maximal target waiting time. The vertical black dotted lines in the plot on the right mark the time interval

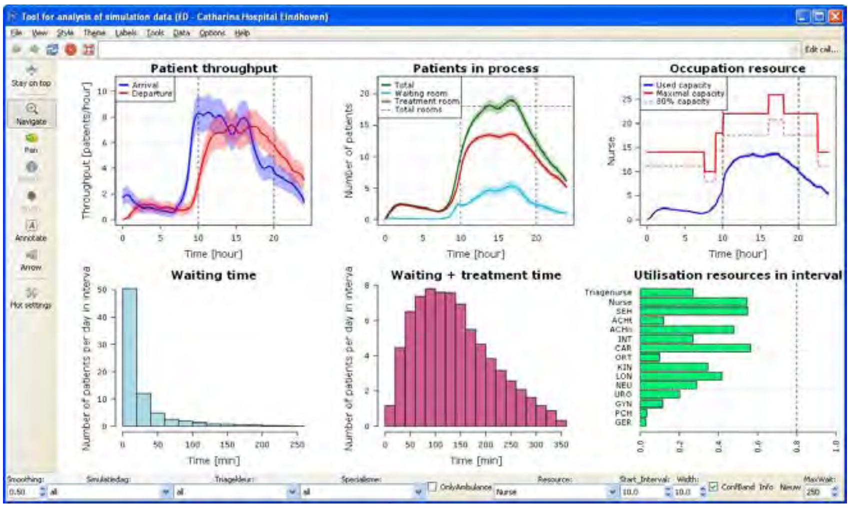


Figure 4.1: Print screen of tool for analysis of simulation output. The output is shown for a simulated Monday and the selected time interval is 10:00h to 20:00h.

of 10:00h to 20:00h. In the following sections, five different improvement opportunities are trialled; an increased number of treatment rooms, no priority for ambulance patients, increased nursing capacity, increased physician capacity and treatment time reduction. This chapter ends by evaluating a combination of these opportunities.

## 4.2 Increase treatment rooms

In this section, the influence of an extra treatment room is investigated. The results are shown in Figure 4.3. The increase of one treatment room leads to an decrease of 0.5% and 20.3% for respectively yellow and green patients that exceed maximal target waiting time. Corresponding with this result, one can see that the waiting room gets less filled.

However, a decrease of one treatment room results in 8.47% of the yellow and 12.49% of the green patients that exceed the target maximal waiting time.

## 4.3 No priority for ambulance arrivals

In the current situation, ambulance arrivals are given priority. The paramedics have to continue with their next request and therefore they would like to have minimal delays at the ED. What if an extra waiting room is created for patients arrived by ambulance? Using this extra waiting room, the senior nurse can follow the same procedure for ambulance arrivals as for patients arriving by own transportation.

The results are shown in Figure 4.4. A small increase in waiting times can be seen for yellow

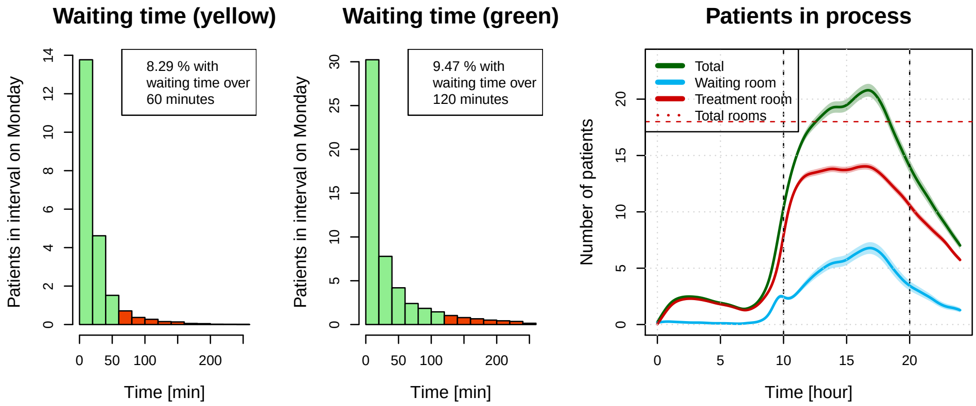


Figure 4.2: Unadapted results of a simulated Monday.

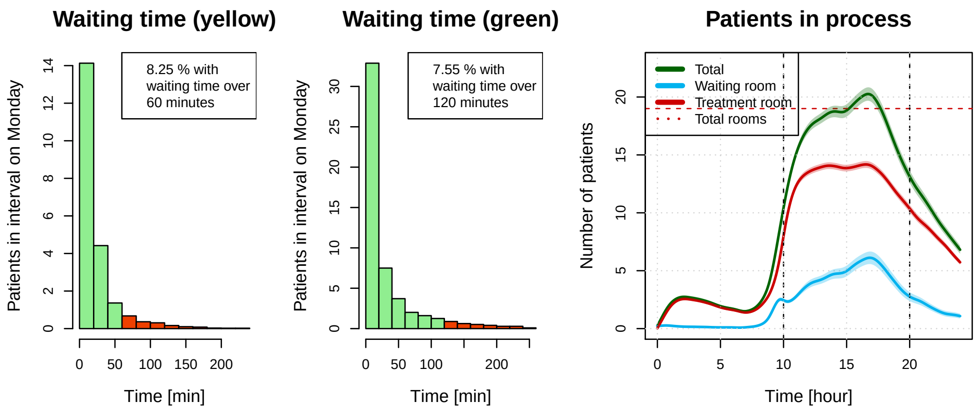


Figure 4.3: Simulation output using 19 treatment rooms.

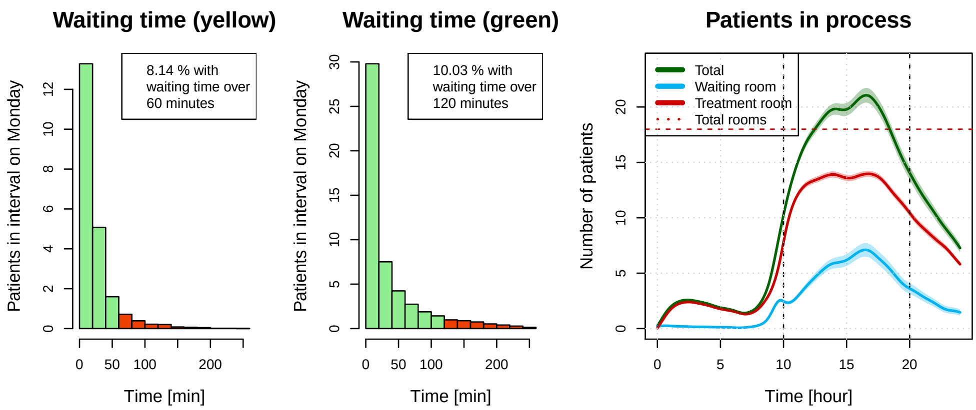


Figure 4.4: Simulation output without distinguishing patients by type of arrival.

and green patients. This result can be explained by the fact the former ambulance patients are not directly transferred to a treatment room but have to wait for available capacity. The effect is not very large due to the fact that only 12% of the total patients arrived by ambulance.

## 4.4 Increase nursing capacity

Next, the influence of nursing capacity is examined. A simulation is conducted in which the nursing capacity is increased by four tokens from 10:00 h until 20:00 h. As mentioned in Section 3.2, four tokens represent one nurse. Figure 4.5 shows the results of the simulation including the roster and used nursing capacity on the right-hand side. Despite the increase of nursing capacity, the number of patients that exceed the target time remains the same.

## 4.5 Increase physician capacity

The following improvement opportunity is the increase of physician capacity. First, the capacity of the ED-physician is increased by three tokens. These tokens represent one extra physician. Figure 4.6 shows, similar to the results of increasing nurse capacity, that an increase of ED-physician capacity does not result in a decrease of patients that exceed the target time.

Next, instead of increasing the ED-physician capacity, the maximum capacity of all other physician types is raised with one token between 10:00h and 20:00h. The results are shown in Figure 4.7. In addition, the occupation of the cardiologist is shown to illustrate the increase of capacity. The percentage of patients that exceed the target time substantially drops with 63.1% and 66.8% for respectively yellow and green patients.

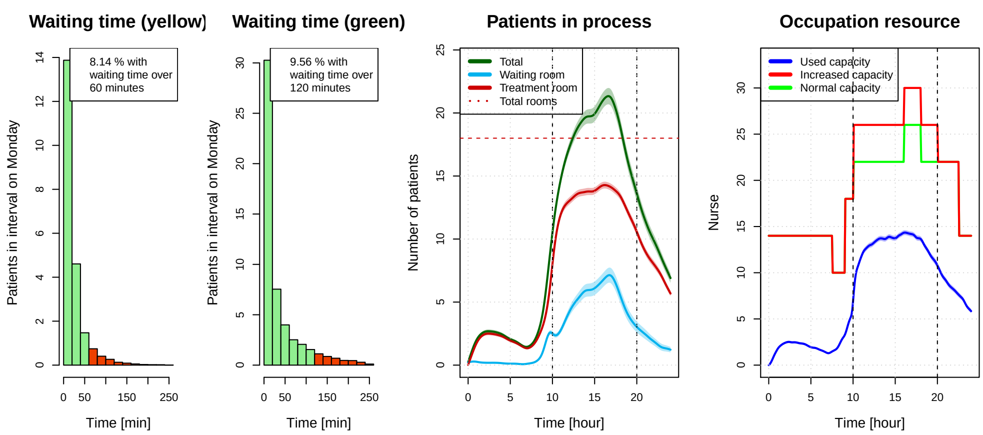


Figure 4.5: Simulation output with extra nursing capacity.

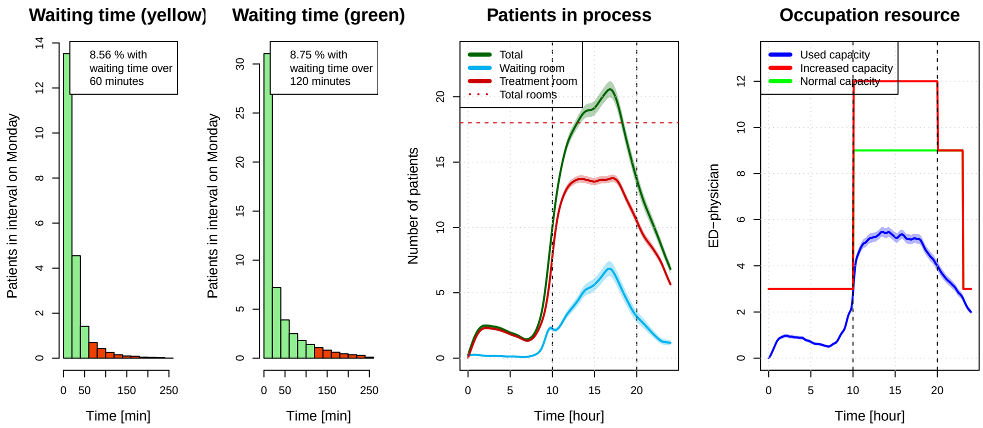


Figure 4.6: Simulation output with extra ED-physician capacity.

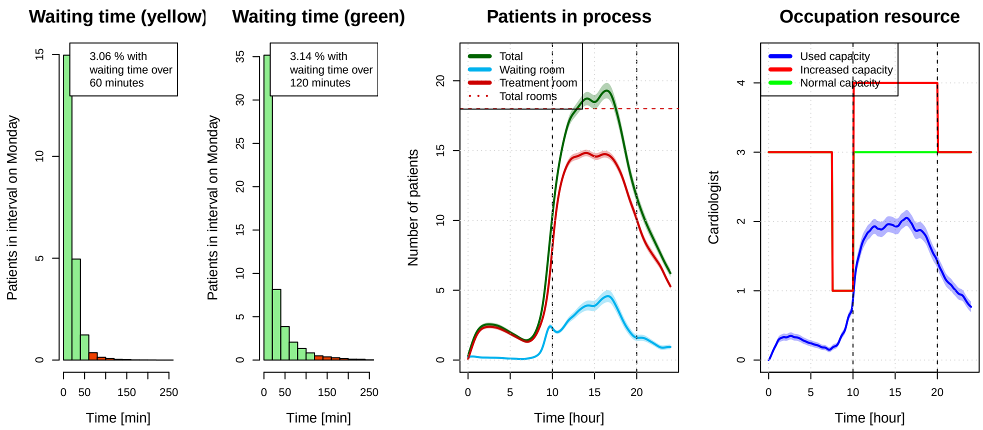


Figure 4.7: Simulation with one extra token per physician type, except for the ED-physician.

## 4.6 Treatment time reduction

As mentioned in Section 1.5.2, the LEAN concept is also introduced in the CZE to improve operational processes [21]. The aim is to eliminate unnecessary operations to achieve better performance using existing resources. This approach can for example result in a reduction of the treatment time by 10 minutes per patient.

One potential way to achieve this reduction is to shorten the time for hospitalization. This time starts from the moment that the treatment actually finishes until the patient is picked up by the nurse of the ward. About 30% of the patients, mostly elderly, are hospitalized.

A simulation is performed in which the treatment time per patient is reduced with 10 minutes. The results are shown in Figure 4.8. The number of patients waiting as well as the number of occupied treatment room is decreased. The percentage of patients that exceed

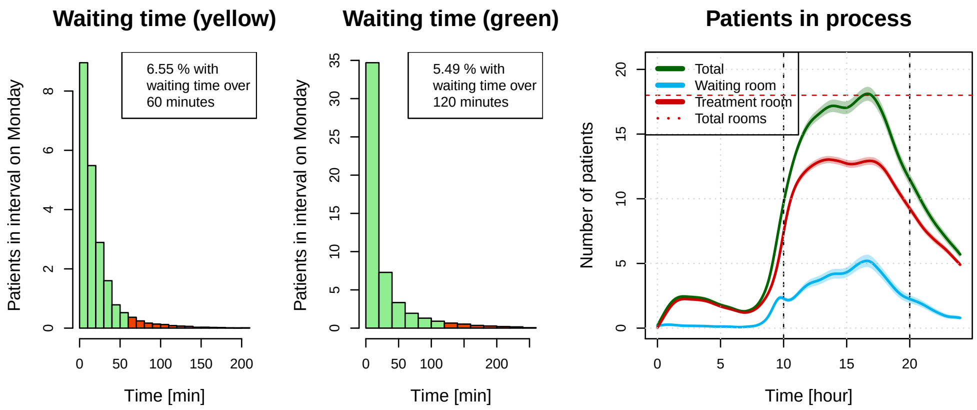


Figure 4.8: Simulation output for 10 minutes treatment time reduction per patient.

the target maximal waiting time is reduced with 21.0% and 42.0% for respectively yellow and green patients.

However, if the treatment time is increased by 10 minutes per patient, an opposite result can be observed. The increase results in 9.12% and 14.76% of the yellow and green patients exceeding the target.

## 4.7 Combination of improvements

In the previous sections, several improvement opportunities are elaborated. Subsequently, the most influential opportunities are (partially) combined and the results are discussed in this section. The simulation that is performed includes a 10 minutes decrease in treatment time for 50% of the patients, one extra treatment room and one extra token for the cardiologist, medical intern, neurologist and lung specialist from 10.00h until 20.00h.

The simulation results are shown in Figure 4.9 and Figure 4.10. With the introduced adaption, the percentage of patients that exceed the target waiting time is decreased by 47.0% for yellow patients and by 68.3% for green patients.

The improvement in cycle time factor is shown in Figure 4.10 between the unadapted and adapted simulation output. One can see that the peak between 16:00h and 22:00h is declined. This means that the waiting time has decreased relatively more than the introduced decrease in treatment time. Note that the results in this chapter are derived from a simulated Monday.

In this chapter, the simulation model is used to trial several improvement opportunities. Treatment time reduction and increased availability of physicians from other departments tend to be powerful improvement opportunities. It can be concluded that a combination of these opportunities also leads to a large improvement.

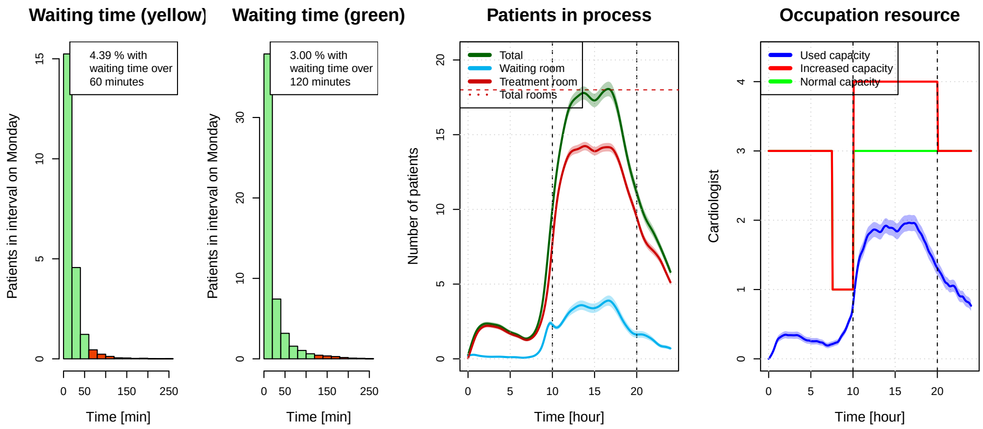


Figure 4.9: Simulation output for treatment time reduction, physician capacity increase and an extra treatment room.

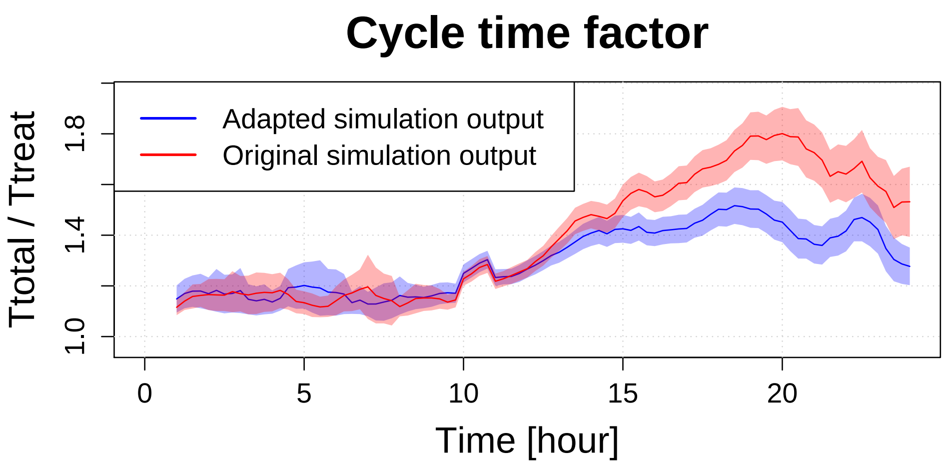


Figure 4.10: Cycle time factor comparison between the unadapted simulation output and the output for a simulation with treatment time reduction, physician capacity increase and an extra treatment room.

# Chapter 5

# Scenario analysis

In Chapter 4, the simulation model is used to improve the 2011 situation. This chapter explores other possibilities for the use of the simulation model. As described in Section 3.1, the software package contains an Excel file that is used to create the input files for the simulation model. This gives the user the ability to trial different scenarios. In this chapter, the following three scenarios are elaborated:

- In general holds that older patients have longer treatment times. What effect has an increase of ED visits by elderly patients, due to the aging population?
- What extra capacity is needed if a neighboring ED closes and the CZE ED has to partially take care of their patients?
- What if the average urgency of patients increases? For example due to less self-referrals.

Next to these scenarios, it is also possible to use the simulation model to trial cases such as:

- What if more accurate triage results in less second consults and thus a decrease of treatment times?
- What capacity of ED-physicians is needed if more patients are consulted by the ED-physician instead of the specialist of the attending medical speciality.

## 5.1 Scenario 1

The first scenario that is discussed focusses on the aging population. At the CZE ED, the number of patients that visit the geriatric is already increasing. To illustrate, more than 220 patients have visited the geriatric in 2011. In 2012, this number was already reached at

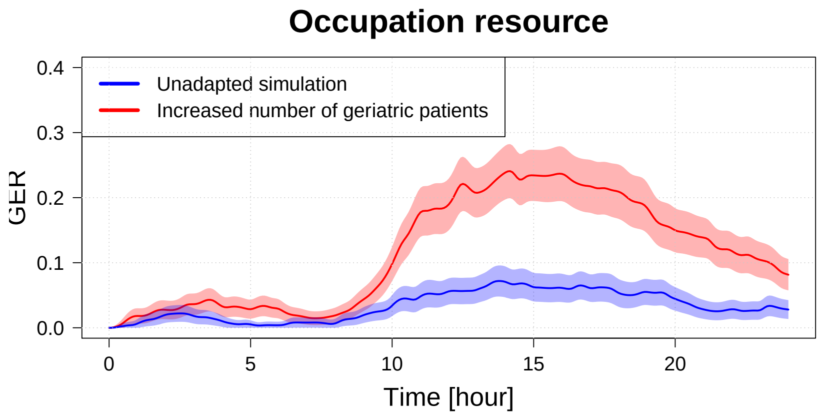


Figure 5.1: Capacity usage of the geriatrist for a simulation with normal arrivals and with increased arrivals of geriatric patients.

the end of August, which roughly means an increase of 50%. As a consequence of the aging population, also the number of geriatric patients will increase. As shown in Table 2.2.5, the treatment times for geriatric patients are on average 30 minutes longer than for other patients.

In this scenario, the number of geriatric patients has been increased by 300% in the simulation. The results show that more capacity is demanded from the geriatrist, see Figure 5.1. It can be concluded that the shifted distribution of medical speciality to more geriatric patients has a negligible effect on the performance of the entire ED.

## 5.2 Scenario 2

In the second scenario, the situation is examined what happens if a neighboring ED, see Figure 5.2, has to (temporarily) close. This closure can be caused by a MRSA-outbreak or by financial cutbacks. In this case, the introduced closure results in an increase of 15% in patient arrivals.

The growth of patient arrivals by 15% results in a large increase of waiting times. On average, there will be more than 12 patient waiting during peak hours. 22.42% of green patients, present between 10:00h and 20:00h, exceed the target waiting time. For yellow patients, the percentage is 10.97%.

## 5.3 Scenario 3

As mentioned in Section 1.2, the general partitioner is first care and the ED is officially second care. Therefore, patients are not supposed to visit the ED by self-referral. To reduce

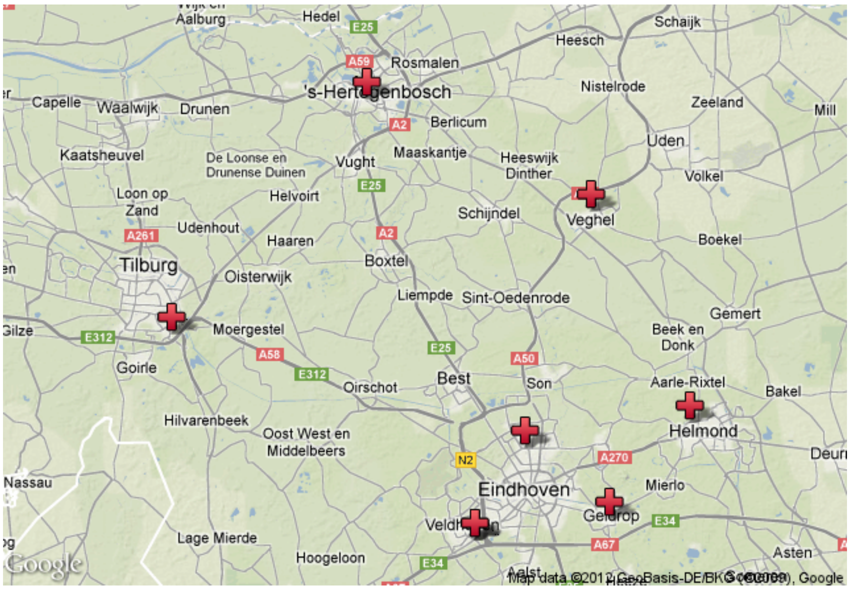


Figure 5.2: Location of emergency departments in the Eindhoven area.

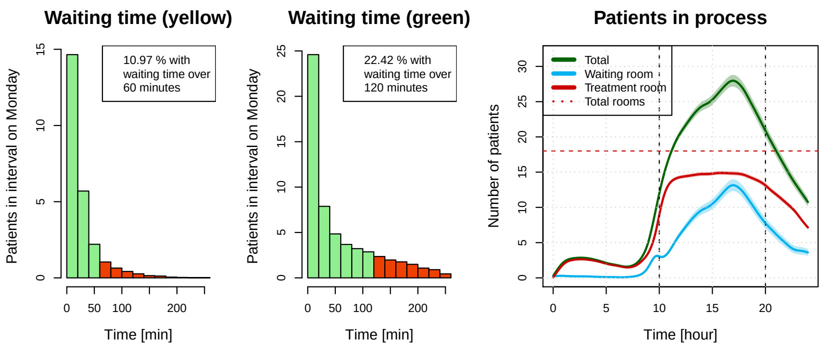


Figure 5.3: Simulation output for the CZE ED on Mondays with a 15% growth of patient arrivals.

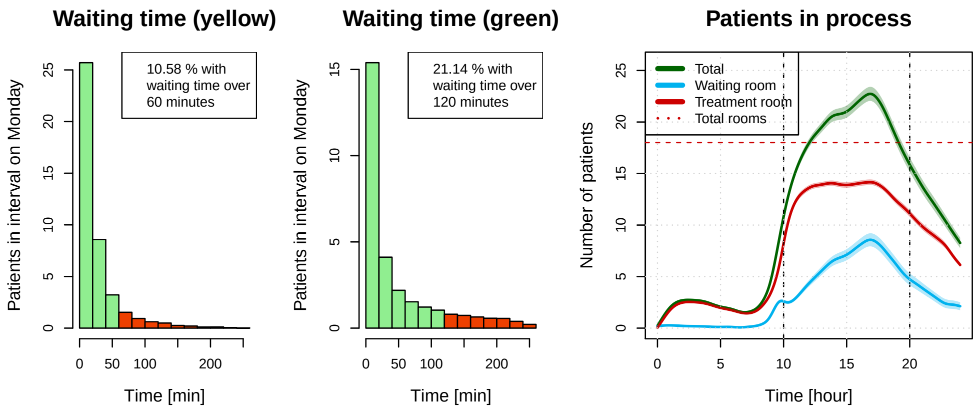


Figure 5.4: Simulation output for a shifted triage distribution. Relatively more urgent patients arrive.

this group, several action are taken. The GP and ED try to provide patients with better information and the national government has plans to ask an extra charge from self-referrals.

In this scenario, the actions to reduce self-referrals is paying off for patients that have less urgent complaints. They are more likely to understand that their GP is also able to consult them for their health complaints. More urgent patients are more impatient and want directly ED care. Thus, a decrease of self-referrals can result in a shifted mix of patients. The patients visiting the ED are more urgent and to conduct this in the simulation, the percentage of red and yellow patients is increased and the percentage of green and blue patients is reduced.

In line with the expectations, the percentage of the reduced group of green patients that exceeds the target time is increased. They have to wait longer before yellow patients, who get priority, are treated.

# Chapter 6

# Conclusions and recommendations

The objective of this research has been to give insight in the current efficiency of the ED, to identify the current bottlenecks and to supply answers to scenarios. First, the processes of the ED have been studied and a data analysis has been conducted. This resulted in better understanding of the ED which assisted in developing the simulation model. The model represents the reality relatively close. That enabled the possibility to use the model to search for effective improvements and for trailing scenarios. The simulation model is part of a software package. This package also contains tools for analysis of historical data and simulation output. These tools proved to be very useful to give insight in the data by generating clear graphs.

Next, the simulation model is used to investigate improvement opportunities. In this report, a case study is conduced to find effective improvements to decrease the number of patients on Monday that have to wait longer than the target waiting time. Increase of physician capacity from other departments and a decrease of the treatment time tend to be the most powerful improvements. A combination of these factors can result in a 68% decrease of green patients that exceed the waiting time target.

Moreover, it is also possible to use the model to investigate potential future scenarios. Among others, the expected impact of an increase of arriving patients is explored. The number of patients in the waiting room increases and the percentage of green patients that exceed the target waiting time is doubled.

It can be concluded that the simulation model represents the reality relatively close. Moreover, the tools can be used to search for bottlenecks and the simulation model is capable of trailing opportunities for improvement as well as several scenarios. Also, the software package is useful and easy to handle for generating input files, performing a simulation of the ED, and for analysis of historical data and simulation output.

The cycle time factor plots are used to visualize the efficiency of the ED at a glance.

However, no aggregate model is build next to the detailed simulation model. An aggregate model would be a useful addition because it can be used to compare different days with each other or to compare the efficiency of this ED with the ED of another hospital.

All in all, a step in the right direction is made with respect to the accuracy of the simulation model. However, there is room for improvements. More accurate and more complete data is required to integrate more detailed processes. Also, the time that a historical event gets logged is not always the same as the time that the event actually happens.

The simulation model can be improved by extending the treatment process. In the current simulation model, the treatment is modeled on a high abstraction level. The second consult and increased capacity for red patients are included but other influences such as the waiting time for x-rays or blood test results is not included.

The waiting room process uses a strict dispatching policy. However, reality shows the rules are not always strictly followed. According to the policy, all yellow patients are served prior to the green patients. Though, often an exception is made. For example, a long waiting green patient is often dispatched first by the senior nurse if a yellow patient just arrives. Hereby the possibility is reduced that a green patient exceeds the maximal target waiting time.

Finally, an increase of integrated possible improvements would be very useful. One can think of different dispatching policies or the restrictions of treatment rooms for different types of patients. Also, to be able to trial improvement opportunities such as the introduction of an extra waiting room for patients that have to wait during their treatment for test results would be an enrichment to the package.

# Bibliography

[1] I.J.B.F. Adan, A.T. Hofkamp, and J.E. Rooda. Chi 3.0.0 documentation. http://chi.se.wtb.tue.nl/, 2012. [Online; last accessed 10-Oct-2012].

[2] C. Alexopoulos. Modeling patient arrivals in community clinics. *Omega: The International Journal of Management Science*, 36:33–43, 2008.

[3] R. Ceglowski, L. Churilov, and J. Wasserthiel. Combining data mining and discrete event simulation for a value-added view of a hospital emergency department. *The Journal of the Operational Research Society — A Special Issue: Operational Research in Health*, 58(2):246–254, 2007.

[4] J. Crane and C. Noon. *The definitive guide to Emergency Department Operational Improvement*, chapter 8 Lean Flow. CRC Press, 2011.

[5] Catharina Hospital Eindhoven. Samenvatting jaardocument 2011. http://www.catharinaziekenhuis.nl/files/Over_Catharina/Samenvatting_Jaardocument_2011_Catharina_Ziekenhuis.pdf, February 2012. [Online; last accessed 02-Oct-2012].

[6] J.J. Allaire et al. R studio ide. http://www.rstudio.com, 2012. [Online; last accessed 26-Oct-2012].

[7] J.W. Fowler, J.C. Benneyan, P. Carayon, B.T. Denton, P. Keskinocak, and G.C. Runger. An introduction to a new journal for healthcare systems engineering. *IIE Transactions on Healthcare Systems Engineering*, 1:1–5, 2011.

[8] R. Gentleman and R. Ihaka. R project. http://www.r-project.org, 2012. [Online; last accessed 25-Oct-2012].

[9] L.V. Green. Capacity planning and management in hospitals. In *Operations research and health care*, chapter 2, pages 15–42. Kluwer's International Series, 2004.

[10] J. Grone. Nederlandse zorg onder de maat. http://medischcontact.artsennet.nl/Nieuws-26/Nieuwsbericht/115033/Nederlandse-zorg-onder-de-maat.htm, May 2012. [Online; last accessed 02-Oct-2012].

[11] C.A. Grummels. Simulation and optimization of an emergency department. Master thesis SE 420676, Eindhoven University of Technology, Manufacturing Networks Group, Department of Mechanical Engineering, Eindhoven, The Netherlands, 2012.

[12] A. Jeninga. Hoe maak ik mijn spoedeisende hulp lean? http://www.cczorgadviseurs.nl/nl-nl/updates/11695/hoe-maak-ik-mijn-spoedeisende-hulp-lean.html, July 2012. [Online; last accessed 02-Oct-2012].

[13] Microsoft. Office excel 2007. http://office.microsoft.com/en-us/excel, 2012. [Online; last accessed 25-Oct-2012].

[14] ORACLE. Oracle java development kit. http://www.oracle.com/technetwork/java/javase/downloads/index.html, 2012. [Online; last accessed 25-Oct-2012].

[15] N. Rijk, M.M.M. Schepens, E.A.T.E. Termeczky, and S.M.J. Verhoeven. Doorlooptijd SEH — kan het sneller? Master's thesis, School of Industrial Engineering, Eindhoven University of Technology, 2010.

[16] S.H.E. Tacken. Aankomstproces spoedpatiënten in het Catharina Ziekenhuis te Eindhoven. Bachelor's thesis, School of Industrial Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands, July 2010.

[17] T. Therneau and B. Ripley B. Atkinson. Package 'rpart' — recursive partitioning. http://cran.r-project.org/web/packages/rpart/rpart.pdf, 2012. [Online; last accessed 02-Nov-2012].

[18] J.J.D. Timmermans. Documentation of the simulation model tools of the emergency department at catharina hospital eindhoven. MN Report MN-420703, Eindhoven University of Technology, Manufacturing Networks Group, Department of Mechanical Engineering, Eindhoven, The Netherlands, 2012.

[19] J.J.D. Timmermans. Patient transit simulation. SE Report 420679, Eindhoven University of Technology, Manufacturing Networks Group, Department of Mechanical Engineering, Eindhoven, The Netherlands, 2012.

[20] R.E. Visser. Modeling an integrated emergency post. Master's thesis, Industrial Engineering and Management Group, School of Management and Governance, University of Twente, 2011.

[21] M. Wolleswinkel. In de start blokken — tweegesprek over het meerjarenbeleidsplan. *Ons Catharien*, 2(2):6–7, April 2012.

[22] M.E. Zonderland. *Curing the Queue*. PhD thesis, Center for Telematics and Information Technology, Twente, The Netherlands, University of Twente, Enschede, the Netherlands, 2012.

# Appendix A

# Treatment time decision tree

In this Appendix, the treatment time decision tree is given, see Figure A.1. For more information, see Section 2.2.5. The mean treatment time resulting from the decision tree and from simulations is given for each end node in Table A.1. Approximately 10.000 simulated and 34.000 historical patients are used to calculate the mean treatment times.

Table A.1: Comparison between simulated and tree average treatment time per patient group.

| End node | Simulation mean | Tree mean |
|---|---|---|
| 14 | 179 | 171 |
| 15 | 204 | 201 |
| 22 | 134 | 116 |
| 23 | 178 | 186 |
| 24 | 92 | 98 |
| 34 | 144 | 135 |
| 35 | 251 | 217 |
| 38 | 131 | 107 |
| 39 | 158 | 189 |
| 40 | 104 | 104 |
| 42 | 131 | 123 |
| 43 | 140 | 146 |
| 50 | 119 | 121 |
| 51 | 135 | 143 |
| 52 | 119 | 126 |
| 54 | 155 | 153 |
| 55 | 170 | 169 |
| 65 | 97 | 97 |
| 66 | 83 | 83 |
| 72 | 52 | 55 |
| 73 | 80 | 83 |
| 82 | 113 | 111 |
| 83 | 144 | 136 |
| 106 | 117 | 113 |
| 107 | 148 | 146 |
| 128 | 51 | 52 |
| 134 | 88 | 92 |
| 135 | 118 | 118 |
| 148 | 47 | 63 |
| 149 | 82 | 85 |
| 150 | 92 | 95 |
| 258 | 63 | 66 |
| 259 | 78 | 75 |
| 303 | 131 | 128 |
| 604 | 106 | 104 |
| 1210 | 105 | 97 |
| 1211 | 119 | 123 |

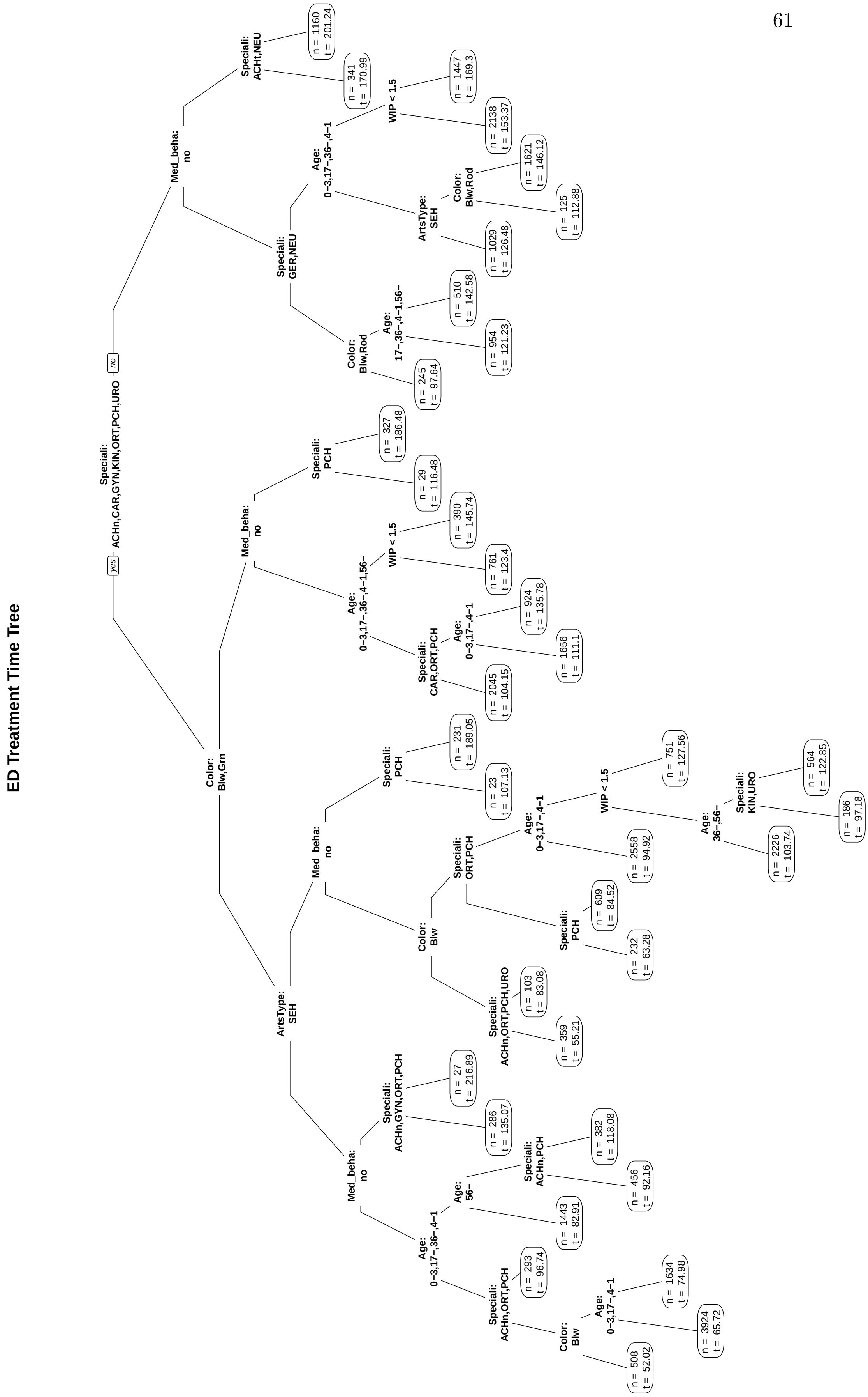


Figure A.1: Treatment time decision tree.

# Appendix B

# Chi 3 listings simulation model

In this Appendix, the Chi 3 listings of the simulation model of the CZE ED are given.

```
const real maxreal = 9e9,
      int  AR      = 24,   # different arrival rates
      int  DS      = 400,  # distribution intervals
      int  DC      = 12;   # different specialities

type patient    = tuple(int spec, color, age, amb, room, cons2;
                        staff st, st2, wip;
                        timelist t),
     staff      = tuple(int  t, n, s;
                        list(DC) int p),
     inter      = tuple(real b, e),
     timelist   = tuple(int  simday;
                        inter total, triage, treat),
     roster     = tuple(list real t;
                        list staff st),
     stlvl      = tuple(int day;
                        real ttime;
                        staff st),
     siminits   = tuple(int  weekday, simdays, NR, NRR, amb;
                        real ArrAll; list(DC) real ArrSpec; list(4) real ArrCol;
                        list(6) real ArrAge; list(2) real ArrPhy, Arr2con;
                        int TmtAll; real TmtPer; list(DC) int TmtSpec; list(4) int
                            TmtCol;
                        list(6) int TmtAge; list(2) int TmtPhy, Tmt2con),
     arrival    = tuple(list(2)  list(AR) real rate;
                        list(2)  list(DS) int spec;
                        list(2)  list(DC) list(DS) int col;
                        list(DC) list(DS) int age;
                        list(DC) real phys;
                        list(DC) list(4)  real cons2;
                        list(DC) list(DS) int cons2by),
     t_treat    = tuple(list(DC) list(4) list(6) list(9) list(2) list(2) real a;
                   list(DC) list(4) list(6) list(9) list(2) list(2) real b),
     strm       = tuple(bool rm;
                        staff st);

model M():
   siminits  si  = read_inits("simulation_init.txt");
   arrival   arr = read_arr_input(si);
   roster    ros = read_schedule_input(si.weekday);
   t_treat   tmt = read_treatment_input("treatment.txt");
   file    f_pat = open("resultpatient.txt", "w"),
            f_st = open("resultstaff.txt", "w");
```

```
   for i in range(si.simdays):
      run SimOneDay(arr, ros, f_pat, f_st, si, tmt, i)
   end

  close(f_pat);
  close(f_st)
end

proc SimOneDay(arrival arr; roster ros; file f_pat, f_st;
            siminits si; t_treat tmt; int i):
   chan patient a, c, d, pe;
   list(2) chan patient b;
   chan stlvl pst;
   chan void pa, e;
   chan strm sa;
   chan real dt;

   run
      unwind j in range(2):
         G(a, i, j, arr, pa, time, si)
      end,
      D(a,d, e),
      W(d, b, pst, sa, e, ros, i, si),
      T(b[0], a, sa, e, dt),
      unwind k in range(si.NR):
         R(b[1], c, sa, e, dt, tmt, si, k)
      end,
      E(c, pe, e),
      P(pa, pe, pst, e, f_pat, f_st, i, time, si.NR)
end

func siminits read_inits(string s):
   siminits si;
   string h;

   file f = open(s, "r");
   h = read(f, string);
   si.weekday = read(f, int) - 1;
   h = read(f, string);
   si.NR = read(f, int);
   h = read(f, string);
   si.simdays = read(f, int);
   h = read(f, string);
   si.amb = read(f, int);
   h = read(f, string);
   si.NRR = read(f, int);

   h = read(f, string);
   si.ArrAll = read(f, real);
   h = read(f, string);
   for i in range(DC):
      si.ArrSpec[i] = read(f, real)
   end
   h = read(f, string);
   for i in range(4):
      si.ArrCol[i] = read(f, real)
   end
   h = read(f, string);
   for i in range(6):
      si.ArrAge[i] = read(f, real)
   end
   h = read(f, string);
   for i in range(2):
      si.ArrPhy[i] = read(f, real)
   end
   h = read(f, string);
   for i in range(2):
      si.Arr2con[i] = read(f, real)
   end

   h = read(f, string);
   si.TmtAll = read(f, int);
   si.TmtPer = read(f, real);
   h = read(f, string); #s
```

```
    for i in range(DC):
        si.TmtSpec[i] = read(f, int)
    end
    h = read(f, string);
    for i in range(4):
        si.TmtCol[i] = read(f, int)
    end
    h = read(f, string);
    for i in range(6):
        si.TmtAge[i] = read(f, int)
    end
    h = read(f, string);
    for i in range(2):
        si.TmtPhy[i] = read(f, int)
    end
    h = read(f, string);
    for i in range(2):
        si.Tmt2con[i] = read(f, int)
    end

    close(f);
    return si
end

func list int calck(list real v):
    int  k;
    real c, t;
    list(DS) int ds;

    for i in range(size(v)):
        t = t + v[i]
    end

    for i in range(DS):
        if ( i / DS ) >= ( v[k] / t + c ):
            c = c + v[k] / t;
            k = k + 1;
        end;
        ds[i] = k;
    end;
    return ds;
end

func t_treat read_treatment_input(string s):
    file f = open(s, "r");
    t_treat tmt;
    real m, v;
    string h;
    for i in range(8):
        h = read(f, string)
    end
    for i in range(DC):
        for j in range(4):
            for k in range(6):
                for l in range(9):
                    for n in range(2):
                        for o in range(2):
                            m = read(f, real);
                            v = read(f, real);
                            tmt.a[i][j][k][l][n][o] = m*m/v;
                            tmt.b[i][j][k][l][n][o] = v/m
                        end
                    end
                end
            end
        end
    end
    close(f);
    return tmt
end

func arrival read_arr_input(siminits si):
    int weekday = si.weekday;
    list(7) string d_day = ["mon", "tue", "wed", "thu", "fri", "sat", "sun"];
```

```
    file f = open("arrival" + d_day[weekday] + ".txt", "r");
    arrival arr;
    string h;

    list (2) list(DC) real          ds;
    list (2) list(DC) list(4) real dc;
    list(DC) list(6)  real da;
    list(DC) list(DC) real d2;

    h = read(f, string);
    for i in range(AR):
       for j in range(2):
          arr.rate[j][i] = read(f, real) * si.ArrAll
       end;
    end;

    h = read(f, string);
    for i in range(DC):
       for j in range(2):
          ds[j][i] = read(f, real) * si.ArrSpec[i]
       end;
    end;
    h = read(f, string);
    for i in range(DC):
       for k in range(2):
          for j in range(4):
             dc[k][i][j] = read(f, real) * si.ArrCol[j]
          end;
       end;
    end;
    h = read(f, string);
    for i in range(DC):
       for j in range(6):
          da[i][j] = read(f, real) * si.ArrAge[j]
       end;
    end;
    h = read(f, string);
    for i in range(DC):
       arr.phys[i] = read(f, real) * ( si.ArrPhy[1] / si.ArrPhy[0] )
    end;
    h = read(f, string);
    for i in range(DC):
       for j in range(4):
          arr.cons2[i][j] = read(f, real) * ( si.Arr2con[1] / si.Arr2con[0] )
       end;
    end;
    h = read(f, string);
    for i in range(DC):
       for j in range(DC):
          d2[i][j] = read(f, real)
       end;
    end;

    arr.spec[0] = calck(ds[0]);
    arr.spec[1] = calck(ds[1]);
    for i in range(DC):
       arr.col[0][i]  = calck(dc[0][i]);
       arr.col[1][i]  = calck(dc[1][i]);
       arr.age[i]     = calck(da[i]);
       arr.cons2by[i] = calck(d2[i])
    end;
    close(f);
    return arr
end

func roster read_schedule_input(int weekday):
    roster ros;
    int n;
    string h;
    list(7) string d_day = ["mon", "tue", "wed", "thu", "fri", "sat", "sun"];
    file f = open("schedule" + d_day[weekday] + ".txt", "r");
    list(DS) staff st;
    list(DS) real t;

```

```
   n   = read(f, int);

   for i in range(DC+3):
      h = read(f, string);
   end

   for i in range(n):
      t[i] = read(f, real) * 60.0;
      st[i].t = read(f, int);
      st[i].n = read(f, int);
      st[i].s = read(f, int);
      for j in range(DC):
         st[i].p[j] = read(f, int);
      end
   end

   for i in range(1,n):
      st[n-i].t =  st[n-i].t - st[n-i-1].t;
      st[n-i].n =  st[n-i].n - st[n-i-1].n;
      st[n-i].s =  st[n-i].s - st[n-i-1].s;
      for j in range(DC):
         st[n-i].p[j] =  st[n-i].p[j] - st[n-i-1].p[j];
      end
   end

   close(f);
   ros.t  = t[:n];
   ros.st = st;

   return ros;
end

func patient CreatePatient(arrival arr; int spe, col, age, ext, amb,
                           c2b, i; real phy, co2, tim; siminits si):
   patient x;
   int cp;

   x.t.simday  = i;
   x.t.total.b = tim;

   x.spec      = arr.spec[amb][spe];
   x.color     = arr.col[amb][x.spec][col];
   x.age       = arr.age[x.spec][age];
   x.st.t      = 1;
   x.amb       = min(si.amb, amb);

   if x.color == 0:
      cp = max(2, ceil(ext / 2));
      x.st.n = ext;
   else:
      cp = 1;
      x.st.n = 1
   end

   if arr.phys[x.spec] > phy:
      x.st.p[x.spec] = cp
   else:
      x.st.s = cp;
   end

   if arr.cons2[x.spec][x.color] > co2:
      x.cons2 = 1;
      x.st2.p[arr.cons2by[x.spec][c2b]] = 1
   end

   return x
end

proc G(chan! patient a; int i, amb; arrival arr;
       chan! void pa; real tstart; siminits si):
   patient    x;
   int        n;
   real       t, h = 60.0;
   dist real  darr  = exponential(1.0);
```

```
   dist real  phys  = uniform(0.0, 1.0),
              cons2 = uniform(0.0, 1.0);
   dist int   age   = uniform(0, DS),
              spec  = uniform(0, DS),
              color = uniform(0, DS),
              co2by = uniform(0, DS),
              dplus = uniform(2, 6);

   t = sample darr * h / arr.rate[amb][0];

   while ( time - tstart + t ) < ( h * 24 ):
      delay t;

      x = CreatePatient(arr, sample spec, sample color, sample age, sample dplus,
                        amb, sample co2by, i, sample phys, sample cons2, time, si);
      a!x;
      pa!;

      n = floor((time - tstart) / h);
      t = sample darr * h / arr.rate[amb][n];
      while floor( ( time - tstart + t ) / h ) > n and n < 23:
         n = n + 1;
         t = sample darr * h / arr.rate[amb][n];
         delay tstart + h * n - time;
      end
   end
end

proc D(chan patient a,d; chan void e):
   real m = 8.0;
   dist real dtransp = exponential(m);
   list patient xs;
   patient x;
   list timer ts;
   bool go = true;
   real dt;

   while go:
      select
         a?x:
            xs = xs + [x];
            dt = sample dtransp;
            if x.color == 0:
               dt = dt / 4
            end
            ts = ts + [timer(dt)]
      alt
         unwind i in range(size(xs)):
            size(xs) > 0 and ready(ts[i]), d!xs[i]:
               xs = xs - [xs[i]];
               ts = ts - [ts[i]]
         end
      alt
         e?:
            go = false
      end
   end
end

func list(2) int dispatch_triage(list(4) list patient xss;
                                 list(4) int cnts; staff ast;
                                 list(4) int dc):
   int kout = 12,
         k = calc(cnts);
   real   t = maxreal;
   list int ks1 = [0,1], ks2 = [2,3], ks;

   if k == 4:
      return [kout, 0]
   elif k < ks2[1]:
      ks = ks1
   else:
      ks = ks2
```

```
   end

   for i in ks:
      if cnts[i] > 0 and xss[i][0].t.total.b < t:
         t    = xss[i][0].t.total.b;
         kout = i
      end
   end

   if kout < 12 and st_check(ast, xss[kout][0].st, dc) == false:
      kout = 12
   end;
   return [kout, 0];
end

func list(2) int dispatch_treatment(list(12) list patient xss;
                                    list(12) int cnts; staff ast;
                                    list(4)  int dc; int ra, rr):
   # dispatch the first patient from the most urgent occupied
   # triage level with enough available capacity
   int k0   = calc( cnts[ :4]  ),
       k1   = calc( cnts[4:8]  ) + 4,
       k2   = calc( cnts[8:12] ) + 8,
       kout = 12,
       i = 0, j1 = 0, j2  = 0;
   bool xgo = false;

   if k2 < 12:
      while i < cnts[k2] and xgo == false:
         xgo = st_check(ast, xss[k2][i].st, dc);
         i   = i + 1;
      end
      return [k2, i - 1];
   end

   if k0 < k1 - 4:
      while i < cnts[k0] and xgo == false:
         xgo = st_check(ast, xss[k0][i].st, dc);
         i   = i + 1
      end
      kout = k0
   elif k0 == k1 - 4 and k0 < 4:
      while (j1 + j2) < (cnts[k0] + cnts[k1]) and xgo == false:
         if j1 < cnts[k0] and
             (j2 == cnts[k1] or xss[k0][j1].t.total.b < xss[k1][j2].t.total.b):
            xgo = st_check(ast, xss[k0][j1].st, dc);
            j1 = j1 + 1;
            kout = k0;
            i = j1;
         elif j2 < cnts[k1]:
            xgo = st_check(ast, xss[k1][j2].st, dc);
            j2 = j2 + 1;
            kout = k1;
            i = j2;
         end
      end
   elif k1 < 8:
      while i < cnts[k1] and xgo == false:
         xgo = st_check(ast, xss[k1][i].st, dc);
         i   = i + 1
      end
      kout = k1
   end;

   if (xgo == false or ra <= rr) and kout != 0 and kout != 4:
      kout = 12;
   end

   return [kout, i - 1];
end

func int calc(list int cnts):
   int kmax = size(cnts);
   for k in range(kmax):
```

```
      if cnts[k] > 0: return k end
   end;
   return kmax
end

func bool st_check(staff st_avail, st_pat; list(4) int a):
   if (a[0] > 0 and st_avail.t < st_pat.t) or
      (a[1] > 0 and st_avail.n < st_pat.n) or
      (a[2] > 0 and st_avail.s < st_pat.s):
      return false
   end
   for i in range(DC):
      if a[3] > 0 and st_avail.p[i] < st_pat.p[i]:
         return false
      end
   end
   return true
end

func staff st_update(staff st_avail, st_pat; list(4) int a):
   st_avail.t      = st_avail.t    - a[0] * st_pat.t;
   st_avail.n      = st_avail.n    - a[1] * st_pat.n;
   st_avail.s      = st_avail.s    - a[2] * st_pat.s;
   for i in range(DC):
      st_avail.p[i] = st_avail.p[i] - a[3] * st_pat.p[i];
   end
   return st_avail
end

func staff modWIP(staff wip, st; int i):
   wip.n = wip.n + i;
   if st.s > 0:
      wip.s = wip.s + i;
   else:
      for r in range(DC):
         if st.p[r] > 0:
            wip.p[r] = wip.p[r] + i;
         end
      end
   end
   return wip
end

proc U(chan strm sa; roster ros):
   for i in range(1,size(ros.t)):
      delay ros.t[i] - ros.t[i-1];
      sa!(false, ros.st[i])
   end
end


proc W(chan patient a; list(2) chan patient b; chan stlvl p;
       chan strm sa; chan void e; roster ros; int day; siminits si):
   list(3)  list(4) int dc = [[ 1, 0, 0, 0], [ 0, 1, 1, 1], [-1,-1,-1,-1]];
   list(12) list patient xss;
   list(12) int cnts;
   list(2)  list(2) int k;
   patient  x;
   staff wip, ast = ros.st[0];
   bool  go = true;
   int   idx, rdx, ra = si.NR, rr = si.NRR;
   strm  strmx;

   start U(sa,ros);
   p!(day, time, ast);

   while go:

      k[0] = dispatch_triage(xss[ :4], cnts[ :4], ast, dc[0]);
      k[1] = dispatch_treatment(xss, cnts,   ast, dc[1], ra, rr);
      if k[0] == k[1]:
         k[1][0] = 12
     end
      if k[1][0] < 12:
```

```
            xss[k[1][0]][k[1][1]].wip = wip
        end

        select
            a?x:
                idx = x.color;
                if x.amb == 1:
                    idx = idx + 8;
                elif x.t.triage.b != 0.0:
                    idx = idx + 4;
                end
                xss[idx]  = xss[idx] + [x];
                cnts[idx] = cnts[idx] + 1;
        alt
            unwind i in range(2):
                k[i][0] < 12, b[i]!xss[k[i][0]][k[i][1]]:
                    idx       = k[i][0];
                    rdx       = k[i][1];
                    x         = xss[idx][rdx];
                    ast       = st_update(ast, x.st, dc[i]);
                    xss[idx]  = xss[idx] - [xss[idx][rdx]];
                    cnts[idx] = cnts[idx] - 1;
                    if i == 1:
                        ra = ra - 1;
                        wip = modWIP(wip, x.st, 1)
                    end
                    p!(day, time, ast)
            end
        alt
            sa?strmx:
                ast = st_update(ast, strmx.st, dc[2]);
                p!(day, time, ast);
                if strmx.rm:
                    ra  = ra + 1;
                    wip = modWIP(wip, strmx.st, -1)
                end
        alt
            e?:
                go = false
        end
    end
end

proc dT(chan real dt; real a, b):
    dist real dtreat  = gamma(a, b);
    dt!(sample dtreat)
end

proc T(chan patient a, b; chan strm sa; chan void e; chan real dt):
    staff st, st_empty;
    real ttriage;
    patient x;
    bool go = true;

    while go:
        select a?x:
            x.t.triage.b = time;
            start dT(dt, 3.0, 3.5);
            dt?ttriage;
            delay ttriage;
            x.t.triage.e = time;
            st = st_update(st_empty, x.st, [-1,0,0,0]);
            sa!(false,st);
            b!x
        alt
            e?:
                go = false
        end
    end
end

func real adapttmt(patient x; siminits si; real ttreat, dtmt; int doctype):
    if dtmt < si.TmtPer:
        ttreat = ttreat + si.TmtAll
```

```
    end
    ttreat = ttreat + si.TmtSpec[x.spec]
                    + si.TmtCol[x.color]
                    + si.TmtAge[x.age]
                    + si.TmtPhy[doctype]
                    + si.Tmt2con[x.cons2];
    return max(0,ttreat)
end

proc R(chan patient a, b; chan strm sa; chan void e; chan real dt;
       t_treat tmt; siminits si; int i):
    dist real dhigh   = triangle(15.0, 20.0,  30.0);
    dist real dtmt    = uniform(0.0, 1.0);

    staff st_empty, st_back, st_pat, st_temp;
    real  ttreat, ta, tb;
    int   doctype, wip, n, k;
    bool  t2b, trb, go = true;
    patient x;
    timer tt,tr,t2;

    while go:
       select a?x:
          x.room      = i;
          x.t.treat.b = time;
          st_pat = x.st;

          if x.st.s > 0:
             doctype          = 0;
             wip              = min(7,x.wip.s)
          else:
             doctype          = 1;
             wip              = min(7,x.wip.p[x.spec])
          end
          ta      = tmt.a[x.spec][x.color][x.age][wip][doctype][x.cons2];
          tb      = tmt.b[x.spec][x.color][x.age][wip][doctype][x.cons2];
          start dT(dt, ta, tb);
          dt?ttreat;
          ttreat = adapttmt(x, si, ttreat, sample dtmt, doctype);

          tt = timer(ttreat);
          t2 = timer(x.cons2 * real(t2) * 0.5);  t2b = true;
          tr = timer( min( sample dhigh, real(t2) ) ); trb = true;

          delay min(real(t2),real(tr));
          for j in range(2):
             select ready(t2) and t2b:
                k = max( max(x.st2.p), x.st2.s );
                n = max( max(st_pat.p), st_pat.s );
                st_temp = st_update(st_empty, st_pat, [0, 0, -k, -k]);
                st_back = st_update(st_temp, x.st2, [0, 0, n, n]);
                sa!(false, st_back);
                st_pat  = st_update(st_pat, st_back, [0, 0, 1, 1]);
                t2b     = false
             alt ready(tr) and trb:
                st_temp   = modWIP(st_empty, st_pat, -1);
                st_back   = st_update(st_temp, st_pat, [0, -1, -1, -1]);
                sa!(false, st_back);
                st_pat = modWIP(st_empty, st_pat, 1);
                trb    = false
             end;
             delay max(real(t2),real(tr))
          end

          delay real(tt);
          x.t.treat.e = time;
          sa!(true, st_pat);
          b!x;
       alt
          e?:
             go = false
       end
    end
end
```

```

proc E(chan? patient a; chan! patient p; chan void e):
   patient x;
   bool go = true;

   while go:
      select a?x:
         x.t.total.e = time;
         p!x;
      alt
         e?:
            go = false
      end
   end
end

proc P(chan? void pa; chan? patient pe; chan? stlvl pst;
       chan void go; file f_pat, f_st; int i; real t; int NR):
   list(4)  string d_color = ["Red", "Yellow", "Green", "Blue"];
   list(12) string d_spec  = ["ACHt", "ACHn", "INT", "CAR", "ORT", "KIN",
                              "LON",  "NEU",  "URO", "GYN", "PCH", "GER"];
   string h_pat1 = "Day\tSpecialism\tAge\tColor\ttb_total\tte_total\t",
          h_pat2 = "dt_total\ttb_triage\tte_triage\tdt_triage\ttb_treat\t",
          h_pat3 = "te_treat\tdt_treat\tdt_wait\tAmbulance\tRoom",
          h_st1  = "Day\tTime\tTriagenurse\tNurse\tSEH\tACHt\tACHn\tINT\t",
          h_st2  = "CAR\tORT\tKIN\tLON\tNEU\tURO\tGYN\tPCH\tGER";
   real dt_total, dt_triage, dt_treat, dt_wait;
   bool amb, go_again = true, streceived = false;
   timelist tl;
   real tnow;
   patient x;
   staff st;
   stlvl s;
   int dx, j, k;
   timer tt = timer(0.0);

   if i == 0:
      writeln(f_pat, "%s%s%s", h_pat1, h_pat2, h_pat3);
      writeln(f_st, "%s%s", h_st1, h_st2);
   end

   while go_again:
      tnow = time;
      select
         pa?:
            dx = dx + 1
      alt
         pe?x:
            dx = dx - 1;
            tl = x.t;
            dt_total  = tl.total.e  - tl.total.b;
            dt_triage = tl.triage.e - tl.triage.b;
            if dt_triage == 0:
               tl.triage.b = t;
               tl.triage.e = t
            end
            dt_treat  = tl.treat.e  - tl.treat.b;
            dt_wait   = tl.treat.b  - tl.total.b - dt_triage;
            if x.amb == 1:
               amb = true
            else:
               amb = false
            end
            writeln(f_pat ,"%d\t%s\t%d\t%s\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%b\t%d",
               tl.simday, d_spec[x.spec], x.age, d_color[x.color],
               tl.total.b-t, tl.total.e-t, dt_total, tl.triage.b-t, tl.triage.e-t,
               dt_triage, tl.treat.b-t, tl.treat.e-t, dt_treat, dt_wait, amb, x.room)
      alt
         pst?s:
         streceived = true
      alt
         ready(tt) and streceived:
```

```
                tt = timer(5.0);
                if j < 289:
                    j = j + 1;
                    st = s.st;
                    writeln(f_st,"%d\t%.2f\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d",
                        s.day, time-t, st.t, st.n, st.s, st.p[0], st.p[1], st.p[2],
                        st.p[3], st.p[4], st.p[5], st.p[6], st.p[7],
                        st.p[8], st.p[9], st.p[10], st.p[11])
                end
        end
        if dx == 0 and tnow > (1440.0 + t) and j >= 289:
            go_again = false;
            while k < (NR + 4):
                go!;
                k = k + 1
            end
        end
    end
end
```

Listing B.1: Chi 3 listings of the ED simulation model.