MASTER'S THESIS

Creation of a simulation model for the optimized design of a 3D printing centre

Stefan Paal BSc.

in order to obtain the academic degree

Diplom-Ingenieur

Master's degree Mechanical Engineering Business Economics

under supervision of

Dipl.-Ing. Philipp Url BSc.

Ass.Prof. Dipl.-Ing. Dr.-techn. Nikolaus Furian

Ass.Prof. Dipl.-Ing. Dr.-techn. Wolfgang Vorraber

Institute for Mechanical Engineering and Business Informatics
Graz University of Technology

Graz, June 2020

AFFIDAVIT

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master’s thesis.

27 August 2020

………………………… ………………………………………………..

Date Signature

Abstract

As part of the COMET-K project CAMed (Clinical Additive Manufacturing for Medical Applications), a 3D printing centre (3D-PC) was set up at the LKH Med Uni Graz. In the future, patient-specific implants, special tools and anatomical models are to be manufactured in this centre. Scientific and industrial partners are working together to implement this manufacturing centre, furthermore, to create the process chain and the necessary infrastructure.

Since this project was a new concept, which had not yet been implemented in this form, it was difficult to make predictions about the future requirements for the 3D-PC - and also for the entire process. This also complicated estimating the required infrastructure. The future workload of the system, as well as the requirements for it, depend on various factors (application area, costs, availability, etc.). These factors could have changed later, which meant, that the requirements for the system to be operated in the future could not be predicted exactly at the planning time of this thesis.

In order to design the system and optimize it with regard to its requirements, a model was developed in this master thesis. The model considers all possible influencing factors and includes performance indicators of the 3D-PC.

The aim was to use this model to obtain clear requirements for the 3D-PC (number of workstations, number and type of printers, number of hired employees, etc.). The model will be used to design the system based on reports and data from the past. In the future, this model shall be updated with more precise requirements to further optimize the 3D-PC.

Acknowledgments

At the outset of this work, I would like to extend my sincere gratitude to all those who have supported me in this thesis, but also to those who have guided me in my preparation for it over the past few years.

First of all, I would like to thank **Mr. Dipl.-Ing. Philipp Url BSc.** for his mentoring and support, which I was always sure of despite the circumstances during the COVID-19 pandemic.

No less my thanks go to **Mr. Ass.Prof. Dipl.-Ing. Dr.-techn. Nikolaus Furian** for his detailed explanations and his comprehensive advice and support about software and programming.

I extend my gratitude to **Mr. Ass.Prof. Dipl.-Ing. Dr.-techn. Wolfgang Vorraber** and the *Institute for Mechanical Engineering and Business Informatics* for giving me the opportunity to work on this project and I would like to thank everyone involved.

Also, this work was supported by CAMed (COMET K-Project 871132) which is funded by the Austrian Federal Ministry of Transport, Innovation and Technology (BMVIT) and the Austrian Federal Ministry for Digital and Economic Affairs (BMDW) and the Styrian Business Promotion Agency (SFG).

From the bottom of my heart, I would like to express my thankfulness to **my parents and family members**, who have always supported me morally and economically during my studies.

And last but not least my gratitude goes to all my friends who directly or indirectly helped and guided me through my study and this master's thesis.

Thank you!

Content

List of Abbreviations

3D	3 Dimensional
3D-PC	3D Printing Centre
CAD	Computer Aided Design
MRI	Magnetic Resonance Imaging
CT	Computed Tomography
DES	Discrete Event Simulation
SD	System Dynamics
AB	Agent Based Simulation
CM	Conceptual Model
HCCM	Hierarchical Control Conceptual Modelling
CU	Control Unit
UML	Unified Modelling Language
ERM	Entity Relationship Model
TCO	Total Cost of Ownership
ROI	Return on Investment
IT	Information Technology
FDM	Fused Deposition Modelling
FFF	Fused Filament Fabrication
DED	Direct Energy Deposition
SLS	Selective Laser Sintering
SLM	Selective Laser Melting
EBM	Electron Beam Melting
MJF	Multi Jet Fusion
UV	Ultraviolet
STL	Stereolithography
DLP	Digital Light Processing
PLA	Polylactide Acid
ABS	Acrylonitrile Butadiene Styrene
PA	Polyamid / Nylon
PETG	Polyethylenterephthalate + Glycerol

Table 0.1 List of Abbreviations sorted by their appearance

1 Introduction and Task Definition

Behind every running production, there are a lot of different processes and even more planning activities to coordinate them. To start a new production, it is therefore not only important to know what to do, but also how much of what to do and when. The COMET-K project CAMed (Clinical Additive Manufacturing for Medical Applications) brought up exactly this task.

With the project a 3D printing centre (3D-PC) will be built at the LKH Graz, which does not exist in such a form anywhere else. The aim is to produce patient-specific implants, anatomical models (used for surgical planning and doctor patient consultations) as well as tools that assist in individual patient care and surgeries. For this purpose, Computer Aided Design (CAD) models of the required parts are generated from the data of imaging procedures (e.g. Magnetic Resonance Imaging (MRI), Computed Tomography (CT)). These models are then produced on site with 3D printers.

To ensure an economical and reliable workflow within the 3D-PC, data is needed to plan the resources of the whole production. It is essential to define its possible production capacity in a way that future requirements can be met in an optimized way. To sum it up, the boundary conditions for an economical production line have to be outlined.

It is to be expected that the boundary conditions for the design depend on the future orders. As the project was still in an early phase at the time of this thesis, it was not clear how many orders could be expected and how they would differ. The idea was therefore to use a software to simulate the entire planned 3D-PC. This would allow to investigate different scenarios and subsequently optimize the design of the 3D-PC.

The aim of this work was to identify key performance indicators for the workload and the profitability of the 3D-PC and to create a corresponding simulation tool. The simulation should allow a comparison of different scenarios, on the basis of key performance indicators to determine the boundary conditions for the 3D-PC to be built.

2 3D Printing

“3D printing is developed as a modification of an old inject printer. Today, it is rapidly expanding: almost every week new printers and printing materials offering novel possibilities as well as new exciting applications appear” (Dodziuk, 2016, p. 1). With 3D printers it is possible to manufacture even complex geometries without having to acquire expensive special tools. Thus, it is not surprising that also the fields of application for 3D printing gets bigger and bigger every day. The use of 3D printers for medical purposes is quite new, but nevertheless already very advanced due to the rapid pace at which new methods, technologies and materials are developed. Today it is even possible to print bones as implants with the inner structure like in real human bones (see Figure 2.1). For this reason, the application of 3D printing in medical areas is increasing. (Dodziuk, 2016)

Figure 2.1 Bone scaffold fabricated by 3D printing (Cai et al., 2012)

2.1 3D Printing Techniques

Various methods for printing parts with 3D-printers exist. Each method has its field of application and different properties, which also depends on the used material. The most common methods are listed below:(3dhubs, 2020; Additively, 2020; Mann & Thum, 2020)

- Material extrusion
  To extrude a material, it is heated up above its melting point, the melted material then gets pressed through an extrusion chamber and through a nozzle at the end of the chamber. Mostly the material is transported with a conveyor screw within a heated extrusion chamber. The nozzle defines the thickness and the shape of the extruded material (Table 2.1). After the material passes the nozzle it cools down and hardens in the given shape. A major advantage of this technique is that it is cheap. All thermoplastics can be used. Methods for extrusion are Fused Deposition Modelling (FDM), Fused Filament Fabrication (FFF) or Direct Energy Deposition (DED). DED provides the advantage that the material is melted by a laser beam just before the material gets applied. The material is heated up for a shorter while and so the risk of thermal degradation is minimized. (3dhubs, 2020)

Min. layer	0,178mm
Accuracy	+/- 0,178mm

Table 2.1 Data extrusion (3dhubs, 2020)

- Powder bed fusion
  The power bed fusion process works, as the name suggests, with materials in powder form, which are distributed as a layer on a surface and welded at the desired points with a heat source. Then a layer of powder is applied again and melted together again by heat at the right place (Table 2.2). In this way, the desired part is created layer by layer. Technologies for this are Selective Laser Sintering (SLS), Selective Laser Melting (SLM), Electron Beam Melting (EBM) and Multi Jet Fusion (MJF). SLM is often used for medical applications such as crowns for teeth or smaller implants. (3dhubs, 2020)

Min. layer	0,03mm
Accuracy	+/- 0,1mm

Table 2.2 Data powder bed fusion (3dhubs, 2020)

- <u>Photopolymerization</u>
  The photopolymerization technology works with materials (e.g. resins) that react with ultraviolet (UV) or laser light. A thin layer of resin is applied on a surface and light beam is pointing where the material shall be solidified. One of the photopolymerization techniques is Stereolithography (STL) where a UV laser points into a resin bath and hardens the resin point by point. Another method is Digital Light Processing (DLP), which is quite similar to STL with the difference that not a single laser beam is shining, but a complete picture of the shape produced by a digital light projector, therefor it is slightly faster than STL (Table 2.3). (3dhubs, 2020)

Min. layer	0,016mm
Accuracy	+/- 0,05mm

Table 2.3 Data photopolymerization (3dhubs, 2020)

- <u>Jetting</u>
  Jetting is pretty similar to printing on a paper and is also often compared to it. Jetting can be divided into material jetting and binder jetting. The first one uses multiple printing heads to spray molten wax onto a flat layer (Table 2.4). The second one, Binder jetting, applies a thin binder layer on a powder layer to harden the material. (3dhubs, 2020)

Min. layer	0,01mm
Accuracy	+/- 0,05mm

Table 2.4 Data jetting (3dhubs, 2020)

At this point it should be mentioned that the accuracies and minimum layer thicknesses are not limit values but represent guide values for the respective production methods.

2.2 **Materials**

For 3D printing many different materials can be used and more and more new materials are designed for special purposes. This chapter shall give a short overview over the most common materials for 3D printing and particularly the ones which are used in this project. It happens that the most used materials are polymers, because of their cheap price and low melting points.

- PLA (Polylactide Acid)
  PLA is, beside ABS, the most common thermoplastic for 3D printing. This is caused by its cheap price and its low melting point which makes it easy to use (Table 2.5). (3dhubs, 2020)

melting point	160-190 °C
tensile strength	37 MPa

Table 2.5 Data PLA (3dhubs, 2020)

- ABS (Acrylonitrile Butadiene Styrene)
  ABS is a thermoplastic, that is easy to handle, though it is not biodegradable. It is widely used for electronic housings, car parts etc.. One of ABS's biggest problems are the vapours that it produces when the material gets heated (Table 2.6). (3dhubs, 2020)

melting point	220-250 °C
tensile strength	27 MPa

Table 2.6 Data ABS (3dhubs, 2020)

- PETG (Polyethylenterephthalate + Glycerol)
  PETG is a thermoplastic which is quite similar to PLA but with better mechanical properties (Table 2.7). This material was particularly designed for 3D printing and produces hardly any vapours by heating it up. (3dhubs, 2020)

melting point	220-235 °C
tensile strength	50 MPa

Table 2.7 Data PETG (3dhubs, 2020)

- PA (Polyamide)
  Polyamide also called Nylon is a thermoplastic and often used for gliding parts and moveable connections because its low friction (3dhubs, 2020).

melting point	220-260 °C
tensile strength	55 MPa

Table 2.8 Data PA (3dhubs, 2020)

- PEEK (Polyetheretherketon)
  "PEEK is an engineering thermoplastic with excellent mechanical and thermal properties. Can be used to replace metal parts" (3dhubs, 2020). PEEK has only a limited UV resistance but is resistant against nitric or sulphuric acids, almost all organic and inorganic chemicals. For those reasons, it suits perfectly for medical uses, despite its very high material price. (Han et al., 2019)

melting point	335 °C
tensile strength	95 MPa

Table 2.9 Data PEEK (3dhubs, 2020)

Figure 2.2 provides an overview of the differences between polymers.

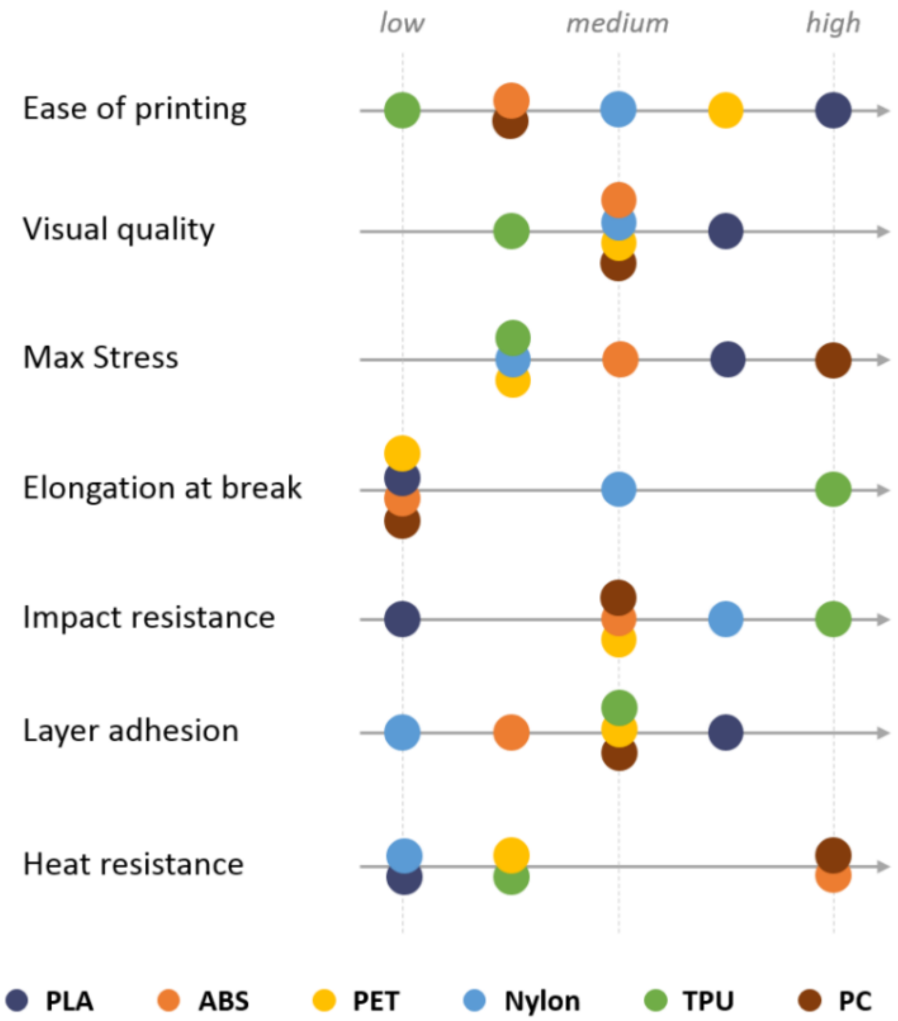

Figure 2.2 Research results for all six polymers displayed in one graph (3dhubs, 2020)

Beside polymers it is also possible to use metals and other materials like ceramics, but these require special printers and result in much higher prices. Therefore the use of metal printed parts is very low.(3dhubs, 2020)

2.3 Types of 3D Printers

Actually on the market you can find many different types of 3D printers that work with the techniques, that where described in chapter 2.1 “3D Printing Techniques” and many more. Figure 2.3 shows a classification scheme of 3D printers, whilst nearly all combinations of materials, designs and applications are possible.

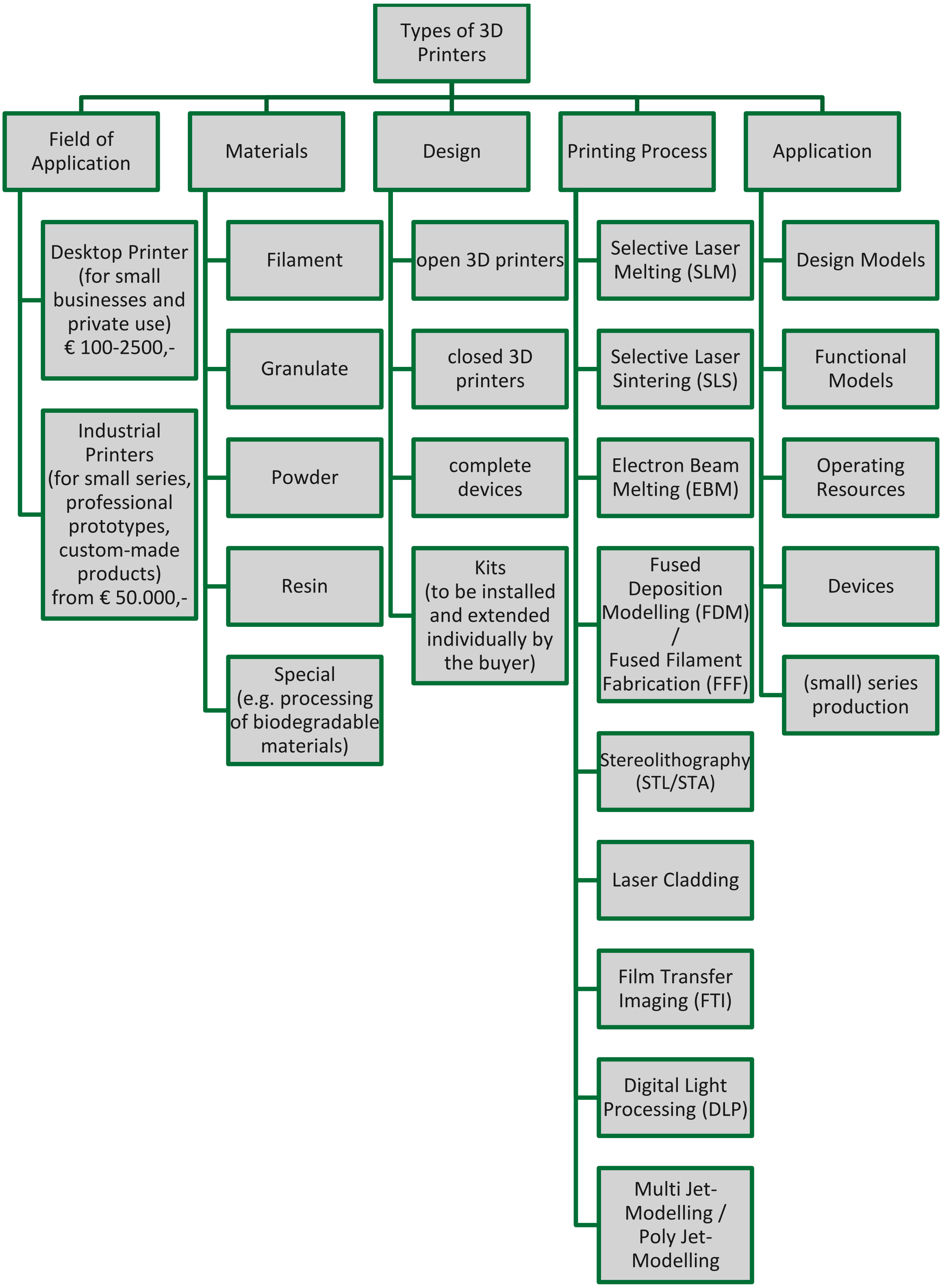

Figure 2.3 Classification of 3D printer (illustrated by author based on data by *3dhubs*, 2020; *additively*, 2020)

3 Theory of Simulation

This chapter deals with the definition of a simulation, its areas of application, the reasons for its use and what is required to get a suitable simulation.

3.1 Definition and Use of a Simulation

"[A simulation is an] imitation (on a computer) of a system as it processes trough time" (Robinson, 2004, p. 2 words in square braket added by author).

To understand this definition at first a system has to be defined and explained. A system is a set of elements that correlate with each other. Systems can be classified in four classes: natural systems, designed physical systems, designed abstract systems and human activity systems. (Checkland, 1981)

Most systems are a combination of different classes (e.g. an airport where the buildings and computer represent designed physical systems and the people within the airport, who are represented by a human activity system) (Robinson, 2004).

Contraire linear programming, that gives the user an optimal solution, or heuristic methods that provide near optimum answers, a simulation predicts the performance of a certain system. The simulation gives outputs on given inputs. So, these inputs have to be changed and defined as scenarios in order to get different results that can be compared. With a simulation it is also possible not only to compare different scenarios, but also different system designs and alternative policies, if the simulation is programmed flexible enough. A simulation does not provide optimal solutions or decisions, it gives outputs that shall help the user to obtain a sufficient understanding of the system and its functionalities. It is still the user's responsibility to compare the results from different scenarios and find an optimal solution by defining the right scenario. (Robinson, 2004)

A system, as defined above, can have many interconnections. This means one step is dependent from one or many other steps. To explain this with an example, if there is a variation in the required production A time for a part A1, it affects the start of post treatment process B, which works on part A1. Such interconnections make it very hard to predict influences of changes in a system, particularly in big systems. The more elements a system has, the more potential interconnections can exist (see Figure 3.1). (Robinson, 2004)

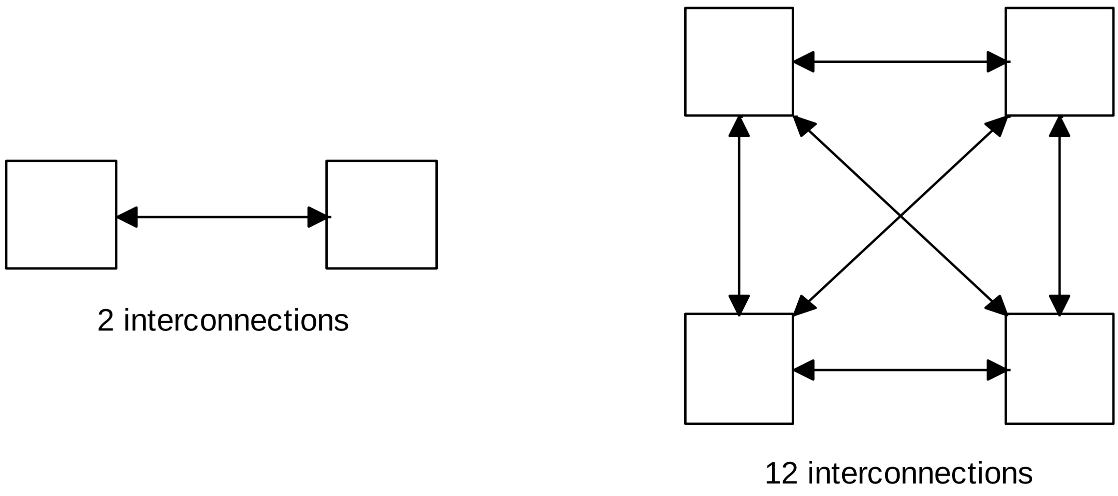

Figure 3.1 interconnection and combinatorial complexity (illustration by author based on Robinson, 2004)

Beside simulations there is still the possibility to do experiments on the real system. This would save the time for programming a simulation, but also take a lot more time do the experiments, since it is not possible to fast-forward in time. Beside the fact that experiments on a real system are only possible if the system already exists and is not in its planning phase, like it is the case at the CAMed project. Also, real system experiments are exposed to uncontrollable influences (e.g. weather, traffic, etc.). Though it has to be evaluated if not the costs for an experiment are higher than the one for a simulation. (Robinson, 2004)

3.2 **Problems with Simulations**

Simulations often need a high amount of data which can by either not available or have to be collected and prepared under great effort. This data collection and nevertheless the creation of the simulation program consume a lot of time to get to a suitable level. It is very important to have a mature simulation to get reliable results. No matter how carefully a simulation gets designed, the problem of overconfidence shall never be underestimated. A computer can display various scenarios and, if the simulation provides it, show quite realistic animations. Every scenario will create a result, but it is still in the user’s responsibility to ask if it is plausible and matches with the assumptions that were made prior. (Robinson, 2004)

3.3 **Simulation Paradigms**

A simulation paradigm can also be called a simulation method. There are mainly three different methods to simulate a system, Discrete Event Simulation (DES), System Dynamics (SD) and Agent Based Simulation (AB) (Maidstone, 2012). This chapter gives a short overview of these methods and highlights their differences in chapter 3.3.4 ”Comparison of Simulation Methods”. The DES was used within the project of this thesis and is hence explained a bit more in detail.

3.3.1 **Discrete Event Simulation (DES)**

For simulating it is often required to model processes over time. One method to do so it the DES (Robinson et al., 2010). *"In discrete-event simulation only the points in time at which the state of the system changes are represented"* (Robinson, 2004, p. 15). DES is based on entities, events and activities (which are represented by two events at their beginning and end). This means that the system gets modelled as a series of events, which are instants in time, every time a state change occurs. Figure 3.2 provides a graphical representation of a DES. (Robinson, 2004)

Figure 3.2 Part of a DES model created using SIMUL8 to model an A&E department. (Maidstone, 2012)

- <u>Entities</u>

  Entities are objects of a system that have properties. The entities can be classified in external and internal entities. External entities are explicitly created by the modeller (e.g. a technician that works in the 3D-PC). Internal entities are those, who are created by the simulation itself (e.g. a new order that gets created within an event called "order arrival"). (Schriber & Brunner, 2013)

  It is also possible to divide entities into active and passive entities. Active entities flow through the system and change their state, when they move (e.g. and order that is send from a customer to technician). Passive entities represent objects that stay at a fixed point in the system, but their state can be changed by events and/or active entities (e.g. a printer changes its state from "idle" to "in use" when an event "start printing" is executed). (Furian et al., 2015)

- <u>Events</u>

  Events - these are state-changes at certain points in time as mentioned above – can be divided into B (bounded) events and C (conditional) events. Bounded events are scheduled to occur at a predicted time, like for example the end of a well-known process. Conditional events depend on the conditions of the model. Whenever a certain condition comes up the event occurs. During a simulation process all events that are going to occur are stored in an event list and are removed from the list when they are executed. Between such events entities can move and change their state. The states of the entities are part of the system state. (Robinson, 2004)

- <u>Activities</u>

  For DES an activity (e.g. a printing process) is defined with two events, a start event (e.g. a printer is started and changes its state from idle to work) and an end event (e.g. a printer stops printing, changes its state to idle and a part is created). (Robinson, 2004)

The process of simulating with DES can be explained with the three phase simulation approach. The program continuously repeats the following A-, B- and C-phases (Robinson, 2004):

- A-phase: The time, where the next event occurs, is determined by inspection of the event list and the simulation clock gets set to time where the next event occurs.
- B-phase: All B events of the event list at current clock time are executed.
- C-phase: The systems conditions are checked and all C events, where the conditions meet the system's current condition are executed. After executing all C events the systems conditions are checked again and possible C events are executed, since a C event before possibly may have changed the state of the system. If no more C events are to execute because of the system state the simulation returns to A-phase, unless a defined simulation end is reached (this could be a B event "end of simulation").

3.3.2 **System Dynamics (SD)**

Another method for modelling a simulation is SD, which uses stocks, flows and delays. A system gets represented by stocks, that are storage items for objects (e.g. the number of orders waiting to be printed, or stock the number of finished produced parts). These stocks can only store items of the same type and with the same characteristics. The stocks are connected with flows, that represent the movement of objects from one stock to another. The third element, the delay, can be used to regulate flows or the arrival of objects at a stock. A typical illustration of an SD is shown in Figure 3.3. (Borshchev & Filippov, 2004; Maidstone, 2012)

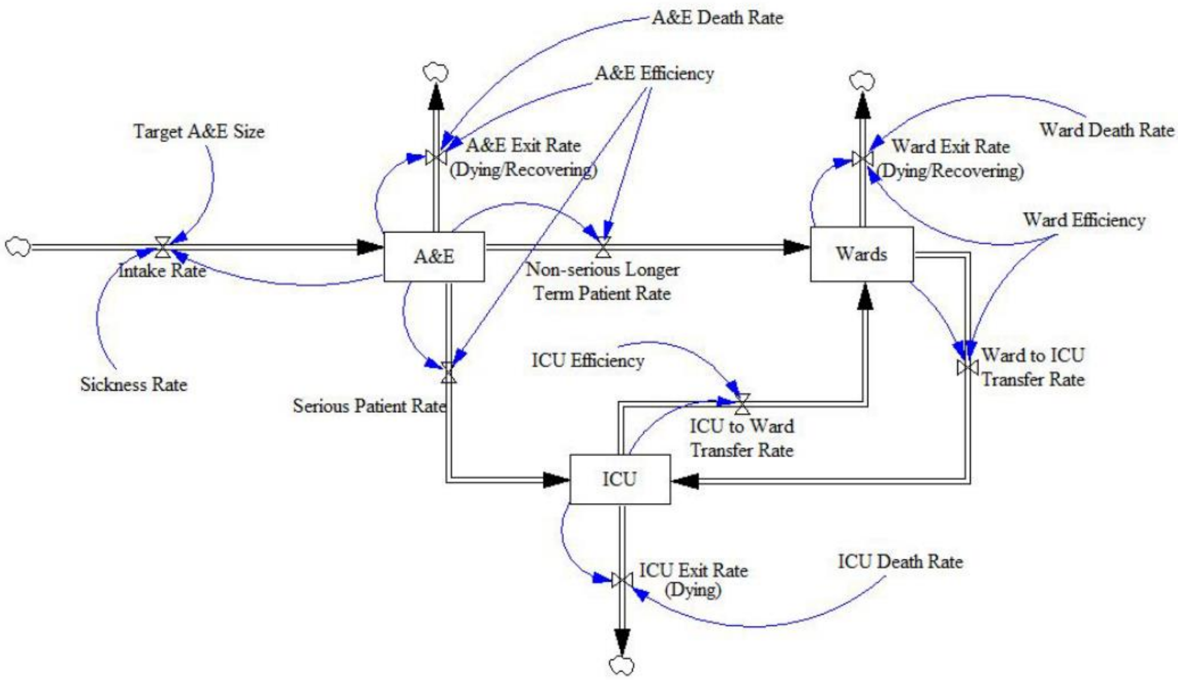

Figure 3.3 An SD model created using Vensim to simulate patients flow through a hospital. (Maidstone, 2012)

SD provides the advantage, that the system's behaviour can be predicted just by looking at the structure, when it is illustrated (Maidstone, 2012). As a negative aspect it can be said, that SD can mathematically be seen as a system of differential equations and the modeller has to think about structural dependencies, data correlations and provide accurate data (Borshchev & Filippov, 2004).

3.3.3 Agent Based Simulation (AB)

The agent based paradigm is the newest of the three mentioned methods in this thesis (see chapter 3.3.1 “Discrete Event Simulation (DES)” and 3.3.2 ”System Dynamics (SD)”). In comparison to DES and SD, AB does not provide any place to define the global system behaviour or to define global structure. As the name tells, AB is modelled with autonomous agents. (Maidstone, 2012)

While Borshchev & Filippov (2004, p. 6) claim that *“There are no universally accepted definitions […] what kind of properties an object should have to ‘deserve’ to be called an ‘agent’ ”.* Maidstone (2012) defines agents as self-directing objects which move in the system (see Figure 3.4). All agents have rules, that define how they interact with other agents and how they move about the system. The systems behaviour is therefore a result of the individual behaviours of all agents interacting. (Maidstone, 2012)

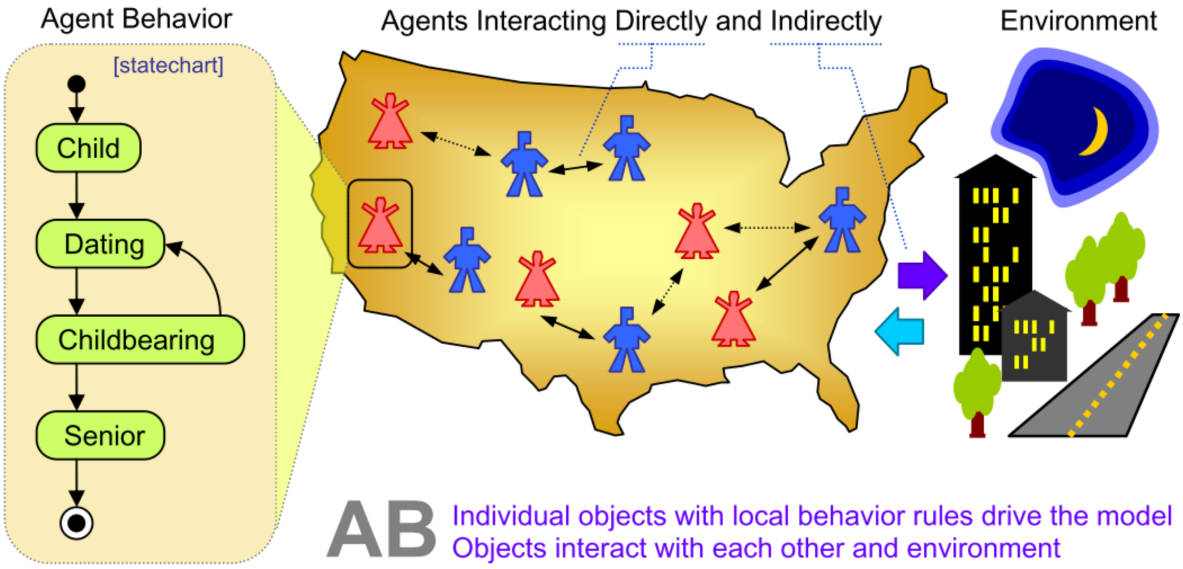

Figure 3.4 Agent Based Model Generic Architecture. Behaviour (Statechart) in AnyLogic™ (Borshchev & Filippov, 2004)

3.3.4 Comparison of Simulation Methods

All three simulation methods in the chapters above (3.3.1, 3.3.2 and 3.3.3) can be used to simulate systems and are applicable in different situations. Depending on the system one can be more appropriate than the other. DES is intended to be used for smaller systems and to take a more detailed look on processes of a system, whereas SD focuses more on the macroscopic perspective and provides an overview. Though AB can be used on big systems it is modelled at the micro-level, with rules and correlations between agents. This significant differences are illustrated in Figure 3.5. (Maidstone, 2012)

Compared to DES and SD where it is possible to define the global behaviour of the system, AB has no ability for it. The modeller describes the individual behaviours of agents and the global behaviour results from them. Therefore, AB is called bottom-up modelled, whilst SD is modelled top-down. (Borshchev & Filippov, 2004)

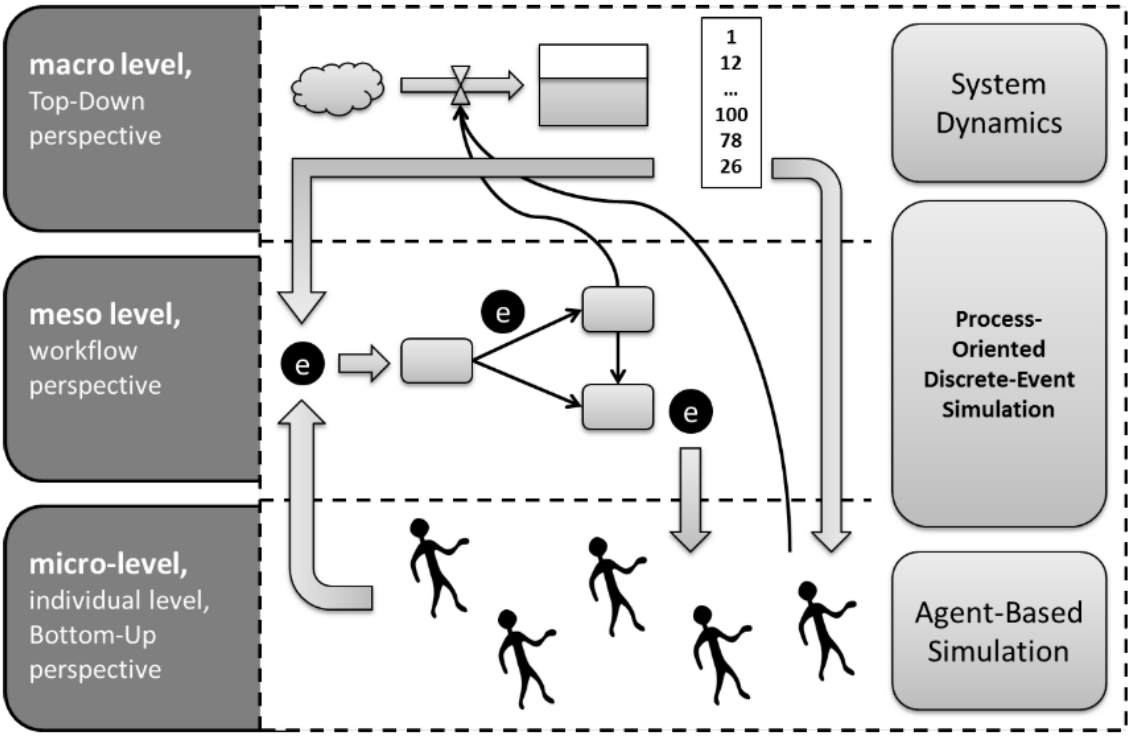

Figure 3.5 Overview of important ProHTA processes between abstraction levels. (Djanatliev & German, 2013)

DES is a stochastic simulation approach, which results in the fact, that different simulation runs create different results. Therefore, a DES model has to be run multiple times, so the user gains a significant result. SD models work deterministic, thus every simulation run gives the same result, if the parameters are constant. So the simulation only needs to be run once. (Maidstone, 2012)

Another difference exists between DES and AB. DES is based and built around queues and entities that flow through the system. In AB there is no concept at all for queues, since every agent acts according to its own rules. (Maidstone, 2012)

For many applications DES and SD are an efficient solution to simulate a system and get to appropriate results. AB is very often a lot harder to develop and is less efficient for small systems. However, AB provides a particularly advantage if a system contains active objects that interact and have individual behaviour. (Borshchev & Filippov, 2004)

What method is best to be used depends on the system. To avoid the tendency that a modeller uses the method he/she feels the most comfortable with, it is necessary to have a sufficient knowledge of different methods. (Maidstone, 2012)

3.4 The Conceptual Modelling Process

A simulation is now an abstraction of the real world, where all necessary connections between items are represented in a model. When the decision has been made to simulate a system, the system and its problems have to be analysed in order to get such a virtual model with all its processes. The modellers job is to understand the nature of the problem and to develop a model that deals with it (see Figure 3.6). This process is called conceptual modelling. (Robinson, 2004)

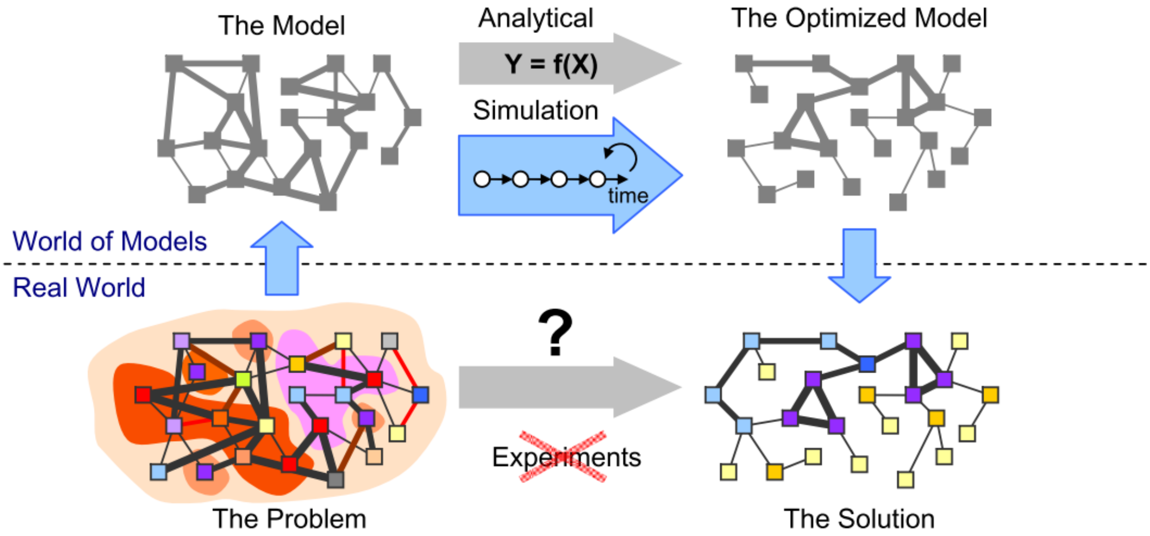

Figure 3.6 Analytical (Static) and Simulation (Dynamic) Modelling (Borshchev & Filippov, 2004)

Robinson (2004) claims that up to 50 percent of the benefits, when a simulation is done, don't derive from the simulation results, but from the development of a conceptual model (CM). Even today, where it is possible to design simulation software with useful software packages and derive prototypes of a simulation model within a short time, CM is important for building a sufficient understanding of the system. (Robinson, 2004)

In advance modern computers with faster processors and increasing memory resources tempt to design more complex, but less understood, systems. It is more important than ever to use CM and get simpler models. (Chwif et al., 2001)

Robinson (1994) illustrated the need for less complexity in Figure 3.7 and wrote, that more complex simulations are harder to understand and therefor their results are more difficult to interpret.

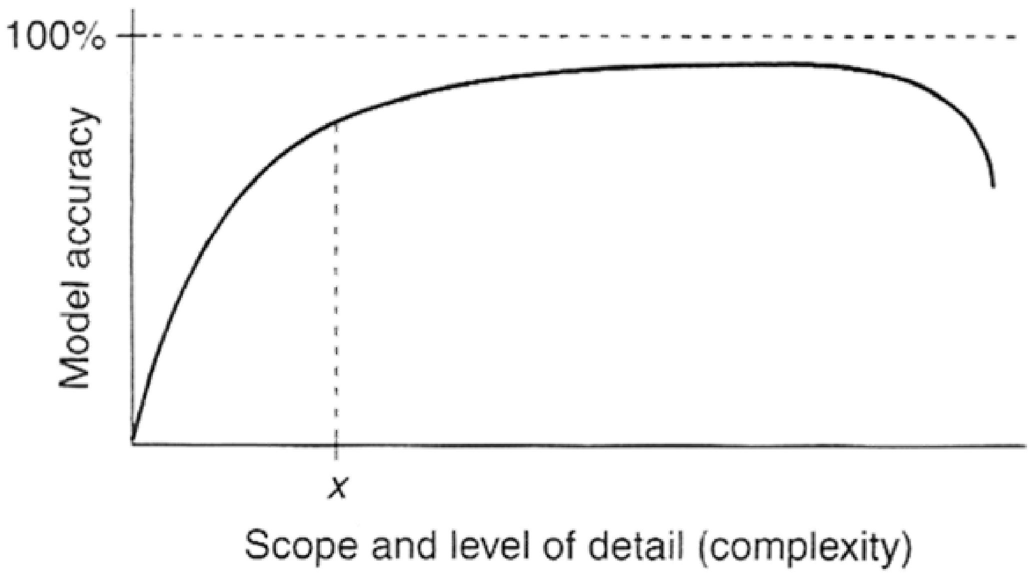

Figure 3.7 Simulation Model Complexity and Accuracy (Robinson, 2004)

Although the high importance of conceptual modelling, there are rarely any instructions and papers about how to do it. Robinson (2004) mentioned that conceptual modelling is more learned by experience but also gives an overview of methods. The conceptual modelling can be structured into the following sub-processes (Robinson, 2004):

- Understand the problem situation
- Determine modelling objectives
- Design the conceptual model (inputs, outputs, content)
- Collect and analyse the required data

A more concrete instruction was created by Furian et. a (2015), who took conceptual modelling as basic and refined it to create the Hierarchical Control Conceptual Model (HCCM) (see chapter 3.6).

3.5 **Conceptual Model (CM)**

A CM is a representation of a system. It helps to divide the real-world system into elements, which can then be defined by how they interact with each other. There are many different versions how to create a CM for abstract simulation models. More complex systems increased the interest in a method to structure simulation modelling, in the last years. (Robinson et al., 2016)

According to Furian et. al. (2015) one of the first who introduced such a model was Zeigler (1987). Furian et. al. (2015) explain, that Zeigler (1987) divided the system into the elements: the real system, the experimental frame, the base model, the lumped model, and the computer model.

Others such as Nance (1994) split systems into a representative model for communication, a conceptual model and the model that exists in the mind of the modeller (Furian et al., 2015). In a work from Robinson (2008) a conceptual model comes from an understanding of the situation, which is external to the conceptual model. Robinson (2008) also created the following definition:

‘‘The conceptual model is a non-software specific description of the computer simulation model (that will be, is or has been developed). It describes the objectives, inputs, outputs, content, assumptions and simplifications of the model’’ (Robinson, 2004, p. 65).

Before starting to program a model and creating the data base many considerations have to be made. The modeller has to be clear what has to be done, where it should lead and how it has to work in the very beginning phase of the project. It is useful to use methods, such as business process diagrams, flow charts, hierarchical diagrams or others, that help to define the single elements. Such elements are entities, activities, processes and rules of a system. If the assumptions at the beginning are too simple, unclear definitions can lead to heading into the wrong direction. It is also important to have the possibility to discuss main points of the project sufficiently transparent and clear with all stakeholders, in order to avoid any misinterpretation (see chapter 6.2.2 ”Limitations”). A method that provides this and can be used to structure the considerations at the very beginning is to create a CM. (Furian et al., 2015)

3.6 **Hierarchical Control Conceptual Modelling (HCCM)**

There has to be a clear distinction between CM and model design. Common widely used methods, such as flowcharts and business process diagrams have in common that they use queuing for their activities, which is an assumption that does not fit to every system. One of the latest works which considers this problem is from Furian et. al. (2015). They defined a Hierarchical Control Conceptual Modelling (HCCM) which *“[…] breaks with the assumption that all Discrete Event Simulation models are best represented by queuing systems”* (Furian et al., 2015, p. 87).
In HCCM it is not specified which of the various design methods are best to use. On the contrary, Furian et. al. (2015) write, that each developer has his preferred design elements and they describe a method that can be applied with different graphical representations. To define the individual elements, they use, among others, the definition of Arbez and Birta (2007) and revised it. *“An activity is an indivisible unit that characterizes an interaction among entities, is associated with a purposeful task and evolves over a nonzero (but usually finite) interval of time”* (Furian et al., 2015, p. 85). Furian et. al. (2015) also mentioned that the modeller has to find out under which circumstances the different behaviour occurs. It differs if they take place at fixed, defined points in time, or regarding to conditions or decisions made before, equal to the B- and C-events, from Robinson (2004) which were described in chapter 3.3.1.

Furian et. al. (2015) defined an 8 steps model for their HCCM framework (see below) based on Robinsons (2004) 4-phase model: understanding the problem; determine modelling objectives; design the conceptual model; collect and analyse the required data.

1. **Understanding the problem situation**
   This is the very first step in the CM process where the problems have to be figured out and stated. For this reason, it is necessary to use problem structuring methods and become clear about the initial condition. (Furian et al., 2015)

   The result of this step is an *"informal, textual description of the problem situation"* (Furian et al., 2015, p. 89).

2. **Identification of objectives**
   There exist two types of objectives, the general objectives and the modelling objectives. General objectives describe the requirements on the simulation tool, whilst modelling objectives bring up what can be achieved from the model's development and use. Both objectives have to be identified and declared in a form to structure them. (Furian et al., 2015)

3. **Outputs**
   Outputs are what the model delivers. Such outputs can be single variables or a set of data. By considering the model's output the modeller also has to define how to measure the output values (e.g. with key figures, factors, etc.). (Furian et al., 2015)

   This step delivers a table or graph with all outputs and their evaluation. (Furian et al., 2015)

4. **Input factors**
   A model requires various inputs to generate outputs. Some of the inputs can be experimental factors or values that may change later. Therefore, a clear distinction has to be made between fixed and variable inputs. Fixed inputs can be set as constant values that impact the system, but their values are not changeable. Variable inputs depend on the simulation scenario and can thus be changed by the simulation user. In this step all inputs have to be defined including their expected range plus a description how they are involved in the model. Inputs may also be policies, which have to be handled with a toolset for their integration. (Furian et al., 2015)

   The outcome of this step is a table with all inputs plus their possible range. (Furian et al., 2015)

5. **The model structure**

"The model structure is defined by the entity structures that are included" (Furian et al., 2015, p. 90). Entities can be differentiated into active and passive types. *"Active entities have a particular behaviour associated with them […] [whilst] […] passive entities are not associated with a behavioural flow [in the system (for example a computer that is used to model a 3D part)]"* (Furian et al., 2015, p. 90 words in square braket are added by author). All entities are assessed, and the modeller decides which are included in the model. (Furian et al., 2015)

This step can either be done with tables or UML class diagrams or an Entity Relationship Model (ERM). In advance the model's flow shall be illustrated by moving entities, this can be in a textual form but mostly it is more informal to draw a picture. (Furian et al., 2015)

6. **Individual model behaviour**

All entities have individual behaviours, which have to be identified and reported. It is in the modeller's responsibility to decide what behaviours are included and in which detail they are included in the model. In other words, the modeller has to define the scope of the model and to set borders. (Furian et al., 2015)

This step delivers a visual representation of the entity's flow through the system (e.g. with activity cycle diagram, Unified Modelling Language (UML), sequence diagram, flow chart or similar) plus a table with the definitions of all activities. (Furian et al., 2015)

7. **Systems behaviour**

This step of the framework is the major extension of Furian et. al. (2015), where the control structure of the system and its rules are declared. That happens via control units that must be defined. These control units (CU) form a hierarchical tree together with activities and events. The control unit builds the base of the tree where the decisions are made which activities or events are to be executed. In the later simulation the CU gives the decision which request (to start an event) and what entities are affected. Such a decision is based on rules, or a set of rules, that are given to the control units and so *"[…] determine the conditional behaviour of the model"* (Furian et al., 2015, p. 92).

A system can have more than one CU, but one control unit has to be the main CU (like division managers that are under control of the head of a company). Rules cannot only start activities, or force events, but may also change the state of the system, hold a set of attribute and state variables. *"The depth of the designed tree is a modelling choice, which should be made with great care. Too much granularity leads to unnecessarily complex models that are cumbersome to deal with. Whereas too few control structures also can lead to rising complexity as conditions, dispatching and interactions get harder to handle within each control unit"* (Furian et al., 2015, p. 92).

The advantage of these CUs is that dependencies of activities, entities and the systems state can be modelled much easier and the system is able to react on state changes more flexible. With classical queuing models only conditional events can request the next event. By the integration of CUs, it is possible to control the behaviour of the system, including entities that are resources for many tasks by requesting the next event dependent on different values. This means that interactions of entities and the system can be modelled better. (Furian et al., 2015)

The control unit and its rules have to be determined by the modeller asking himself the following questions (Furian et al., 2015, p. 13):

- *“What decisions are made?*
- *Where are decisions made and by whom?*
- *On what basis are decisions made (e.g. what state-variables)?*
- *How are decisions structured (e.g. overruled)?*
- *What simplifications and assumptions are made?”*

Answering the questions above shall help the modeller to create a tree structure of responsibilities with the main CU as root item. Furian et. al. (2015) suggests a graphical form to document this tree with the following guidelines (see Figure 3.8) (Furian et al., 2015):

- Control Units …drawn as… Rectangles with Cut Corners
- Requested Activities …drawn as… Normal Rectangles
- System Activities …drawn as… Rounded Rectangles

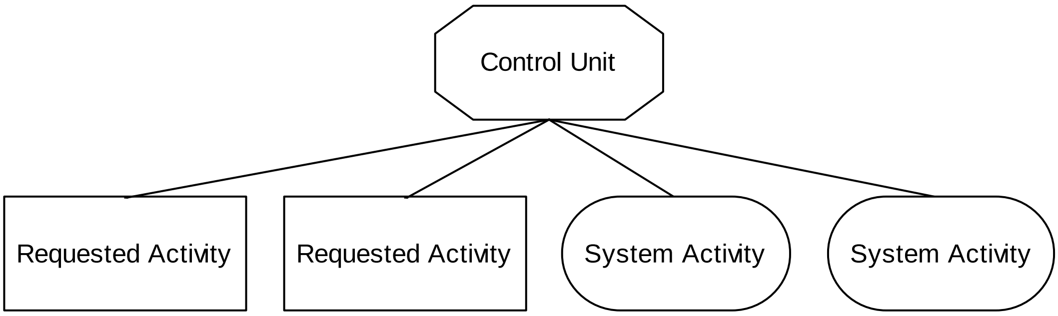

Figure 3.8 Graphical tree for control unit elements defined by (Furian et al., 2015)

A set of rules for the control units has to be declared, either in textual form, pseudo code or in a graphical form (e.g. a logical flow diagram with rule sets for control units). (Furian et al., 2015)

8. **Simplifications and assumptions**
During all the above 7 mentioned steps simplifications and assumptions have to be made. All of them have to be reported. This has to be done in a way that describes their impact on the system and what happens if one of them turns out to be wrong. (Furian et al., 2015)

3.7 **Entity Relationship Model (ERM)**

As explained in chapter 3.5 "Conceptual Model (CM)" a method to structure and visualize the entities and their relation has to be used. One of the easiest to understand and thus simple to create for complex systems is the Entity-Relationship Model (ERM). The ERM structures every system into entities, attributes and relationships in between them (see Figure 3.9 and Figure 3.10). (Furian et al., 2015)

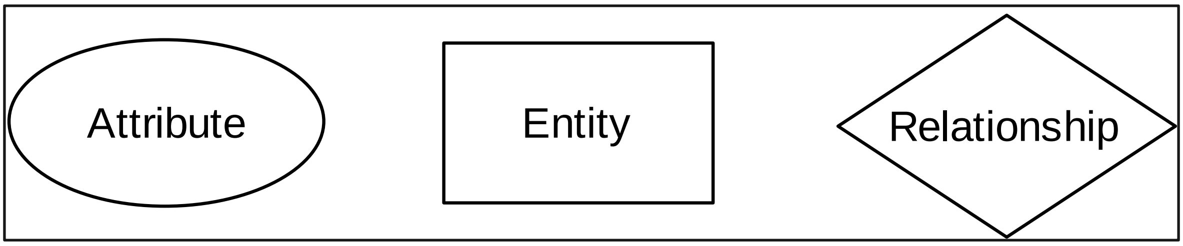

Figure 3.9 The three elements of ERM (illustration by author based on Chen, 1976)

The ERM was created in 1976 by Dr. Peter Pin-Shan Chen (1976) at the Massachusetts Institute of Technology and he revolutionized the data analysis with his model. Chen describes the entity-relationship model as *"[…] a framework from which the three existing data models may be derived. The reader may view the entity-relationship model as a generalization or extension of existing models"* (Chen, 1976, p. 10).

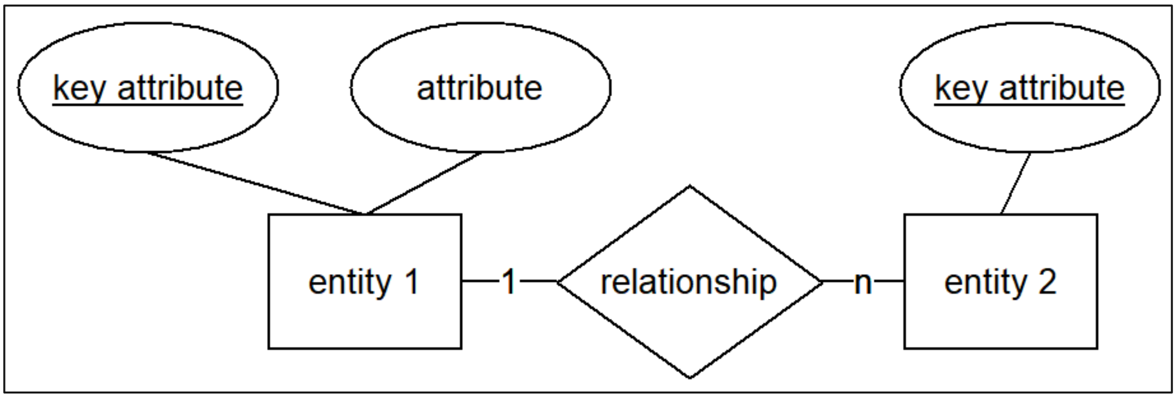

Figure 3.10 Example of a relationship between entities (illustration by author based on Chen, 1976)

The three elements of ERM can be explained as follows (Chen, 1976):

- Entities serve as basic elements and represent an individually identifiable object of reality (e.g. an order). Entities can have several attributes, but at least one must be assigned.
- Attributes are unique characteristics of the entities, which allow the entity to be subdivided, structured and distinguished. One attribute is defined as a so-called key attribute; it must be unique so that there is no possibility of confusion between different elements of the same entity. Such a key attribute could be, for example, the purchase order number, if the entity is an order.
- Last but not least there are relationships between entities. They connect two entities and set them in correlation. The relationship is extended with cardinalities (1:1, 1:n, n:m) to show how many items of an entity refer or access an item of the other entity. Possible values and their meaning are shown in Table 3.1.

Cardinality	Explanation	Example
1:1	exactly one element of this entity is related to one other	price of a product is related to the product
1:n	one element of this entity is related to any others	one order includes various products
n:m	any number of elements of this entity is related to any number of elements of the second entity	many products can contain any number of different materials

Table 3.1 Cardinalities of ERM (created by author based on Chen, 1976)

4 Total Costs of Ownership (TCO)

Not only for every project, also for every investment and running process it is necessary to know its costs. The knowledge of costs incurred is important for economic and efficient planning on one hand, but also to be able to make statements regarding optimisation and saving potentials of processes. Decisions regarding an investment or a project are increasingly made at the upper management levels of a company on the basis of calculations and profitability figures. Not least because of this, an exact breakdown of costs is essential. (Wild & Herges, 2000)

Since in the case of this thesis a new concept was planned and implemented, which should be economically feasible and vital over several years, it was even more important to include costs as a planning factor. Therefore, a method had to be used which ensures that all costs are included. Costs detecting methods are particularly required in systems that are influenced by information technologies (IT), as it is the case in this project, because of the high usage of 3D modelling software, 3D slicing software and 3D printer firmware. In order to determine all expenses, a systematic approach is required to avoid incompleteness and wrong estimations. (Redman et al., 1998)

There are many models created by IT analysts and specialists to determine the amount of expenses, but these are mostly custom-made solutions, which are also intended to benefit the issuing company and are therefore not or only partially published. As a result, in various documents you will find either very vague descriptions of the methods, or very extensive and detailed procedures, which are limited to individual applications. Often the individual models differ in how the cost factors are calculated and how strongly they are weighted. As a common factor, however, all of them have a cost structure that is as transparent and as accurate as possible. (Redman et al., 1998; Wild & Herges, 2000)

One of the most comprehensive methods, which is nevertheless described generally enough to be applicable in many ways, is the Total Costs of Ownership (TCO) developed by the Gartner Group Inc. (2020) in 1987. The Gartner Group Inc. (Gartner Group Inc., 2020) defined a method with extensive analysis of costs. After its publication, this method initially received little attention, but in recent years it has gained more and more importance with the increasing interconnectedness of systems. The more complex a system is, the easier it gets to overlook costs, as they cannot be clearly assigned to one position. In order not to lose sight of the costs, it is advisable to categorize the costs according to factors and to break them down subsequently. (Gartner Group Inc., 2020)

Many different Companies designed their own cost models (e.g. Real Costs of Ownership developed by Meta Group Consulting (2020)). Wild & Herges (2000) compared different cost provision models and figured out, that all of them head for the same target, the provision of transparent and realistic cost structures. With this comparison Wild & Herges (2000) came to the insight, that it is possible reduce the costs to 4 basic factors with the same share for all models.

These basic factors are (Wild & Herges, 2000):

- 50% - Performing original tasks of an IT department by end users (e.g. Peer-to-Peer-Support, data management)
- 20% - Assets in IT infrastructure components (Hard- and Software)
- 17% - Tech support (e.g. administration for the system)
- 13% - IT-related administration (e.g. management of the IT department, training activities)

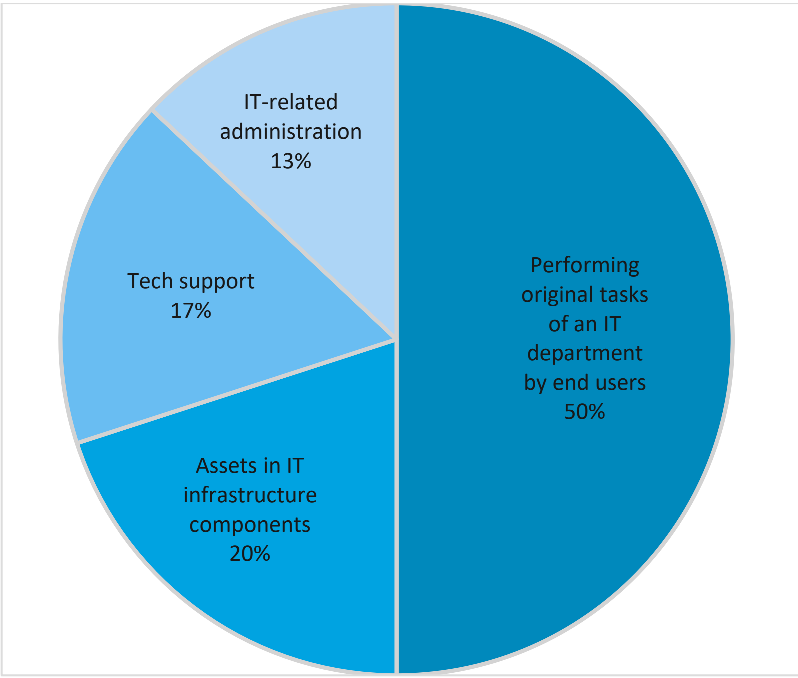

Diagram 4.1 Share of basic factors (illustration by author based on Wild and Herges, 2000)

The Gartner Group Inc. (2020) defined the TCO model by splitting the costs into direct and indirect costs, which are further separated into subgroups. The classification is described below.

4.1 Direct Costs

These are expenses for provision of services, depreciation of investments, leasing fees as well as wages and salaries. Probably an easier way to imagine, direct costs are every cost, where you have bills, payrolls and other documents. (Wild & Herges, 2000)

- Hard- and Software
  This category includes expenses for the procurement of goods and software required for processes and operations (such as machine parts, printer materials, software licenses, etc.) (Wild & Herges, 2000).
- Operations
  This group includes all costs for the payment of staff and employees directly involved in IT-infrastructure, so called computer science professions (e.g. technical service, process management, data controller) (Wild & Herges, 2000).
- Administration
  Here are all salaries and wages summed up, as well as expenditures for financial and administration tasks (e.g. budgeting, contract administration) (Wild & Herges, 2000).

4.2 Indirect Costs

Indirect costs represent a loss of value caused by processes that reduce the efficiency of a system or process. These costs can be forced by planed events or due to unplanned incidents. This could be a peer-to-peer-support for your computer system or also a downtime caused by a machine failure. Mostly they are hard to quantify and measure and therefore missed out. Not even The Gartner Group Inc. (2020) provides a method to measure these costs reliably in its published policies. (Wild & Herges, 2000)

- End-User-Operations
  This refers to costs that represent a loss of value, resulting from time an end-user requires for training measures and tasks that are originally scheduled to be done by the IT department. These costs can be caused, for example, by a service provided by the IT department that does not satisfy the needs of the users and thus takes up time from the user. The user could normally use this time for productive activities, and it therefore represents additional costs (e.g. self- and peer-to-peer-support, creating backups, adapting a software, using resources for private purpose). (Wild & Herges, 2000)

- <u>Downtime</u>
  This takes into account the loss of value and the loss due to the failure of parts of the system (loss of income due to the standstill of necessary process parts) (Wild & Herges, 2000).

Figure 4.1 gives a simplified overview of the described cost structure, a detailed structure of the TCO model V4.0, as it is called, by the Gartner Group Inc. (2020) is attached in chapter 9-C “Attachments”.

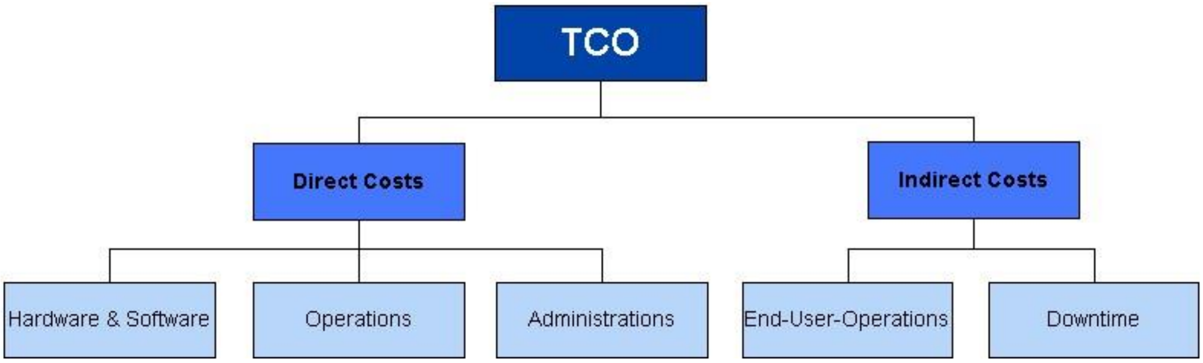

Figure 4.1 TCO-model 4.0 (illustrated by author based on Gartner Group Inc., 2020)

At this point it should be mentioned that the implementation of general TCO strategies leads not only to financial savings but can also bring general improvements in the processes. This is based on the pure fact of repetitive process reviews. In companies and ongoing projects, this requires the integration of all persons and instances involved. However, since a completely new system was to be modelled and set up in the present project, no ongoing operations had to be considered. The entire research project, though, was developed by several people and in several places at the same time, so information on progress and individual cost factors had to be coordinated. (Wild & Herges, 2000)

Critics on the TCO method can be that there is not enough flexibility in the analysis to represent complex and dynamic changing systems. The full extent of this system is illustrated in the figure which is attached in chapter C, while figure 4.1 is a simplyfied version of it. Another aspect to point on is, that the TCO analysis offers no possibility to show the development of values due to process sequences. A method, which takes this into account, is the ROI (return on investment). However, ROI has the problem that it does not include hidden cost factors, which results in the condition that the system has to be fully designed beforehand. (Wild & Herges, 2000)

5 Practical Application and Implementation

At the beginning of the project, a conceptual model was created to analyse and subsequently map the system at hand. It was important to consider all factors and, if informations were missing, assumptions had to be made. These assumptions have been documented in chapter 5.8.

Hay, Valentin and Bijlsma (2006) mentioned that *"[…] in the health care sector, a large proportion of resources are human beings. As humans will often perform a variety of tasks and can be assigned to tasks in many different ways, capturing system behaviour often requires more complex dispatching (of resources – e.g., humans – to entities/consumers – e.g., tasks) and control policies (e.g. staff workload balancing) […] ".* For this reason and also for the fact that activities in this system will not queue at every step in time, the HCCM framework (Furian et al., 2015), explained in chapter 3.6 "Hierarchical Control Conceptual Modelling (HCCM)", was used for this project. In a first iteration step the HCCM framework (Furian et al., 2015) was applied on the process of printing a PEEK implant, as this is the currently best known process in the 3D-PC. Thus, a basic structure of the system was created based on this process. Based on this it was possible to extend the model with more detailed steps. The beginning process only consisted of the basic activities (e.g. "check PrintOrder", "create PrintJob") without going too much into detail how these steps are performed. With this depth of abstraction, it was easier to focus on the main structure of the model. In a second step the activities where split into many activities, which were first summed up into one. (e.g. "check PrintOrder for Post-Treatment", "check Images of PrintOrder"). Constantly using the previous described framework, it was possible to evolve the model without losing the focus on the main structure.

5.1 Understanding the Problem Situation

The initial situation of this project was that a frame research project for the implementation of a 3D-PC was already existing at the LKH Graz. However, it was unclear to which number of technicians, computers and employees the 3D-PC will be built. Since the design of the entire system depends on several factors, a model was to be created which would <u>reliably determine the capacity utilisation of the 3D-PC on the basis of the expected projects</u>. With the figures supplied, the aim was to establish a 3D-PC that was economically and technically feasible and at the same time cost-efficient.

5.2 Identification of Objectives

The previous defined target (see chapter 5.1) was used to clarify the system's objectives. As described in chapter 3.6 "Hierarchical Control Conceptual Modelling (HCCM)" these objectives were divided into general and modelling objectives (see Table 5.1).

General Objectives	**Modelling Objectives**
requirements on simulation tool	*what can be achieved with the model*
finished orders over time	required number of technical employees
time from order to delivered part	required number of printers
current status of orders after certain amount of time	required number of computers
	TCO (total costs of ownership)

Table 5.1 Objectives of the required simulation tool

5.3 Outputs

The required outputs, to fulfil the defined objectives are shown in Table 5.2. They are the minimal outputs of the simulation model and they should be defined at the beginning in order to avoid later changes which may result in immense work.

workload of printers	
definition	ratio of time where printers where in use to overall time
unit	%
note	expected value: 0 – 100 limit value: 100

printer downtime	
definition	time when printers where not available due to maintenance and breakdowns
unit	h
note	-

production costs per order	
definition	total costs for processing an order
unit	€
note	-

required material	
definition	amount of required material for processed orders
unit	#
note	-

material costs	
definition	costs for used material
unit	€
note	-

handled orders	
definition	number of handled orders
unit	#
note	-

repair costs for printers	
definition	total cost for repairing and maintaining printers
unit	€
note	-

used inputs	
definition	used inputs for the simulation
unit	-
note	-

overall working time	
definition	overall productive time
unit	h
note	-

Table 5.2 Defined outputs of the CM for the 3D-PC

5.4 Input Factors

Just like the outputs, the inputs also had to be carefully considered in advance (see Table 5.3), as later changes may be hard to implement in the software.

pre-simulation time	
definition	time to start the simulation before set time
unit	hh:mm
range	06:00 - 07:00

working shift start	
definition	time when employees start working
unit	hh:mm
range	06:00 - 07:00

working shift end	
definition	time when employees end their work
unit	hh:mm
range	15:00 – 16:00

number of technicians	
definition	number of overall technical employees
unit	#
range	-

salary technicians	
definition	salary of a technical employee per hour
unit	€/h
range	--,--

number of radiologists	
definition	number of overall radiologists
unit	#
range	-

salary radiologist	
definition	salary of a radiologist per hour
unit	€/h
range	--,--

amount computers	
definition	number of overall computers in the 3D-PC
unit	#
range	-

amount working benches	
definition	number of overall working benches (e.g. for post-treatment, checking etc.)
unit	#
range	-

manufacturing overheads	
definition	overhead costs for manufacturing activities
unit	%
range	0 - 100

material overheads	
definition	overhead costs for materials
unit	%
range	0 - 100

fixed order costs	
definition	fixed costs for placing an order
unit	€
range	0 - 100

transportation costs	
definition	costs for transporting an item to another place in hospital
unit	€
range	-

time for placing order	
definition	hour of day when an order is placed
unit	h
range	0 - 24

time when orders arrive	
definition	hour of day when ordered material arrives at the 3D-PC
unit	h
range	0 - 24

order size	
definition	number of units that are ordered when minimal amount on stock is reached
unit	#
range	-

avg. printer repair time	
definition	average time needed to repair a printer
unit	min
range	-

reject part after repair	
definition	define if a part can be used if a printer breakdown occurs while printing
unit	-
range	yes / no

time between orders	
definition	average time between arrivals of new orders
unit	min
range	-

probability false order	
definition	set the probability that an order is not fully defined by customer or not feasible
unit	%
range	0 - 100

probability type implant	
definition	sets the probability that an arrived order requires an implant
unit	%
range	0 - 100

probability type model	
definition	sets the probability that an arrived order requires an anatomic model
unit	%
range	0 - 100

probability urgency low	
definition	sets the probability that an arrived order has low urgency
unit	%
range	0 - 100

probability urgency high	
definition	sets the probability that an arrived order has high urgency
unit	%
range	0 - 100

probability post-treatment	
definition	sets the probability that a part of an arrived order needs post-treatment
unit	%
range	0 - 100

probability interrupt	
definition	sets the probability that the printing process of a part can be interrupted (only if not highly urgent)
unit	%
range	0 - 100

avg. volume implant	
definition	defines the average volume of a part of type implant
unit	g
range	-

avg. volume model	
definition	defines the average volume of a part of type anatomic model
unit	g
range	-

avg. volume tool	
definition	defines the average volume of a part of type tool
unit	g
range	-

probability part fault	
definition	defines the probability that a printed part is faulty
unit	%
range	0 - 100

duration check order	
definition	sets the duration of the activity check a new order
unit	min
range	-

duration consult customer	
definition	sets the duration of the activity consult a customer after a failed order check
unit	min
range	-

duration segmenting	
definition	sets the duration of the activity segmenting an image
unit	min
range	-

duration modelling	
definition	sets the duration of the activity modelling a computer model
unit	min
range	-

duration creating print job	
definition	sets the duration of the activity creating a print job from a computer model
unit	min
range	-

duration prepare printer	
definition	sets the duration of the activity preparing a printer to print
unit	min
range	-

duration printing time per weight	
definition	sets the duration for printing 1 gram of a material
unit	min/g
range	-

duration clean part	
definition	sets the duration of the activity cleaning a part directly after printing
unit	min
range	-

duration check part	
definition	sets the duration for checking a produced part
unit	min
range	-

duration post-treatment	
definition	sets the duration of the activity post-treatment
unit	min
range	-

duration sterilization	
definition	sets the duration for sterilizing a part
unit	min
range	-

duration pack part	
definition	sets the duration of the activity packing a sterilized part
unit	min
range	-

duration send part	
definition	sets the duration for sending a finished part to another place in hospital
unit	min
range	-

Table 5.3 Defined inputs of the CM for the 3D-PC

5.5 **Model Structure**

The structure of the 3D-PC is illustrated in Figure 5.1 with an ERM according to Chen (1976) as described in chapter 3.7 and shows the relationships of the 18 entities.

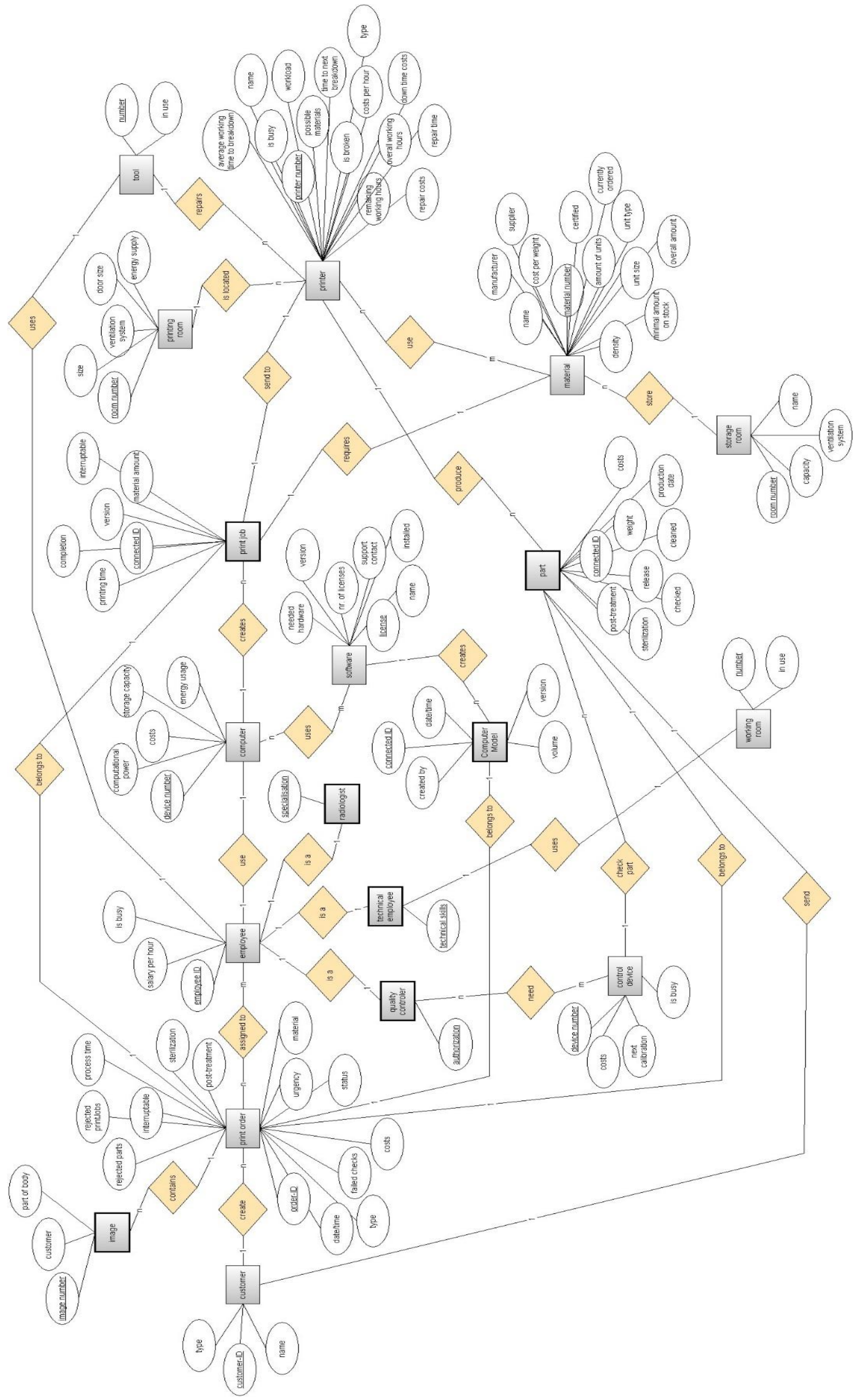

Figure 5.1 Entity Relationship Model of the 3D-PC

Corresponding to the ERM in Figure 5.1 all entities with their attributes are listed in Table 5.4 for a better overview.

Print Order	
Type	active
parent	-
order-ID	
date/time	
customer	
associated images	
status	
interruptible	
type	
urgency	
material	
failed checks	
post-treatment	
sterilization	
rejected parts	
rejected printJobs	
costs	
process time	

Print Job	
Type	active
parent	Print Order
connected ID	
material	
material amount	
printing time	
version	
interruptible	
completion	

Computer Model	
Type	active
parent	Print Order
connected ID	
created by	
date/time	
version	
volume	

Part	
Type	active
parent	Print Order
connected ID	
production date	
cleaned	
weight	
release	
post-treatment	
sterilization	
checked	
costs	

Image	
Type	active
parent	Print Order
customer	
part of body	

Tool	
Type	active
parent	-
number	
in use	

Customer	
Type	passive
parent	-
customer-ID	
type	
name	

Employee	
Type	active
parent	-
employee-ID	
is busy	
salary per hour	

Radiologist	
Type	active
parent	Employee
employee-ID	
specialization	

Technician	
Type	active
parent	Employee
employee-ID	
authorization	

Printer	
Type	passive
parent	-
printer number	
name	
possible material	
avg. working time to breakdown	
is busy	
is broken	
remaining working hours	
overall working hour	
repair costs	
type	
time to next breakdown	
costs per hour	
workload	
repair time	
down time costs	

Material	
Type	active
parent	-
material number	
name	
amount of units	
unit size	
unit type	
stock limit	
cost per weight	
density	
manufacturer	
supplier	
certified	
overall amount	
unit type	
currently ordered	

Storage Room	
Type	passive
parent	-
room number	
name	
capacity	
ventilation system	

Computer	
Type	passive
parent	-
device number	
is busy	
software	

Control Device	
Type	active
parent	-
device number	
is busy	
costs	
next calibration	

Working Room	
Type	passive
parent	-
room number	
in use	

Table 5.4 System Entities of the CM of the 3D-PC

To visualize the process, Figure 5.2 displays the flow of active entities trough the 3D-PC.

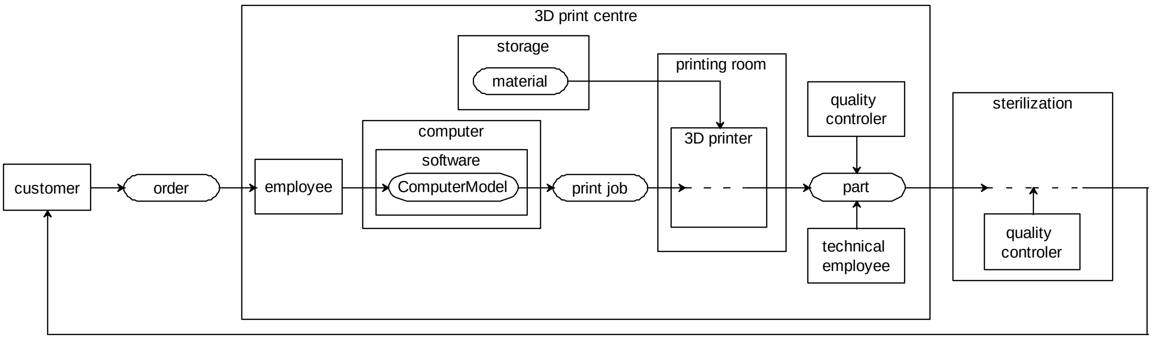

Figure 5.2 Flow of active entities

5.6 Individual Model Behaviour

According to the guidelines, described in chapter 3.6, the following flow charts (see Figure 5.3, Figure 5.4, Figure 5.5, Figure 5.6, Figure 5.7 and Figure 5.8) show the handling of print orders and thus the behaviour of ordered parts and used material for printed parts.

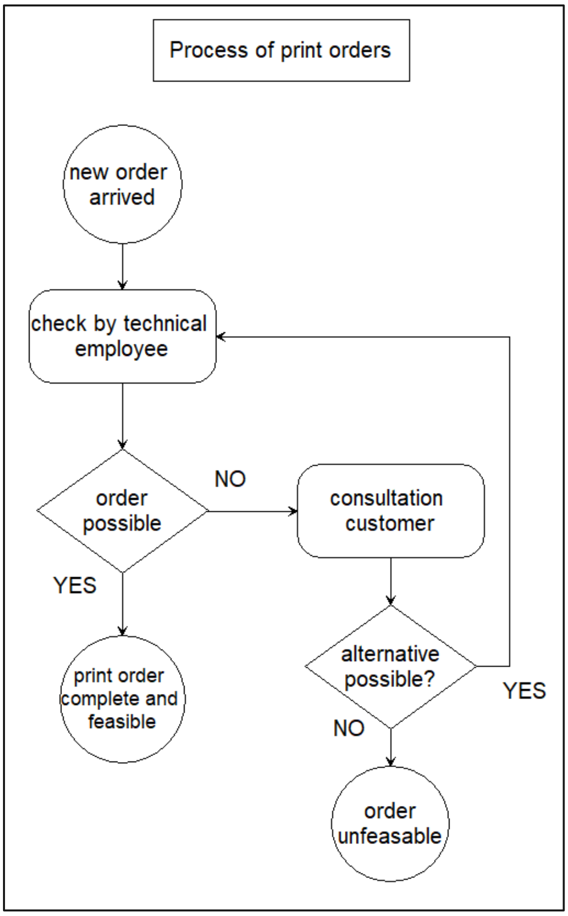

Figure 5.3 Process of print orders

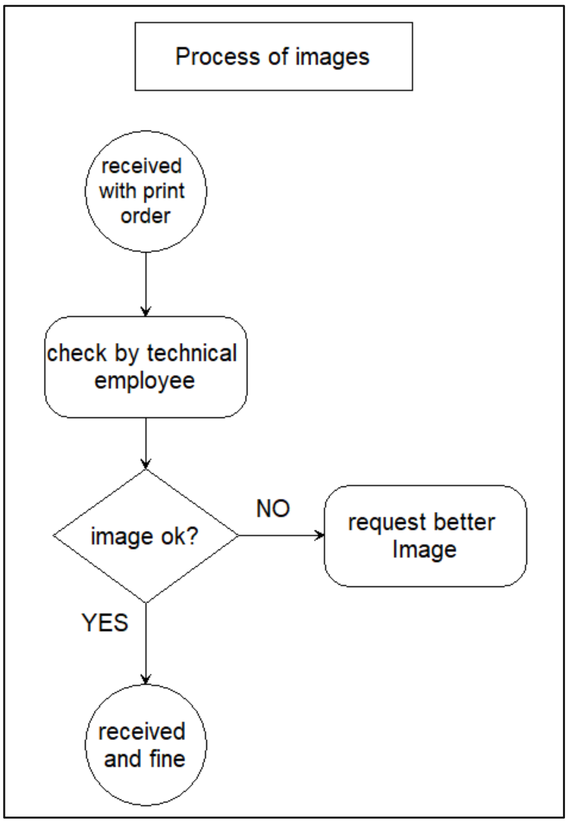

Figure 5.4 Process of images

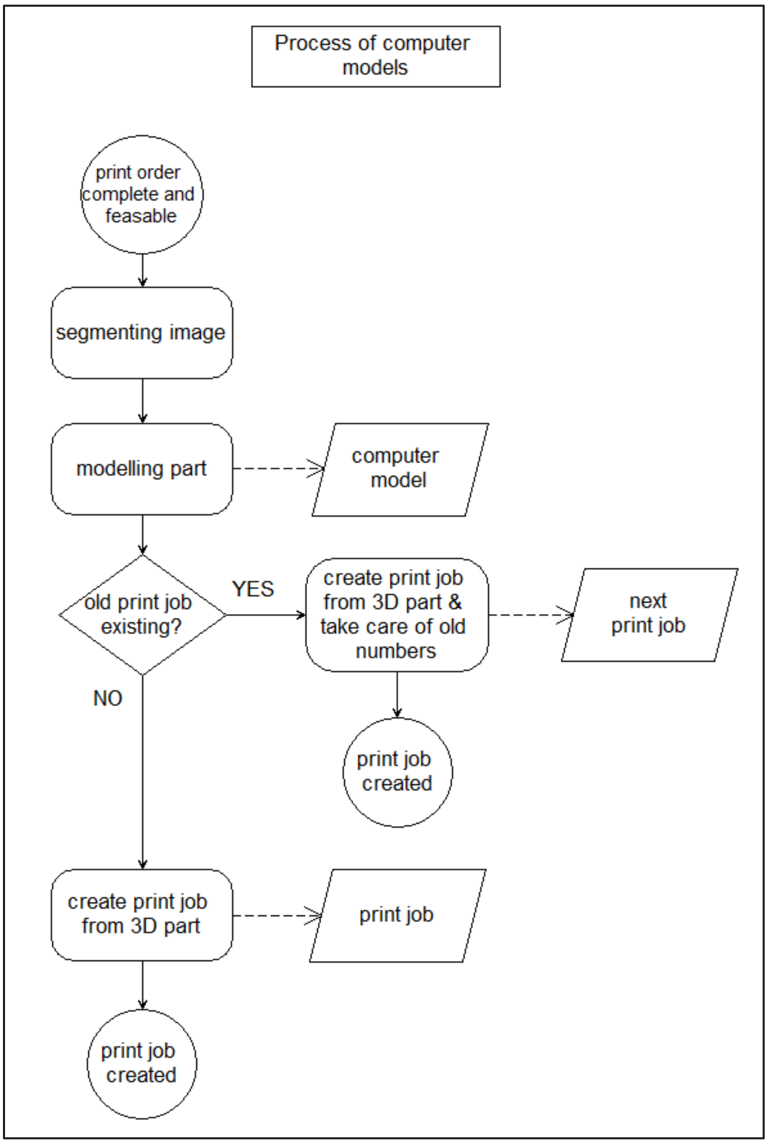

Figure 5.5 Process of computer models

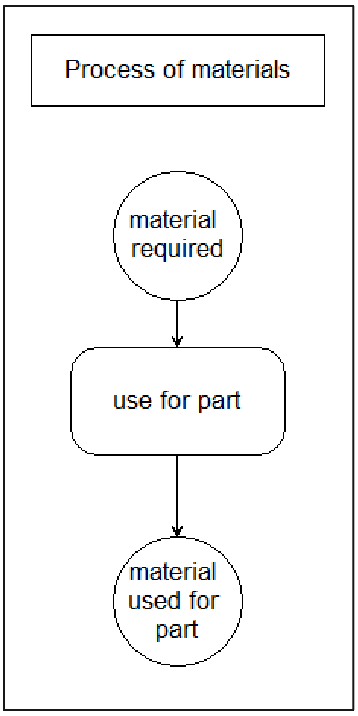

Figure 5.6 Process of materials

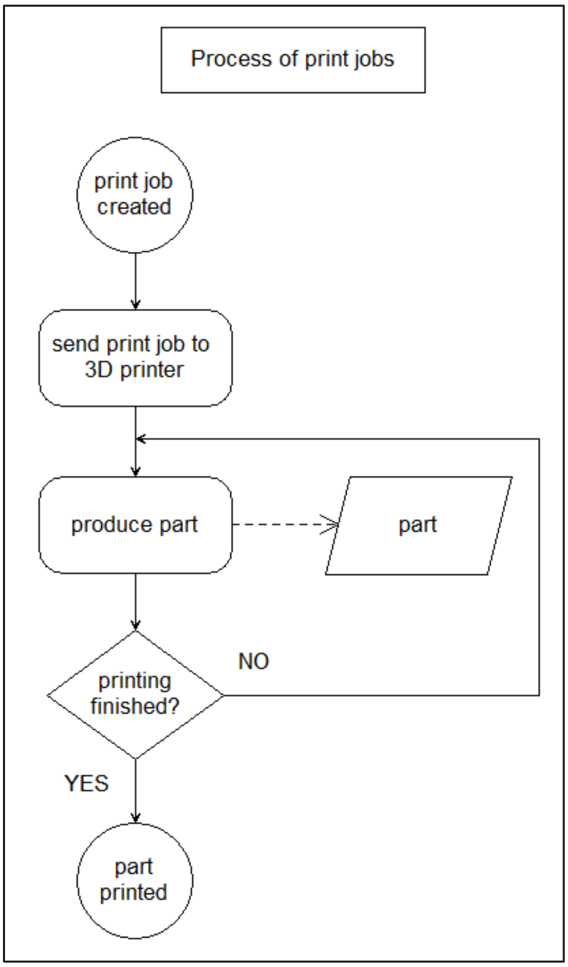

Figure 5.7 Process of print jobs

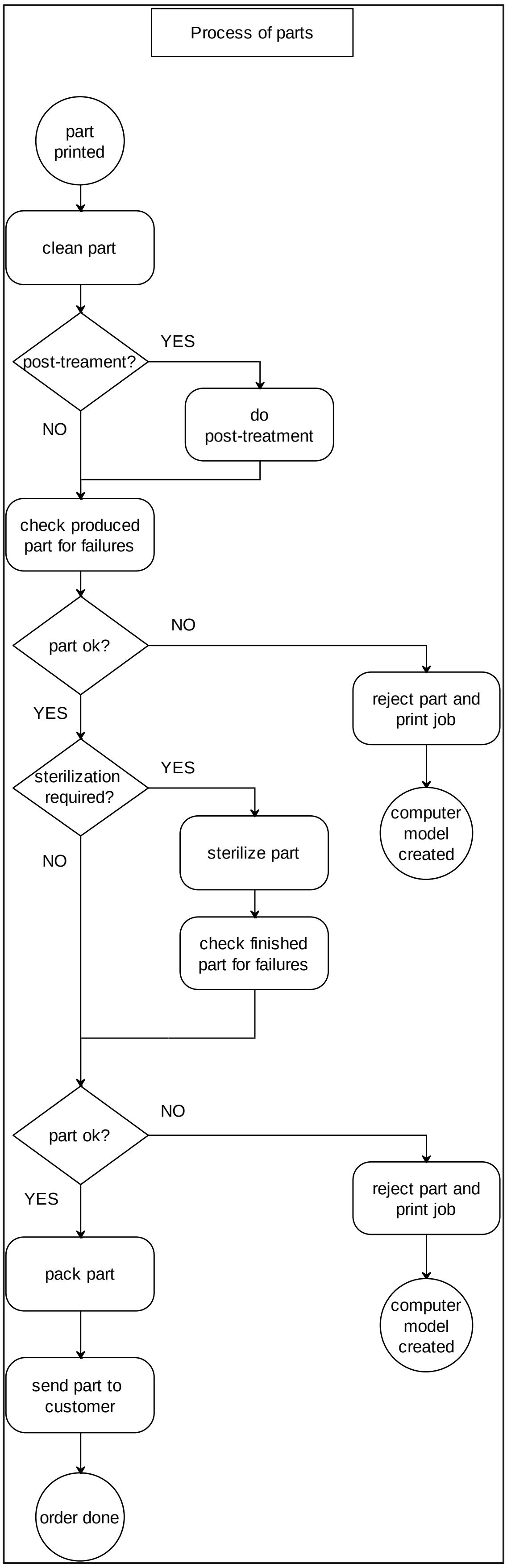

Figure 5.8 Process of parts

As described by Furian et. al. (2015) all activities within the flow charts (see Figure 5.3, Figure 5.4, Figure 5.5, Figure 5.6, Figure 5.7 and Figure 5.8) are additionally described (see Table 5.5).

Order Check	
Participating entities	*PrintOrder, Computer, Employee*
Start type	*triggered*
End type	*scheduled*
Start state changes	*Employee.IsBusy = true* *Computer.IsBusy = true*
End state changes	*order ok / check failed* *Employee.IsBusy = false* *Computer.IsBusy = false* *add costs to Order*
Attributes	*-*
Duration	*15 min*
Request attributes	*order urgency, order checkfailed*
Request specification	*GetOrderChecked*

Consultation Customer	
Participating entities	*PrintOrder, Computer, Employee, Customer*
Start type	*sequential*
End type	*scheduled*
Start state changes	
End state changes	*Employee.IsBusy = false* *Computer.IsBusy = false* *add costs to Order*
Attributes	*order checkfailed*
Duration	*1 min*
Request attributes	*-*
Request specification	*-*

Segmenting Image	
Participating entities	*PrintOrder Employee, Computer*
Start type	*triggered*
End type	*scheduled*
Start state changes	*Employee.IsBusy = true* *Computer.IsBusy = true*
End state changes	*Employee.IsBusy = false* *Computer.IsBusy = false* *add costs to Order*
Attributes	*-*
Duration	*20 min*
Request attributes	*PrintOrder*
Request specification	*GetImageSegmented*

Model Part	
Participating entities	*PrintOrder Employee, Computer*
Start type	*triggered*
End type	*scheduled*
Start state changes	*Employee.IsBusy = true* *Computer.IsBusy = true*
End state changes	*create ComputerModel* *Employee.IsBusy = false* *Computer.IsBusy = false* *add costs to Order*
Attributes	*ComputerModel*
Duration	*30 min*
Request attributes	*PrintOrder*
Request specification	*GetPartModelled*

Creating PrintJob	
Participating entities	*PrintOrder Employee, Computer*
Start type	*triggered*
End type	*scheduled*
Start state changes	*Employee.IsBusy = true* *Computer.IsBusy = true*
End state changes	*create PrintJob* *Employee.IsBusy = false* *Computer.IsBusy = false* *add costs to Order*
Attributes	*PrintJob*
Duration	*15 min*
Request attributes	*PrintOrder*
Request specification	*GetPrintJobCreated*

Prepare Printer	
Participating entities	*PrintOrder Employee, Computer, Printer*
Start type	*triggered*
End type	*scheduled*
Start state changes	*Employee.IsBusy = true* *Computer.IsBusy = true* *Printer.IsBusy = true*
End state changes	*Computer.IsBusy = false* *add costs to Order*
Attributes	*interruptable*
Duration	*1 min*
Request attributes	*PrintOrder*
Request specification	*GetPrintJobSended ToPrinter*

Produce Part	
Participating entities	*PrintOrder, Employee, Printer, MaterialStorage*
Start type	*scheduled*
End type	*scheduled*
Start state changes	*Employee.IsBusy = false*
End state changes	*set Part.completion* *create part* *subtract used Material* *Printer.IsBusy = false* *add costs to Order & Part*
Attributes	*interruptable, completion*
Duration	*1,2 min/g*
Request attributes	*-*
Request specification	*-*

Clean Part	
Participating entities	*PrintOrder, Employee*
Start type	*triggered*
End type	*scheduled*
Start state changes	*Employee.IsBusy = true*
End state changes	*Part.cleaned* *Employe* *add costs to Order* *e.IsBusy = false*
Attributes	*-*
Duration	*5 min*
Request attributes	*PrintJob*
Request specification	*GetPartCleaned*

Check Part	
Participating entities	*PrintOrder, Employee*
Start type	*triggered*
End type	*scheduled*
Start state changes	*Employee.IsBusy = true*
End state changes	*Part.cleaned* *Employee.IsBusy = false* *add costs to Order*
Attributes	*-*
Duration	*15 min*
Request attributes	*PrintJob*
Request specification	*GetPartChecked*

Do PostTreament	
Participating entities	*PrintOrder, Employee*
Start type	*triggered*
End type	*scheduled*
Start state changes	*Employee.IsBusy = true*
End state changes	*Part.postTreatment* *Employee.IsBusy = false* *add costs to Order*
Attributes	*-*
Duration	*20 min*
Request attributes	*PrintJob*
Request specification	*GetPartPostTreated*

Sterilize Part	
Participating entities	*PrintOrder, Employee*
Start type	*triggered*
End type	*scheduled*
Start state changes	*Employee.IsBusy = true*
End state changes	*Part.Sterilization* *Employee.IsBusy = false* *add costs to Order*
Attributes	*-*
Duration	*10 min*
Request attributes	*PrintJob*
Request specification	*GetPartSterilized*

Pack Part	
Participating entities	*PrintOrder, Employee*
Start type	*triggered*
End type	*scheduled*
Start state changes	*Employee.IsBusy = false*
End state changes	*Part.release* *add costs to Order*
Attributes	*-*
Duration	*5 min*
Request attributes	*PrintJob*
Request specification	*GetPartPacked*

Send Part to Customer	
Participating entities	*PrintOrder, Employee*
Start type	*sequential*
End type	*scheduled*
Start state changes	*-*
End state changes	*Employee.IsBusy = false* *add costs to Order*
Attributes	*PrintOrder.Status*
Duration	*2 min*
Request attributes	*-*
Request specification	*-*

Order Material	
Participating entities	*Material*
Start type	*triggered*
End type	*scheduled*
Start state changes	*MaterialOrdered = true*
End state changes	*add new material to inventory* *calculate OrderCosts* *MaterialOrdered = false*
Attributes	*OrderCosts*
Duration	*till delivery time*
Request attributes	*Material*
Request specification	*GetMaterialOrdered*

Repair Printer	
Participating entities	*Employee, Printer*
Start type	*triggered*
End type	*scheduled*
Start state changes	*Employee.IsBusy = true*
End state changes	*Printer.IsBroken = false* *Employee.IsBusy = false* *calculate RepairCosts*
Attributes	*RepairCosts*
Duration	*90 min*
Request attributes	*Printer*
Request specification	*GetPrinterRepaired*

Table 5.5 Definition of processes of the individual model behaviour of the 3D-PC

5.7 System Behaviour

The system behaviour can be described by summing up the major decisions that have to be made and by whom. Furian et. al. (2015) suggested to create a table with all decisions, that have to be made and on which attributes the regarding question has to be decided (see Table 5.6). This table also includes the simplifications and assumptions, that are necessary to program the rules for the simulation.

Question/ Decision	*Decision made by (responsible)*	*Decision based on*	*Simplification / Assumption*
accept new orders	*print centre control*	*daytime, shift hours*	*no weekends considered*
set orders in row	*print centre control*	*urgency, order ID*	*urgency of order is defined as low, normal or high*
interrupt printing	*print centre control*	*is order interruptible*	
order new material	*print centre control*	*limit amount, amount on stock*	*know how much material is used for production* *+ have exact live data about material on inventory.*

Table 5.6 Decision rules for the system behaviour of the 3D-PC

5.7.1 Control Policies

Based on the decision rules, described in Table 5.6, the control policies can be defined and are illustrated in Figure 5.9.

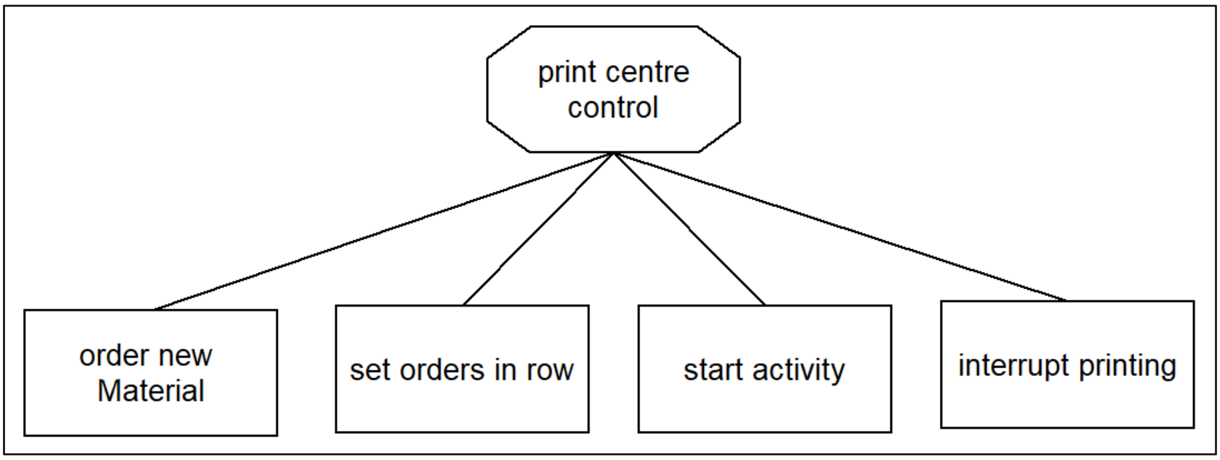

Figure 5.9 Control policies for the 3D-PC

The detailed process for making decisions by the control units are displayed in the following flow charts (see Figure 5.10, Figure 5.11, Figure 5.12 and Figure 5.13).

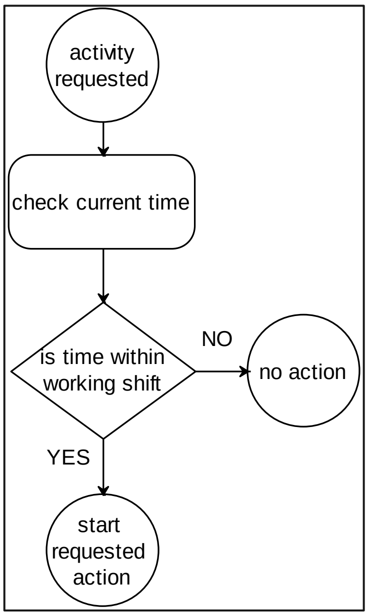

Figure 5.10 Flow chart decision working time

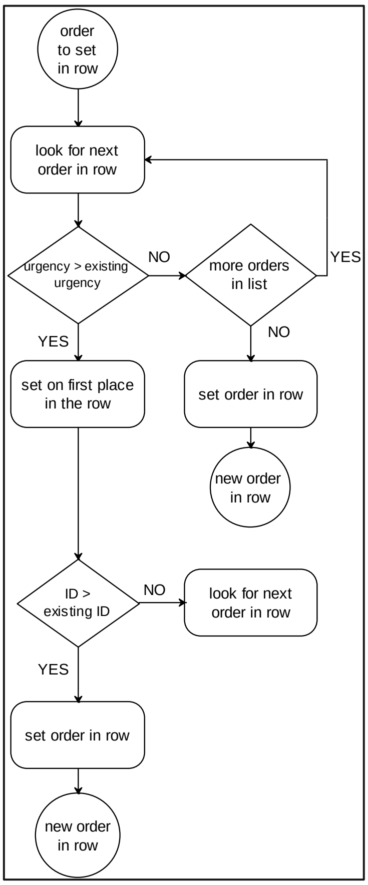

Figure 5.11 Flow chart row orders

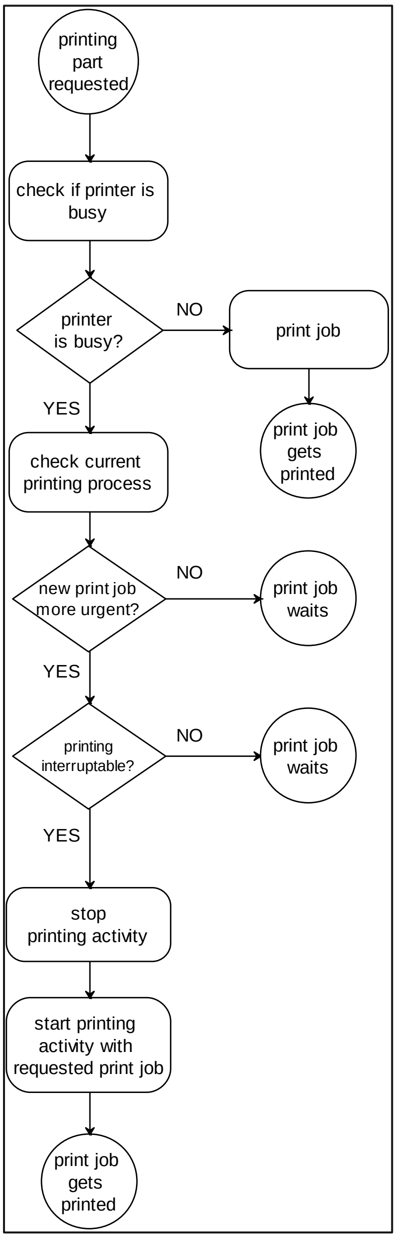

Figure 5.12 Flow chart decision interrupt printing

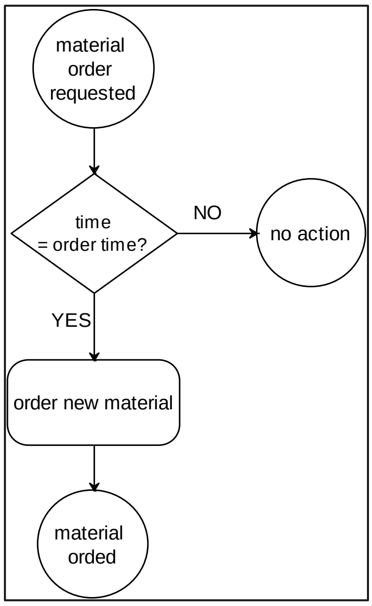

Figure 5.13 Flow chart decision order new material

5.8 Simplifications and Assumptions

Through the modelling process of the system it was necessary to make simplifications and meet a few assumptions, otherwise the effort in programming would have increased unproportionally. These simplifications and assumptions are structured in the part of the system they affect and are stated below:

Works preparation

- All technicians have the same qualification.
- All radiologists have the same qualification.
- Breaks (e.g. lunch break) are not included in the simulation. This assumption is acceptable, since individual employees automatically take breaks throughout the day without interfering the sequences (e.g. while printing parts).
- Orders arrive between 7am and 3pm, if a created order would arrive at any other time it is rescheduled to 7am the next day.

Printer occupancy

- The part size is defined by mean value per type and created with a Gaussian distribution (every amount has the same probability).
- Orders with all combinations (type, urgency, material etc.) are possible
- The completion of a printing process is calculated and documented in full percentages. Decimal numbers are rounded down, this means that the calculated printing time and material amount can be slightly higher. However, since in reality small amounts are used at the beginning and end of printing to set the printer and these are not included in the simulation, the assumption is legitimate.

Printer availability

- The machine breakdowns occur based on a Poisson distribution.
- Printers can be repaired by the own technical staff, therefore no waiting times for external support are considered. The technical employee must be well acquainted with the machine and is not available for handling orders during a machine repair.
- Weekends are considered as working time. The shift time per day (start of shift and end of shift) is adjustable via input values.
- The amount of material on stock is known at every point in time.
- Planned maintenance activities are not specifically implemented in the model. To compensate this, the frequency of breakdowns was increased appropriately and the duration of repairs was adjusted.

5.9 Software Implementation

The simulation tool is coded in C# based on a software library called HCDESlib published on GitHub.com by Nikolaus Furian (2016). This library is built on the structure of HCCM (Furian et al., 2015) (see chapter 3) and can be used to program a discrete event simulation tool. Since the structure of HCCM was already used for the planning of the program, it was the logical consequence to use the corresponding library.

5.9.1 User Interface

The software library already implemented a basic graphic interface, to run the simulation and display state changes of events, thus it had to be adapted to the project's needs. This included the creation of input windows, implementing the possibility to save and restore settings from a file, to edit settings directly in the program and programming the presentation of entities (e.g. orders, employees, workplaces, etc.). The user interface was customized to the design of the CAMed project (see Figure 5.14).

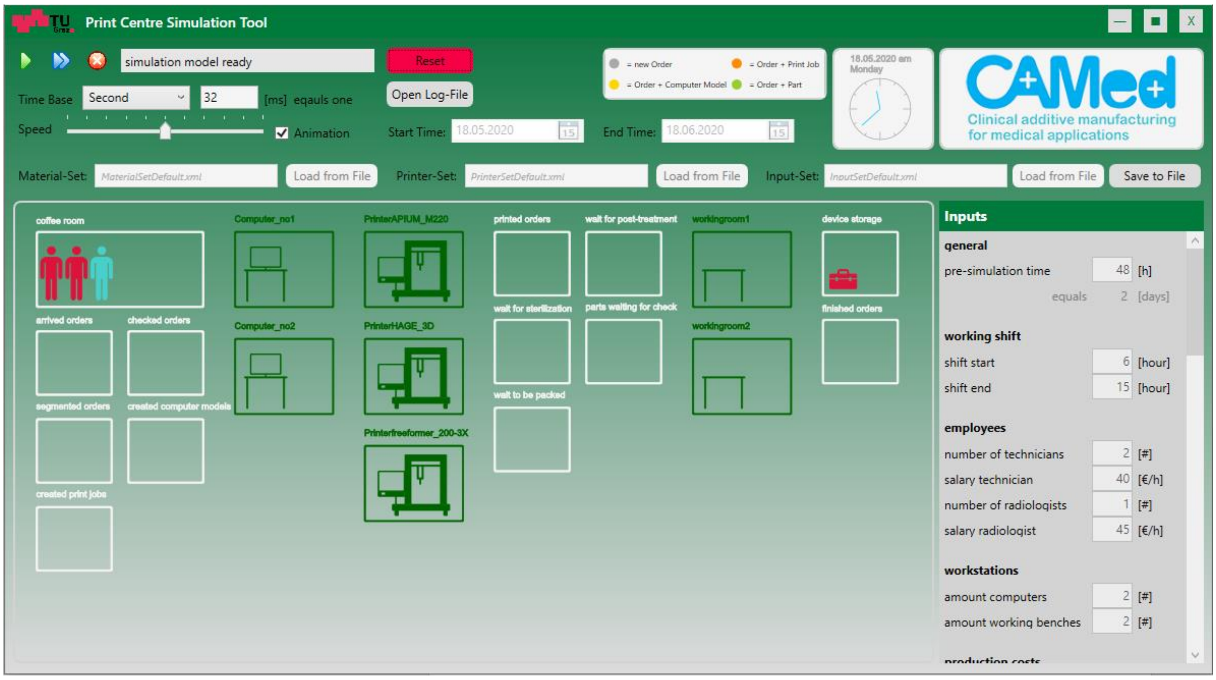

Figure 5.14 User interface of the main program

The software is able to run a simulation and display the live state graphically. The speed of the simulation can be varied via the interface. The animation can also be deactivated by deselecting "Animation". If so, the simulation runs as fast as possible in background and gives out the results as files.

5.9.1.1 Description of the User Interface

Though the user interface is designed as self-explaining as possible the major buttons and functions are described in Figure 5.15.

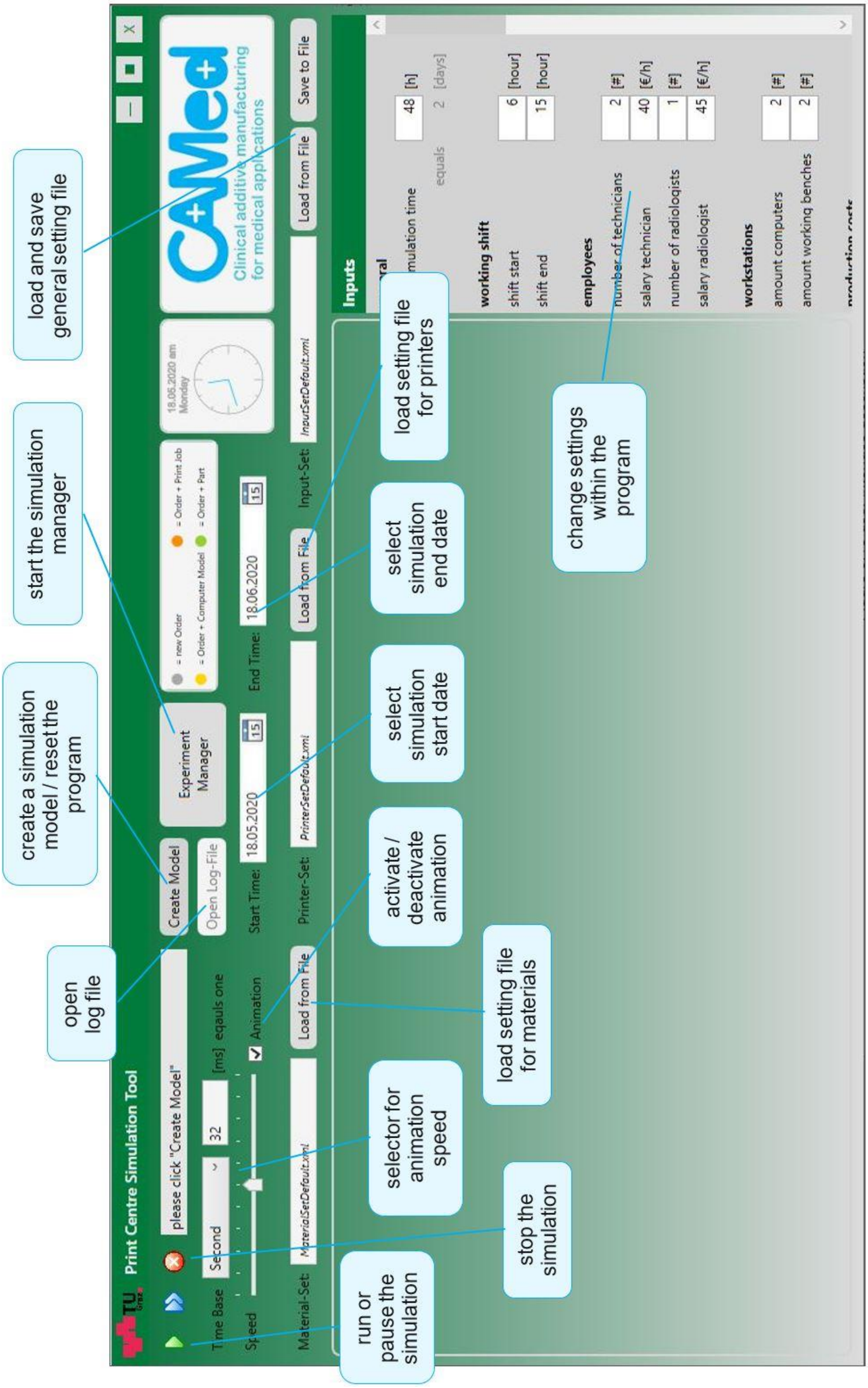

Figure 5.15 User interface description

5.9.1.2 Display of Orders within the Animation

Orders are displayed as circles in different colours and a number above. The colour depends on the order status (see Figure 5.16 and Figure 5.17), while the number represents the number of the order.
Orders that are not currently treated are displayed in holding areas, to show that they wait to be handled.

Figure 5.16 Legend for order display

Figure 5.17 Draw finished orders

5.9.1.3 Display of Employees

There are two different types of employees at the 3D-PC, technicians and radiologists. They are displayed with different colours, red for technicians, blue for radiologists (see Figure 5.18). If an entity is not busy it is drawn in a holding area to represent its availability.

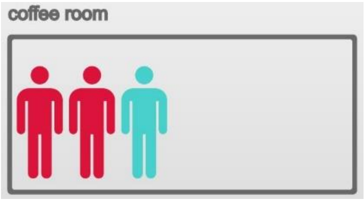

Figure 5.18 Display of idle employees

5.9.1.4 Display of Activity Printer Repair

If a printer is broken and gets repaired by an employee, the corresponding entity is drawn next to the printer and a hammer is added to show the status (shown in Figure 5.19).

Figure 5.19 Display of activity printer repair

5.9.2 Random Numbers

At certain points of the simulation, random numbers are used to make decisions (e.g. failure of parts, order properties). To ensure the traceability and repeatability of the simulation results, it was necessary to use one random number generator, that creates a list of random numbers per configuration and replication. Therefore, a global list with all required random numbers is created at the very beginning of every simulation run. Every time a random function gets called later in the simulation, a number from this list is taken

The seed number is a value to initialize the random function. When the same seed number is used to call the random function at the same time of the simulation, it will create exactly the same random number list. Within this simulation software the seed number is generated with the “hashcode”-function from a string. This function delivers representative number for a given letter sequence. The used string is a combination of the names of the settings files used (material set, printer set and input set).

5.9.3 Experiment Manager

The experiment manager (Figure 5.20) gives the user the ability to run the simulation with a given setup many times and create an average of all simulation runs. By creating many results with the same input values, it is possible to eliminate outliers due randomization. (see 5.9.2 Random Numbers)

The experiment manager window also provides the possibility to run multiple replications of different scenarios (setups). It allows the user to set up all inputs first and start all runs by just one click.

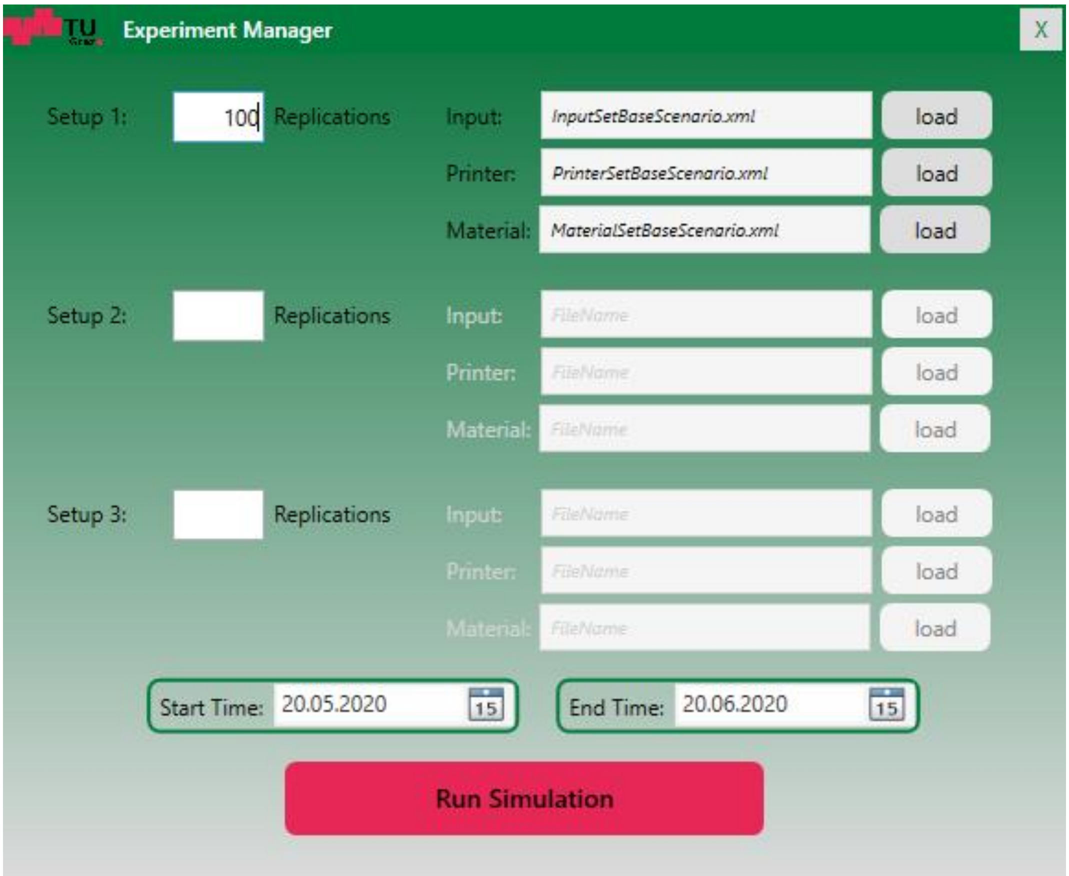

Figure 5.20 Experiment Manager Window

5.9.4 Handling of New Orders

When the simulation model is initialized, the first order is generated. With this order an event of type “newOrderArrival” gets created. The called event ensures that the order, within the event, is added to a waiting list at the controller and the next (new) order is created (see Figure 5.21). With this new order, another event of the type "OrderArrival" is created, but at a later time. The time depends on the selected input "avg. order arrival" and an exponential function.

```
//state change when a new Order arrives
protected override void StateChange(DateTime time, ISimulationEngine simEngine)
{
    // create the next order arrival event
    RequestOrderCheck orderCheckRequest = new RequestOrderCheck("GetOrderChecked", Order, time);

    PrintCentreControl.AddRequest(orderCheckRequest);

    PrintCentreControl.OrderInbox.HoldedEntities.Add(Order);

    // new order gets created
    create new order

    // time between new orders to arrive
    // if time is out of working time request new order not before beginn of the next working time
    DateTime nextOrderArrivalTime;
    double arrivalTimeMinutes = PrintCentreControl.Input.OrderArrivalMinutes;

    nextOrderArrivalTime = time + TimeSpan.FromMinutes(Distributions.Instance.Exponential(arrivalTimeMinutes));

    if (nextOrderArrivalTime.Hour < PrintCentreControl.Input.ShiftStartHour)...
    if (nextOrderArrivalTime.Hour > PrintCentreControl.Input.ShiftEndHour - 1)...
    EventOrderArrival nextOrderArrival = new EventOrderArrival(PrintCentreControl, nextOrderPeek);
    simEngine.AddScheduledEvent(nextOrderArrival, nextOrderArrivalTime);
}
```

Figure 5.21 Code for changing the state of the system when new order arrives

5.9.5 **Properties of New Orders**

The properties of orders, generated by the model during the simulation, can be set via various input parameters. The input parameters are percentages of all orders.

Each property is set by a random number from 0 to 100 (see chapter 5.9.2). The input percentages can be represented by parts of a number ray, which ranges from 0 to 100%. For example, if you choose 50% implants, 20% anatomical models and 30% tools for the type of order, this represents an area as shown in Figure 5.22. The code for the same example is shown in Figure 5.23.

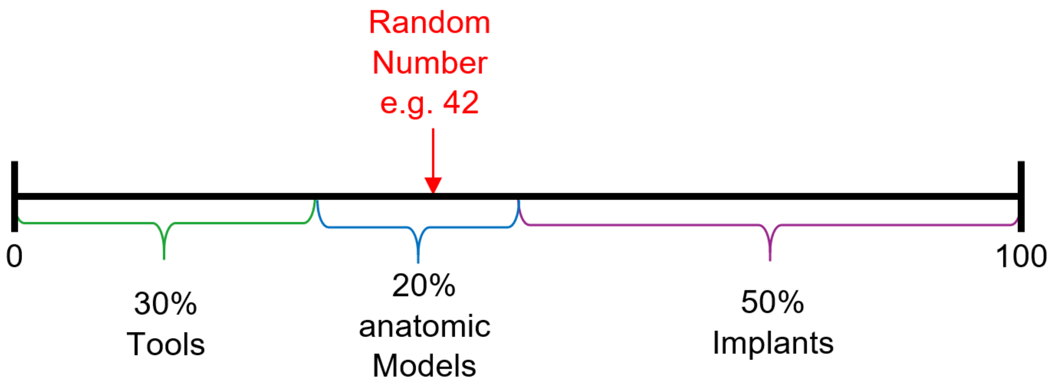

Figure 5.22 Graphical representation of the input percentage

The random number could for example be 42, thus the property type of this order would be “anatomic model”.

```
decisionNum = Distributions.Instance.RandomInteger(0, 100);
if (decisionNum < PrintCentreControl.Input.OrderProbTypeImplant)
{
    SingleSkillsNewOrder[0] = new SingleSkill("implant", 0);
    SingleSkillsNewOrder[5] = new SingleSkill("sterilization", 1);
}
else
{
    if (decisionNum >= (100 - PrintCentreControl.Input.OrderProbTypeModel))
    {
        SingleSkillsNewOrder[0] = new SingleSkill("model", 0);
        SingleSkillsNewOrder[5] = new SingleSkill("sterilization", 0);
    }
    else
    {
        SingleSkillsNewOrder[0] = new SingleSkill("tool", 0);
        SingleSkillsNewOrder[5] = new SingleSkill("sterilization", 1);
    }
}
```

Figure 5.23 Code for setting order properties when a new order is created

This explained method is used for every order property which is set by percentages (urgency, type, post-treatment etc.).

5.9.6 Properties and Attributes

Attributes of entities are primarily programmed as skills of entities and stored in a skill set, which can be imagined like a table where all skills of an entity are saved and can be addressed by their line (= index). The used HCCM framework (Furian, 2016) also offers a method called “hasSingleSkill”, which can be used to find out if an entity has a particular skill. If so, the method returns the value “true”.

To get an easier overview of all used skills they are shown in the following Table 5.7.

Index	Skill Description	Skill Values	Level Values
Print Order			
0	*Type of part*	*Implant / Model / Tool*	*0*
1	*Urgency*	*Low / Normal / High*	*0*
2	*Material*	*[depends on available materials]*	*0*
3	*Failed order check before*	*Failed*	*0, 1, 2, …*
4	*Required post-treatment*	*Post-treatment*	*0 = not required* *1 = required*
5	*Required sterilization*	*Sterilization*	*0 = not required* *1 = required*
Image			
0	*Patient name*	*[Name]*	*0*
1	*Bodypart*	*[Bodypart]*	*0*
Computer Model			
0	*Created by*	*[Employee who created it]*	*0*
1	*Date / Time*	*[Time of creation]*	*0*
2	*Version of the computer model*	*version*	*1, 2, 3, …*
3	*Calculated weight of the computer model*	*Weight*	*[Weight]*
4	*Version of the related print job*	*PrintJobVersion*	*1, 2, 3, …*

Index	Skill Description	Skill Values	Level Values
Print Job			
0	*Used material*	*Order.SkillSet.Skill[2]*	*[Used amount]*
1	*Required time to print*	*PrintingTime*	*[Duration in minutes]*
2	*Version of print job*	*Version*	*1, 2, 3, …*
3	*Interruptibility of the printing process*	*Interruptible*	*0 = false* *1 = true*
4	*Completion of the printing process*	*Completion*	*0 - 100*
Part			
0	*Date / Time*	*[Time of creation]*	*0*
1	*If part is cleaned*	*Cleaned*	*0 = false* *1 = true*
2	*Weight of part*	*Weight*	*[Weight]*
3	*Is part finished*	*Release*	*0 = false* *1 = true*
4	*Required post-treatment*	*Post-treatment /* *No post-treatment*	*No post-treatment:* *0* *Post-treatment:* *0 = to do* *1 = done*
5	*Required sterilization*	*Sterilization /* *No sterilization*	*No sterilization:* *0* *Sterilization:* *0 = to do* *1 = done*
6	*Is part checked*	*Checked*	*0 = no check* *1 = checked & ok* *2 = checked & fault*
Printer			
0, …	*Used materials*	*[One skill for each possible material]*	*0*

Table 5.7 Skills of entities in the code of the simulation of the 3D-PC

5.9.7 Weight of Parts

The weight respectively the volume of a part is defined at the state change end event of the activity “Model Part” therefore, a Gaussian function is created from the input parameters of the average weight per type and a random number that is included in this function is taken. This value describes the volume of the particular computer model and is stored as a property in the skillset. The explained code is shown in Figure 5.24.

```
SingleSkill[] SingleSkillsNewModel = new SingleSkill[5];
SingleSkillsNewModel[0] = new SingleSkill(Employee.ToString(), 0);      //created by
SingleSkillsNewModel[1] = new SingleSkill(time.ToString(), 0);          //time of creation
SingleSkillsNewModel[2] = new SingleSkill("version", 1);                //actual version = 1 at first creation
SingleSkillsNewModel[4] = new SingleSkill("PrintJobVersion", 1);        //actual version of created PrintJob = 1 when new

switch (Order.SkillSet.Skills[0].Skill)
{
    case "implant":
        double i = Distributions.Instance.GaussianDistribution(Controller.Input.AvgComputerModelVolumeImplant, 20);
        SingleSkillsNewModel[3] = new SingleSkill("volume", Convert.ToInt32(i));
        break;
    case "model":
        double m = Distributions.Instance.GaussianDistribution(Controller.Input.AvgComputerModelVolumeModel, 20);
        SingleSkillsNewModel[3] = new SingleSkill("volume", Convert.ToInt32(m));
        break;
    case "tool":
        double t = Distributions.Instance.GaussianDistribution(Controller.Input.AvgComputerModelVolumeTool, 20);
        SingleSkillsNewModel[3] = new SingleSkill("volume", Convert.ToInt32(t));
        break;
}
SkillSet modelSkillSet = new SkillSet(SingleSkillsNewModel);
EntityComputerModel newComuterModel = new EntityComputerModel(_order.Identifier, modelSkillSet);
//adding computer model to order
Order.ComputerModel = newComuterModel;
```

Figure 5.24 Code for creating a computer model of ordered part from a print job

5.9.8 Printer Downtime

To simulate printer failures, each printer was given attributes that store the operating hours and the hours to the next failure.

The total operating time to the next failure is recreated after each failure using an exponential function based on the input parameter "Average Breakdown Hours".

The complete handling is best explained with the example below.

Step 1: A new part shall be printed, therefore the activity “Produce Part” with its start event is called (Figure 5.25).
The start event calculates the printing time depending on the total printing time and the “Completion” of the given print job and sets the end event at the calculated time.
After this step the program checks, if the remaining hours to the next failure of the printer are bigger than the printing time for the actual part. If they are not, the event “Printer Failure” is created and called when the printers remaining working hours are at zero.

```
public override void StateChangeStartEvent(DateTime time, ISimulationEngine simEngine)
{
    // end time for activity printing is calculated from total printing time and completion (= int from 1 to 100)
    _printingTime = TimeSpan.FromMinutes(Order.PrintJob.SkillSet.Skills[1].Level * (100 - Order.PrintJob.SkillSet.Skills[4].Level) / 100);
    _endTimePrinting = time + _printingTime;
    EndTime = _endTimePrinting;

    simEngine.AddScheduledEvent(this.EndEvent, _endTimePrinting);

    if (Printer.RemainingWorkingHours < _printingTime.TotalHours)
    {
        DateTime failureTime = time.AddHours(Printer.RemainingWorkingHours);
        EventPrinterFailure nextPrinterFailure = new EventPrinterFailure(_controller, _printer, _order);
        _possiblePrinterFailure = nextPrinterFailure;
        simEngine.AddScheduledEvent(nextPrinterFailure, failureTime);
    }

    Employee.IsBusy = false;
    Controller.CoffeeRoom.HoldedEntities.Add(Employee);
}
```

Figure 5.25 Code for changing the state of the system when printing process is started

Step 2: The event printer failure is called, it sets the printers attribute “is broken” to true, requests the activity “Repair Printer” and removes the scheduled end event of the activity “Produce Part” and calls the event Produce Part immediately.

```
protected override void StateChange(DateTime time, ISimulationEngine simEngine)
{
    // look for actual printing activity
    Activity currentFailingPrintingProcess = _controller.CurrentActivities.Where
        (n => n.EndEvent.ParentActivity.ActivityName == "ActivityProducePart"
        && n.AffectedEntities.Contains(Printer)).First();

    Printer.IsBroken = true;

    RequestPrinterRepair printerRepairRequest = new RequestPrinterRepair("GetPrinterRepaired", _printer, time);
    Controller.AddRequest(printerRepairRequest);

    simEngine.RemoveScheduledEvent(currentFailingPrintingProcess.EndEvent);
    currentFailingPrintingProcess.EndEvent.Trigger(time, simEngine);

    //request to start printing again happens in ActivityProducePart.EndEvent
}
```

Figure 5.26 Code for changing the state of the system when a printer failure occurs

Step 3: The end event of the activity “Produce Part” is called (Figure 5.27). An if statement checks whether the part has been completely finished, if not (like when a printer breakdown occurs) the following lines are executed.

- The print job’s completion is set to “true”
- The used material is subtracted from the storage
- Delete the part if input parameter “reject part after repair” is true
- The print job is requested to be started again

```
public override void StateChangeEndEvent(DateTime time, ISimulationEngine simEngine)
{
    EntityMaterial neededMaterial = new EntityMaterial(Order.PrintJob.SkillSet.Skills[0].Skill, Order.PrintJob.SkillSet.Skills[0].Level);
    EntityMaterial actualMaterial = Controller.MyStorageRoom.Materials.Where(p => p.Name == neededMaterial.Name).First();

    //check if printing process is finished
    if (_endTimePrinting == time)[...]
    else //if printing not 100% finished
    {
        TimeSpan interval = _endTimePrinting - time;
        double newCompletionPercent = 100 - (100 * (interval.TotalMilliseconds / _printingTime.TotalMilliseconds));
        int newCompletion = Convert.ToInt32(Math.Floor(newCompletionPercent));            //completion in full %
        int oldCompletion = Order.PrintJob.SkillSet.Skills[4].Level;
        Order.PrintJob.SkillSet.Skills[4].Level = newCompletion;

        // subtracting the used material from material on stock
        actualMaterial.OverallAmount -= neededMaterial.OverallAmount * (newCompletion - oldCompletion) / 100;

        // delete printer failure event if printing was interrupted before event
        // eg. if printing is interupted for an urgent part and printer failure was planed
        if (_possiblePrinterFailure != null && Printer.IsBroken == false)[...]
        Printer.WorkingMinutes += interval.TotalMinutes;
        Printer.OverallWorkingHours += _printingTime.TotalHours;

        // delete part after printer breakdown of input is set so
        if (Controller.Input.RejectPartAfterFailure == true && Printer.IsBroken == true)[...]

        RequestPreparePrinter printJobSendingRequest = new RequestPreparePrinter("GetPrintJobSendedToPrinter", Order, time);
        Controller.AddRequest(printJobSendingRequest);
        Controller.OrderInboxSendToPrinter.HoldedEntities.Add(Order);

        Printer.IsBusy = false;
    } // endif printing NOT completely finished

    Order.ProcessTime += (time - this.StartTime);
    Order.Costs += (time - this.StartTime).TotalHours * Printer.CostsPerHour * (1 + Controller.Input.ManufacturingOverheads);
    [order new material when amount on stock is under limit]

}//end StateChangeEndEvent
```

Figure 5.27 Code for changing the state of the system when a printing process is stopped, after a printer failure occurred

Step 4: The start event activity “Repair Printer” is executed. The duration of the activity is calculated with an exponential function and the input parameter “duration printer repair”. The end event is set after the calculated timespan. (Figure 5.28)

```
public override void StateChangeStartEvent(DateTime time, ISimulationEngine simEngine)
{
    Employee.IsBusy = true;
    double duration = Controller.Input.DurationPrinterRepair;
    EndTime = time + TimeSpan.FromMinutes(Distributions.Instance.Exponential(duration));
    simEngine.AddScheduledEvent(this.EndEvent, EndTime);
}
```

Figure 5.28 Code for changing the state of the system when a printer repair is started

Step 5: The printer got repaired and the end event of the activity “Repair Printer” is called (Figure 5.29).

The printers working hours are set to zero and the printers operating hours until the next breakdown is created with an exponential function and the input parameter “average breakdown hours”. Afterwards the repair costs are computed and the printer’s attribute “Is Broken” is set to false.

```
public override void StateChangeEndEvent(DateTime time, ISimulationEngine simEngine)
{
    _printer.WorkingHours = 0;
    _printer.CompleteTimeToNextBreakdown = Distributions.Instance.Exponential(_printer.AverageBreakdownHours);

    Printer.RepairCosts += (time - this.StartTime).TotalHours * Employee.Salary * (1 + Controller.Input.ManufacturingOverheads);
    _repairCosts= (time - this.StartTime).TotalHours * (Employee.Salary * (1 + Controller.Input.ManufacturingOverheads) + Printer.CostsPerHour);

    Employee.IsBusy = false;
    Controller.CoffeeRoom.HoldedEntities.Add(Employee);
    Printer.IsBroken = false;
}
```

Figure 5.29 Code for changing the state of the system when a printer repair has ended

5.9.9 Calculation of Order Costs

Every order entity has an attribute called “costs”, where all processing costs (printing costs, material costs, hourly wages, etc.) are summed up. These processing costs for each order are actualized with each step within the simulation and every time an order is handled. Therefore, the employee’s salary, all machine and material costs as well as transportation costs are added.

For example, if a print job gets executed and the part is printed, the costs are calculated like shown in Equation 5.1 and its belonging Table 5.8. The corresponding code lines are displayed in Figure 5.30.

$$c = m * c_m * \left(1 + c_{m_{overhead}}\right) + t_p * c_p$$

Equation 5.1 Calculation of printing costs

c	costs for activity produce part	[€]
m	used material amount	[g]
c_m	price for material per weight	[€/g]
$c_{m\_overhead}$	material overheads	[%]
t_p	printing time	[h]
c_p	printer costs per hour	[€/h]

Table 5.8 Index for Equation 5.1

```
creating the skillset for produced part

#region material costs
double costs = neededMaterial.OverallAmount * actualMaterial.CostPerWeight * (1 + Controller.Input.MaterialOverheads)
    + Order.PrintJob.SkillSet.Skills[1].Level/60 * Printer.CostsPerHour;          //printing time of print job * machine costs

EntityPart currentPart = new EntityPart(Order.Identifier, costs, partSkillSet);
Order.Part = currentPart;
Order.Costs += currentPart.Costs;
#endregion
```

Figure 5.30 Code for creating a part and calculating its costs

5.9.10 Calculation of Total Costs

After finishing the simulation run, the method "CreateSimulationResultsFileAfterStop" is called. This function's first actions are to calculate the total costs of all orders at different states (e.g. new orders, orders with print jobs waiting to be printed, finished orders, etc.). To do this, the program takes each entity of the type "order" in a list and adds up the costs, that the respective orders have caused up to that point - these are stored as attributes "costs" of the order (see Figure 5.31). Afterwards the end status costs are corrected with the costs of the start time status of the simulation (status after pre-simulation time).

```
foreach (ActivityOrderNewMaterial i in MyPrintCentreControl.MyMaterialOrders)
{
    sumMaterialOrderCost += i.OrderCosts;
}
if (sumMaterialOrderCost >= MyInterimStatus.MaterialOrderCost)
    MyEndStatus.MaterialOrderCost = sumMaterialOrderCost - MyInterimStatus.MaterialOrderCost;

foreach (ActivityPrinterRepair i in MyPrintCentreControl.MyPrinterRepairs)
{
    sumPrinterRepairCost += i.RepairCosts;
}
if (sumPrinterRepairCost >= MyInterimStatus.PrinterRepairCost)
    MyEndStatus.PrinterRepairCost = sumPrinterRepairCost - MyInterimStatus.PrinterRepairCost;
```

Figure 5.31 Code for calculating of the summed order costs for end status

5.9.11 Calculation of Printer Downtime Costs

The printer downtime costs represent the costs for missed production opportunity due to printers are in maintenance or repair. They are calculated and saved for every printer individually.

```
#region calculating printer downtime costs
//average Price of all materials
foreach (EntityMaterial mat in MyPrintCentreControl.Input.MaterialList)
{
    avgMaterialCost += mat.CostPerWeight;
}
avgMaterialCost = avgMaterialCost / MyPrintCentreControl.Input.MaterialList.Count();

//calculating average material amout
avgMaterialAmount = (MyPrintCentreControl.Input.AvgComputerModelVolumeImplant * MyPrintCentreControl.Input.OrderProbTypeImplant / 100
    + MyPrintCentreControl.Input.AvgComputerModelVolumeModel * MyPrintCentreControl.Input.OrderProbTypeModel / 100
    + MyPrintCentreControl.Input.AvgComputerModelVolumeTool * MyPrintCentreControl.Input.OrderProbTypeTool / 100) / 3;

//sum up repair time for every printer
foreach (ActivityPrinterRepair printerRepair in MyPrintCentreControl.MyPrinterRepairs)
{
    printerRepair.Printer.RepairTime += (printerRepair.EndTime - printerRepair.StartTime).TotalHours;
}

foreach (EntityPrinter printer in MyPrintCentreControl.MyPrinters)
{
    TimeSpan overallTime = this.EndTime - this.StartTime;
    overallWorkingTime = overallTime.TotalDays * (MyPrintCentreControl.Input.ShiftEndHour - MyPrintCentreControl.Input.ShiftStartHour);
    double repairPerc = printer.RepairTime / overallWorkingTime;

    printer.DownTimeCosts = printer.Workload * repairPerc * avgMaterialAmount * avgMaterialCost;
}
#endregion
```

Figure 5.32 Code for calculating of the printer downtime costs

To get the costs, first the average material price and the average amount of material required are calculated (Figure 5.32). The next step is to determine the complete repair time for each printer and save it to the printer's variable. Then the repair time of the printers get divided by the working hours during the whole simulation run (this value is further called "repair percentage").
The printer downtime costs are calculated with the assumption, that the printer would have been needed with the same percentage than the printer's workload (e.g. if a printer has a workload of 30% and a repair percentage of 10%, it would have been needed in 30% of the time when it got repaired → 3%) (see Figure 5.33 with Table 5.9 and Equation 5.2).

$$c_d = w * r * m_{avg} * c_{avg}$$

Equation 5.2 Calculation of printer downtime costs

c_d	downtime costs	[€]
w	workload	[%]
r	repair percentage	[%]
m_{avg}	average material amount	[g]
c_{avg}	average material price	[€/g]

Table 5.9 Index for Equation 5.2

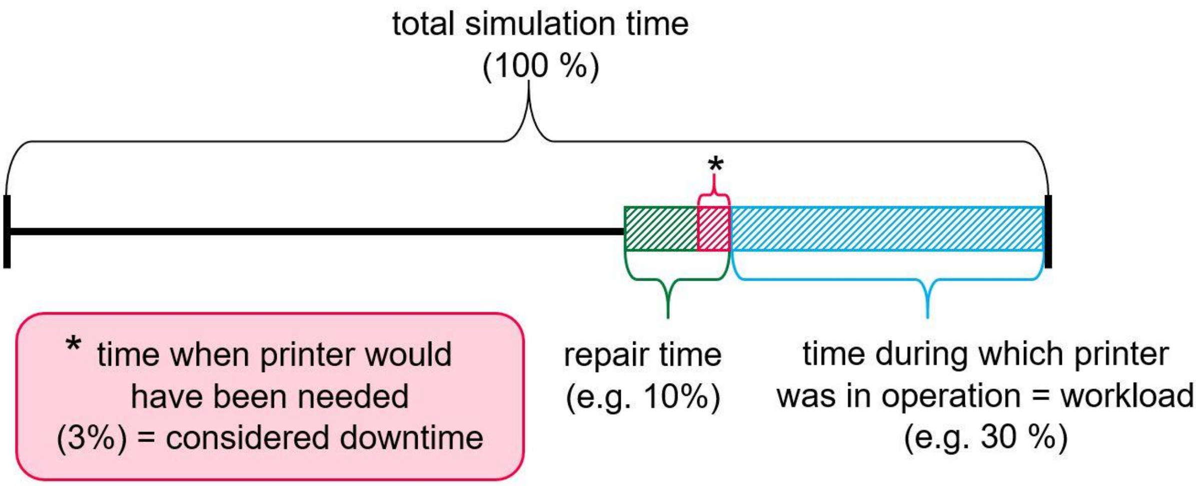

Figure 5.33 Graphical representation of the printer downtime cost calculation

6 Scenarios and Evaluation

As already explained in the introduction, there are hardly any empirical values yet available for the printing centre. However, since the accuracy of the simulation results depends on the input parameters, a basic scenario was defined for the evaluation of the software.

6.1 Base Scenario

This scenario (see Table 6.1) contains values, which allows to judge if the results are reasonable. The definitions and explanations of the different input parameters can be found in chapter 5.4.

general			
pre-simulation time	48	h	~1 order arrives per day, so after 2 days a normal situation should be achieved

working shift			
shift start	7	hour	typical shift start at the LKH Graz
shift end	15	hour	typical shift end at the LKH Graz

employees			
technicians	2	#	expected number of technicians for start phase
salary technicians	35	€/h	average normal salary for a trained technician
radiologists	1	#	expected number of radiologists for start phase
salary radiologists	45	€/h	average normal salary for a trained radiologist

workstations			
amount computers	2	#	assumption that every technician has computer (radiologist uses a computer of a technician)
amount working benches	2	#	limited to the available space at the 3D-PC

production costs			
manufacturing overheads	10	%	start value assumption
material overheads	10	%	start value assumption
fixed order costs	7,00	€	assumed costs for sending a package via post
transportation costs	2,50	€	assumed costs for internal transport service at the LKH on average.

material			
time for placing order	10	h	assumption that employees start to work, have a morning meeting, check new orders and place material orders after
time when orders arrive	8	h	post for business customers delivers in the morning
order size	2	units	local storage capacity is not so big, so only small amounts are stored and reordered

printer			
average printer repair time	90	min	empirical value of smaller printers for demounting the material, cleaning printer and adjust or repair smaller things
reject part after repair	yes		for safety reasons and to prevent faulty parts, they are produced again if the printer fails during production

orders			
time between orders	450	min	7,5 orders per week expected; with 7 days (8 hours per day)
probability false order	50	%	especially at the beginning it is very likely that orders are not exactly defined and technicians have to contact the customer
probability type implant	77	%	*see calculation below*
probability type model	15	%	*see calculation below*
probability urgency low	20	%	assumption since there are no empirical values yet
probability urgency high	20	%	assumption since there are no empirical values yet
probability post treatment	100	%	for medical uses the surface has to be post treated every time
probability interruptible	0	%	currently it is not possible to change the base plate of the printer, thus it is not possible to interrupt the printing and print another part in-between

computer model			
average volume implant	25	g	average weight of head implant
average volume model	280	g	average weight of an anatomic model
average volume tool	40	g	average weight for smaller devices and tools

part			
probability part fault	15	%	assumption since there are no empirical values yet

activity durations			
check order	15	min	expected time, no empirical value
consult customer	15	min	expected time, no empirical value
segmenting	30	min	expected time, no empirical value
modelling	60	min	expected time, no empirical value
creating print job	15	min	expected time, no empirical value
prepare printer	5	min	expected time, no empirical value
printing time per weight	10	min/g	on base of printed test parts
clean part	5	min	expected time, no empirical value
check part	15	min	expected time, no empirical value
post-treatment	20	min	expected time, no empirical value
sterilization	1440	min	will be done in another department, average value for parts is 24 hours
pack part	15	min	expected time, no empirical value
send part	0	min	finished parts are going to be picked up from the 3D-PC by the customers

Table 6.1 Input values base scenario

Calculation for order type percentages

Expected orders: *1* implant per day........................*250* implants per year
1 anatomic model per week ....*50* anatomic models per year
1 medical tool every two weeks ..*25* medical tools per year

(On base of a typical working periods in Austria there are 250 working days per year respectively 50 weeks.)

325 parts per year → percentages of orders: 77% implants
15% anatomic models
8% medical tools

The base scenario was defined including two printers that where installed at 3D-PC at that time.

printer: APIUM M220	
name	APIUM M220
printer number	100
type	filament
costs per hour	€ 40,00
average break down hours	100
skills (printable materials)	PEEK

Table 6.2 Printer APIUM M220 base scenario

printer: HAGE 3D	
name	HAGE 3D
printer number	200
type	filament
costs per hour	€ 30,00
average break down hours	100
skills (printable materials)	ABS, ASA, PA, PA12, PC, PCTG, PETG, PMMA, PP, PPSU, PVDF, TPC, TPU

Table 6.3 Printer HAGE 3D base scenario

The scenario also includes the definition of the available material, which can be used for printing parts. The materials in Table 6.4 were mainly set by the printers' possible materials and is the initial stock for the start of a simulation run. For the price of the individual plastics, various supplier prices were used and averaged (Filamentworld, 2020; Niceshops GmbH, 2020; Reichelt Elektronik Verwaltungs-GmbH, 2020) a detailed price list can be found under attachment 9-A.

Material PEEK	
name	PEEK
amount of units	2
unit size	250 g
unit type	filament
minimal amount on stock	100 g
costs per weight	0,92 €/g

Material ABS	
name	ABS
amount of units	1
unit size	500 g
unit type	filament
minimal amount on stock	300 g
costs per weight	0,025 €/g

Material ASA	
name	ASA
amount of units	1
unit size	500 g
unit type	filament
minimal amount on stock	300 g
costs per weight	0,03 €/g

Material PA	
name	PA
amount of units	1
unit size	500 g
unit type	filament
minimal amount on stock	300 g
costs per weight	0,075 €/g

Material PA12	
name	PA12
amount of units	1
unit size	500 g
unit type	filament
minimal amount on stock	300 g
costs per weight	0,083 €/g

Material PC	
name	PC
amount of units	1
unit size	500 g
unit type	filament
minimal amount on stock	300 g
costs per weight	0,055 €/g

Material PCTG	
name	PCTG
amount of units	1
unit size	500 g
unit type	filament
minimal amount on stock	300 g
costs per weight	0,123 €/g

Material PETG	
name	PETG
amount of units	1
unit size	500 g
unit type	filament
minimal amount on stock	300 g
costs per weight	0,029 €/g

Material PMMA	
name	PMMA
amount of units	1
unit size	500 g
unit type	filament
minimal amount on stock	300 g
costs per weight	0,067 €/g

Material PP	
name	PP
amount of units	1
unit size	500 g
unit type	filament
minimal amount on stock	300 g
costs per weight	0,07 €/g

Material PPSU	
name	PPSU
amount of units	1
unit size	500 g
unit type	filament
minimal amount on stock	300 g
costs per weight	0,026 €/g

Material TPC	
name	TPC
amount of units	1
unit size	500 g
unit type	filament
minimal amount on stock	300 g
costs per weight	0,141 €/g

Table 6.4 Materials base scenario

6.2 Evaluation and Results

It was necessary to check if the simulation was able to react sensitively and understandable on different inputs. Therefore, it was essential to create different scenarios, simulate them with the program and compare the results. This chapter describes the used scenarios and the gained insights. The scenarios for the evaluation were defined by modifying selected values of the base scenario (the base scenarios is described in chapter 0). The base scenario represents a situation as it is expected and seems to be a realistic assumption for the future. The aim for the other scenarios is to create situations where it is easy to predict how the system will behave (e.g. If an extreme high amount of orders is sent to 3D-PC, it is very likely that the system is overloaded and cannot handle all orders.).

The following scenario descriptions are modified scenarios of the base scenario, for this reason, only the changed parameters are mentioned. The presented result values are the ones from the base scenario compared to the results from the respective scenario. For each of these scenarios a simulation over 4 months with 200 replications each was examined.

- **High staff scenario**
  In this scenario more employees are hired and an additional computer is available for the employees (see Table 6.5).

employees		
technicians	4	#
radiologists	2	#
workstations		
amount computers	3	#

Table 6.5 Input changes high staff scenario

Expectation: The number of finished orders and the printer's workload should raise because more employees are available.

Result:	finished orders:	132,74	→	148,71	↑ (slightly raised)
	workload APIUM M220:	8,15%	→	8,32%	↑ (slightly raised)
	workload HAGE_3D:	54,23%	→	60,72%	↑ (slightly raised)

The values changed as expected but the printer's workloads are still far away from an optimal solution. This means, there are still other factors than the staff, which limit the system.

- **2 shift scenario**
  The aim of this scenario is to see how big the difference with two shifts, plus 3 technicians per shift is. To create this situation 6 technicians are hired (see Table 6.6).

working shift		
shift start	6	hour
shift end	22	hour
employees		
technicians	6	#
workstations		
amount computers	3	#

Table 6.6 Input changes 2 shift scenario

<u>Expectation</u>: The workload of the printers should raise, because the employees are longer at work to start the printers.

<u>Result</u>:					
	finished orders:	132,74	→	188,45	↑↑ (raised)
	workload APIUM M220:	8,15%	→	8,30%	↑ (slightly raised)
	workload HAGE_3D:	54,23%	→	83,00%	↑↑ (raised)

The values changed as expected and also raised compared to the "High Staff Scenario". This leads to the conclusion, that the number of working shifts is a limiting factor, that has to be considered. The reason therefore is, that if employees work only till 4 pm and a print job for example lasts only three hours, the printer is idle until the next morning, when the next shift starts.

- **2 shift more reorders scenario**
  The inputs are the same as at the 2 shift scenario but in advance more units of material are reordered if the limit on stock is reached, so that more material is available (see Table 6.7).

working shift		
shift start	6	hour
shift end	22	hour
employees		
technicians	3	#
workstations		
amount computers	3	#
material		
order size	5	units

Table 6.7 Input changes 2 shift scenario

Expectation: The workload of the printers should raise compared to the 2 shift scenario. Due to bigger material order sizes the number of material orders should decrease.

Result:	finished orders:	132,74	→	189,08	↑↑ (raised)
	workload APIUM M220:	8,15%	→	8,32%	↑ (slightly raised)
	workload HAGE_3D:	54,23%	→	83,12%	↑↑ (raised)

The results raised as it was expected and also changed in comparison to the “2 Shift Scenario” which attests the assumption, that the reorders (which affects the available materials) are a limiting factor to the system.

- **Overload scenario**

New orders arrive in a higher frequency, so that in sum more orders arrive over the simulation time (see Table 6.8).

orders		
time between orders	60	min

Table 6.8 Input changes overload scenario

Expectation: The workload of the 3D-PC will get to its maximum (with the given employees, material etc.) and not all orders will be handled.

Result:	arrived orders:	239,51	→	1102,01	↑↑↑ (highly raised)
	finished orders:	132,74	→	138,19	↑ (slightly raised)
	workload APIUM M220:	8,15%	→	10,95%	↑↑ (raised)
	workload HAGE_3D:	54,23%	→	54,50%	↑ (slightly raised)

The results show that, compared to the base scenario, the workloads are only minimally raising, while the arrived orders raised extremely. This scenario confirms the results of the other scenarios and shows that the system is limited by many factors and the software is able to simulate these limiting factors.

- **3 printer scenario**

For this scenario a second printer of type “HAGE 3D”, with the exact same properties as the first one was added.

Expectation: Through this it should be possible to print and finish more orders with the same number of employees.

Result:	finished orders:	132,74	→	166,25	↑↑ (raised)
	workload APIUM M220:	8,15%	→	6,21%	↓↓ (reduced)
	workload HAGE_3D:	54,23%	→	40,06%	↓↓ (reduced)
	workload HAGE_3D_2:	- %	→	32,88%	↑↑ (raised)

The results show that the finished orders raised while the particular workloads reduced. This can be explained by the fact that more print jobs can be executed parallel and it is more likely for a new print job that a printer is free. The reduced workload of the APIUM M220 is possibly caused by too few employees. More printed parts mean more post treatments. A technician first checks if a part is printed and shall be post treated to finish the related order and starts new orders afterwards, so it is possible that print jobs have to wait a little bit longer even if more printers are available.

The relevant data for comparison is shown in Table 6.9. The highest values are marked in red. The simulation results lead to various clear insights (see chapter 6.2.1 "Additional Insights").

	Base scenario	High staff	2 shift	2 shift more reorders	Overload	3 printer
Arrived Orders*	239,51	237,55	237,48	237,91	1102,01	238,07
Orders Finished*	132,74	148,71	188,45	189,08	138,19	166,25
Total Caused Costs [€]	218.769,79	243.813,06	303.636,67	304.112,42	235.569,52	262.675,31
Workloads						
APIUM M220	8,15%	8,32%	8,30%	8,81%	10,95%**	6,21%
HAGE_3D	54,23%	60,72%	83,00%	83,12%	54,50%	40,06%
HAGE_3D_2	-	-		-	-	32,88%
Material Orders*	16,33	17,55	20,86	13,61	17,30	18,67
Printer Repairs*	18,32	20,02	27,25	26,90	19,61	22,72

Table 6.9 Comparison Scenario Results

* Decimal numbers result from averaging over 200 simulation results.

** If the value is put in relation to "Arrived Orders" it is not the highest value.

6.2.1 **Additional Insights**

With the gained values from the simulated scenarios it was possible to identify some factors that limit the system and therefore play a significant role to find an optimal solution. Analysing these factors lead to additional insights which will help to optimize the 3D-PC.

→ Arrived orders as limit
The overload scenario shows that the workload of the printer "HAGE_3D" raised only slightly, compared to the base scenario. This shows that the base scenario has to be very close at the maximum possible level for arriving orders. We can also see that the workload of the printer "APIUM M220" is slightly higher than the one in the base scenario. This is caused by more arrived orders with the same percentage of peek-orders, since the printer "APIUM M220" is the only one which can print peek materials.

→ Employees as limit
As we can see from the results the workload of printer "HAGE_3D" raises only for six percentages from the base scenario compared to high staff scenario. The 2-shift scenario has one technician less than the high staff scenario but delivers a much higher workload of printer "HAGE_3D". It can be assumed, that this may be attributed to the longer overall working shift. This leads to the conclusion that the simulation is able to react on a change of employees. But the result also reveals, that only increasing the technicians does not deliver the highest workload, because also other factors limit the 3D-PC and create a bottleneck.

→ Material as limit
The comparison of the "2 shift"- and the "2 shift more reorders"-scenario leads to the insight, that not only employees and their working shift create a bottleneck, but also the available material. The less material is available the more often material has to be ordered and the more often a print job cannot be executed and has to wait until the material order arrives.

Summarizing the results and the above mentioned aspects, it can be said that the simulation software works as expected and delivers plausible results.

6.2.2 Limitations

Due to environmental situations and boundary conditions of the project, the software, as well as the results, are limited by following factors.

Because of the COVID-19 pandemic during spring 2020 in Austria, it was not possible to meet technicians or radiologist of the LKH Graz. Therefore, it was not possible to personally discuss the boundary conditions of the software and the processes in the 3D-PC. The exchange of information was therefore purely digital, which means that errors in understanding and communication can never be completely ruled out. For the same reason it was also not possible to adjust the inputs for the base scenario with technicians and doctors within the hospital.

7 Outlook

As already mentioned, the validity of the simulation results depends on how well the inputs reflect the reality, so it is essential to adjust them by time to refine the model.

Throughout the whole project assumptions were made which significantly influence the simulation results. These assumptions are explained and justified in chapter 5.8. All assumptions are legitimate at the time of creation of this thesis and have been clarified with the other parties involved in the project. If it should turn out at a later point in time that an assumption is no longer permissible, the affected part in the simulation software must be adapted and the changes have to be documented.

The unrestricted rights of use (with the exception of commercial purposes), as well as editing rights to the software, have been granted to the Institute for Business Informatics of Graz University of Technology.

Some possible changes to the software, that could improve the software's performance and usability are listed at this point:

- **Adapting the input for exact order definition**
  At the current state orders can only be defined by percentages of all orders. All combinations of order types, urgency, material etc. are possible.
  If – at a later point in time – it turns out that specific combinations are not allowed/possible and some combinations of attributes are dependent from each other, the input could be changed in order to predefine exact orders, that arrive over a certain time. To implement this to the software it would be necessary to define individual events for each type of order.
- **Implementing more cost aspects**
  To get a full TCO of the 3D-PC many cost aspects can be added to the simulation. Various costs like investment costs, software costs, amortization of machines etc. are currently not implemented.

8 Index

8.1 Figure Index

8.2 **Diagram Index**

8.3 **Table Index**

8.4 Equation Index

8.5 Bibliography

3dhubs. (2020). *3dhubs*. https://www.3dhubs.com/de/?device=c&position=&utm_source=adwords_g&utm_medium=cpc&utm_campaign=10035253599&utm_content=103924117627&utm_term=%2B3dhubs&hsa_grp=103924117627&hsa_ver=3&hsa_src=g&hsa_kw=%2B3dhubs&hsa_ad=434643396138&hsa_cam=10035253599&hsa_ne

Additively. (2020). *Additively*. https://www.additively.com/de/lernen/3d-printing-technologies

Arbez, G., & Birta, L. G. (2007). Dynamic System Behaviour. In *Modelling and Simulation*. Springer, Cham. https://doi.org/10.1007/978-3-030-18869-6

Borshchev, A., & Filippov, A. (2004). From System Dynamics and Discrete Event to Practical Agent Based Modeling: Reasons, Techniques, Tools. *The 22nd International Conference of the System Dynamics Society*, *66*(11), 25–29. http://www.econ.iastate.edu/tesfatsi/systemdyndiscreteeventabmcompared.borshchevfilippov04.pdf

Cai, S., Xi, J., & Chua, C. (2012). A Novel Bone Scaffold Design Approach Based on Shape Function and All-Hexahedral Mesh Refinement. *Methods in Molecular Biology (Clifton, N.J.)*, *868*, 45–55. https://doi.org/10.1007/978-1-61779-764-4_3

Checkland, P. (1981). *Systems Thinking, Systems Practice*. UK: Wiley.

Chen, P. P. S. (1976). The Entity-Relationship Model - toward a Unified View of Data. *ACM Transactions on Database Systems (TODS)*, *1*(1), 9–36. https://doi.org/10.1145/320434.320440

Chwif, L., Ribeiro, M., Pereira-Barretto, M., & Paul, R. (2001). On Simulation Model Complexity. *Winter Simulation Conference Proceedings*, *1*. https://doi.org/10.1145/510378.510448

Djanatliev, A., & German, R. (2013). Prospective healthcare decision-making by combined system dynamics, discrete-event and agent-based simulation. *Proceedings of the 2013 Winter Simulation Conference - Simulation: Making Decisions in a Complex World*, 270–281. https://doi.org/10.1109/WSC.2013.6721426

Dodziuk, H. (2016). Applications of 3D printing in healthcare. *Kardiochirurgia i Torakochirurgia Polska*, *13*(3), 283–293. https://doi.org/10.5114/kitp.2016.62625

Filamentworld. (2020). *Filamentworld*. https://www.filamentworld.de

Furian, N. (2016). *HCDESlib - Github*. https://github.com/nikolausfurian/HCDESLib/wiki

Furian, N., O'Sullivan, M., Walker, C., Vössner, S., & Neubacher, D. (2015). A conceptual modeling framework for discrete event simulation using hierarchical control structures. *Simulation Modelling Practice and Theory*, *56*, 82–96. https://doi.org/10.1016/j.simpat.2015.04.004

Gartner Group Inc. (2020). *Gartner*. https://www.gartner.com/en

Han, X., Yang, D., Yang, C., Spintzyk, S., Scheideler, L., Li, P., Li, D., Geis-Gerstorfer, J., & Rupp, F. (2019). Carbon Fiber Reinforced PEEK Composites Based on 3D-Printing Technology for Orthopedic and Dental Applications. *Journal of Clinical Medicine*, *8*(2), 240. https://doi.org/10.3390/jcm8020240

Hay, A. M., Valentin, E. C., & Bijlsma, R. A. (2006). Modeling Emergency Care in Hospitals: A Paradox - The Patient Should not Drive the Process. *Proceedings of the 2006 Winter Simulation Conference*, 439–445. https://doi.org/10.1109/WSC.2006.323113

Maidstone, R. (2012). Discrete Event Simulation, System Dynamics and Agent Based Simulation: Discussion and Comparison. *System*, 1–6.

Mann, C., & Thum, M. (2020). *3d-grenzelos*. https://www.3d-grenzenlos.de/glossar/prefix:m/

Meta Group Consulting. (2020). *Meta Group Consulting*. http://www.meta-group.com/meta-services/Pages/Consulting-Services-for-International-Institutions.aspx

Nance, R. E. (1994). The Conical Methodology and the evolution of simulation model development. *Annals of Operations Research*, *53*(1), 1–45. https://doi.org/10.1007/BF02136825

Niceshops GmbH. (2020). *3djake*. https://www.3djake.at/

Redman, B., Kirwin, B., & Berg, T. (1998). TCO : A Critical Tool for Managing IT. *Gartner Group,Inc.*, 48.

Reichelt Elektronik Verwaltungs-GmbH. (2020). *reichelt*. https://www.reichelt.at

Robinson, S. (1994). *Simulation projects: building the right conceptual model*.

Robinson, S. (2004). *Simulation: The Practice of Model Development and Use* (Vol. 1, Issue 1). John Wiley & Sons, Ltd.

Robinson, S. (2008). Conceptual modelling for simulation Part I: definition and requirements. *Journal of the Operational Research Society*, *59*(3), 278–290. https://doi.org/10.1057/palgrave.jors.2602368

Robinson, S., Arbez, G., Birta, L. G., Tolk, A., & Wagner, G. (2016). Conceptual modeling: Definition, purpose and benefits. *Proceedings - Winter Simulation Conference*, *2016-Febru*. https://doi.org/10.1109/WSC.2015.7408386

Robinson, S., Brooks, R. J., Kotiadis, K., & van der Zee, D. J. (2010). Conceptual modeling for discrete-event simulation. In *Conceptual Modeling for Discrete-Event Simulation*. CRC Press. https://doi.org/10.1201/9781439810385

Schriber, T. J., & Brunner, D. T. (2013). Inside simulation software: how it works and why it matters. *Proceedings of the 2013 Winter Simulation Conference - Simulation: Making Decisions in a Complex World*, 23–30.

Wild, M., & Herges, S. (2000). *Total Cost of Ownership (TCO) – Ein Überblick* (Issue 1). https://docplayer.org/1961568-Total-cost-of-ownership-tco-ein-ueberblick.html

Zeigler, B. P. (1987). Hierarchical, Modular Discrete-Event Modelling in an Object-Oriented Environment. *Simulation*, *49*(5), 219–230. https://doi.org/10.1177/003754978704900506

9 Attachments

A Price List for Materials

Material	Price per Unit [€/#]	Weight per Unit [g/#]	Price per weight [€/g]	Average Price [€/g]	Data from
PEEK	199,9 339,99 199,99	250 500 250	0,800 0,680 0,800	0,760	(Niceshops GmbH, 2020) (Filamentworld, 2020) (Reichelt Elektronik Verwaltungs-GmbH, 2020)
ABS	29,99 17,99	1000 750	0,030 0,024	0,027	(Niceshops GmbH, 2020) (Filamentworld, 2020)
ASA	25,39 59,99	750 2300	0,034 0,026	0,030	(Niceshops GmbH, 2020) (Filamentworld, 2020)
PA	59,99 51,99	750 753	0,080 0,069	0,075	(Niceshops GmbH, 2020) (Filamentworld, 2020)
PA12	44,99 52,99	750 500	0,060 0,106	0,083	(Niceshops GmbH, 2020) (Filamentworld, 2020)
PC	49,99 29,99	1000 500	0,050 0,060	0,055	(Niceshops GmbH, 2020) (Filamentworld, 2020)
PCTG	92,72	750	0,124	0,124	(Reichelt Elektronik Verwaltungs-GmbH, 2020)
PETG	70	2500	0,028	0,028	(Filamentworld, 2020)
PMMA	46,99 39,9	500 1000	0,094 0,040	0,067	(Niceshops GmbH, 2020) (Filamentworld, 2020)
PP	34,99 41,99	500 600	0,070 0,070	0,070	(Niceshops GmbH, 2020) (Filamentworld, 2020)
PPSU	159,9 99,9	500 500	0,320 0,1998	0,260	(Niceshops GmbH, 2020) (Filamentworld, 2020)
PVDF	129,9	500	0,260	0,260	(Niceshops GmbH, 2020)
TPC	53,99 129,9	500 750	0,108 0,173	0,141	(Reichelt Elektronik Verwaltungs-GmbH, 2020) (Filamentworld, 2020)

Table Attachment 1 Price list materials

B Result.txt

Final Report

SIMULATION CREATED: 24.06.2020 16:53:19

SIMULATED FOR THE PERIOD FROM: 01.06.2020 TO 01.07.2020
USED INPUT-SETUP:INPUTSET_BASESCENARIO.XML
USED MATERIAL-SETUP: MATERIALSET_BASESCENARIO.XML
USED PRINTER-SETUP: PRINTERSET_BASESCENARIO.XML
USED SEEDNUMBER: -1169878161
(INPUTS ARE DISPLAYED AT THE END OF THE REPORT)

-------------------------OVERVIEW-------------------------

TOTAL WORKING TIME IN PERIOD:	256H	(DAILY SHIFT FROM 7:00 - 15:00)
TOTAL ARRIVED ORDERS:	60	
- OF TYPE IMPLANT:	43	
- OF TYPE MODEL:	9	
- OF TYPE TOOL:	8	
TOTAL FINISHED ORDERS:	23	
TOTAL CAUSED COSTS:	48 126,41 €	

SUM OF ORDERS LISTED BELOW CAN DIFFER FROM "TOTAL ARRIVED ORDERS". TOTAL ARRIVED ORDERS ARE ONLY ARRIVED ONES WITHIN SIMULATION TIME WITHOUT PRE-SIMULATION!

	NUMBER	COSTS
NEW ARRIVED ORDERS:	2	-
CHECKED ORDERS:	0	0,00 €
SEGMENTED ORDERS:	0	0,00 €
CREATED COMPUTER MODELS:	0	0,00 €
CREATED PRINT JOBS:	35	3 455,38 €
PRINTED ORDERS:	1	1 837,37 €
WAITING FOR POST-TREATMENT:	0	0,00 €
WAITING FOR STERILIZATION:	0	0,00 €
PARTS WAITING FOR CHECK:	0	0,00 €
WAITING TO BE PACKED:	0	0,00 €
FINISHED ORDERS:	23	41 983,34 €
MATERIAL ORDERS:	5#	975,00 €
PRINTER REPAIR TIME:	12,41H	850,30 €
PRINTER REPAIRS:	9#	
WORKLOAD APIUM_M220:	14,1 %	
DOWNTIME COSTS :	0,00 €	
WORKLOAD HAGE_3D:	57,4 %	
DOWNTIME COSTS :	0,03 €	

-------------------------DETAILED-------------------------

ORDERED MATERIAL: 5

FROM: 03.06.2020 07:00:00 TO: 04.06.2020 08:00:00 COSTS: 130,00 € PCTG: 1000UNITS; TYPE: FILAMENT
FROM: 06.06.2020 13:24:27 TO: 08.06.2020 08:00:00 COSTS: 467,00 € PEEK: 500UNITS; TYPE: FILAMENT
FROM: 09.06.2020 11:25:00 TO: 11.06.2020 08:00:00 COSTS: 74,00 € PMMA: 1000UNITS; TYPE: FILAMENT
FROM: 21.06.2020 08:06:09 TO: 22.06.2020 08:00:00 COSTS: 37,00 € ASA: 1000UNITS; TYPE: FILAMENT
FROM: 24.06.2020 11:24:01 TO: 26.06.2020 08:00:00 COSTS: 267,00 € PVDF: 1000UNITS; TYPE: FILAMENT

REPAIRED PRINTERS: 9

03.06.2020 07:00:00 DURATION: 49MIN REPAIR COSTS: 56,75 € PRINTERHAGE_3D GETS REPAIRED BY TECHNICALEMPLOYEE_1
04.06.2020 07:00:00 DURATION: 3H14MIN REPAIR COSTS: 222,02 € PRINTERHAGE_3D GETS REPAIRED BY TECHNICALEMPLOYEE_2
05.06.2020 11:47:58 DURATION: 2H22MIN REPAIR COSTS: 162,71 € PRINTERHAGE_3D GETS REPAIRED BY TECHNICALEMPLOYEE_2
08.06.2020 14:36:26 DURATION: 1H17MIN REPAIR COSTS: 88,80 € PRINTERHAGE_3D GETS REPAIRED BY TECHNICALEMPLOYEE_2
11.06.2020 13:25:00 DURATION: 02MIN REPAIR COSTS: 2,31 € PRINTERHAGE_3D GETS REPAIRED BY TECHNICALEMPLOYEE_1
19.06.2020 09:15:00 DURATION: 36MIN REPAIR COSTS: 41,28 € PRINTERHAGE_3D GETS REPAIRED BY TECHNICALEMPLOYEE_2
22.06.2020 07:00:00 DURATION: 1H19MIN REPAIR COSTS: 90,23 € PRINTERHAGE_3D GETS REPAIRED BY TECHNICALEMPLOYEE_1
26.06.2020 07:00:00 DURATION: 1H30MIN REPAIR COSTS: 103,53 € PRINTERHAGE_3D GETS REPAIRED BY TECHNICALEMPLOYEE_1
28.06.2020 07:35:00 DURATION: 1H12MIN REPAIR COSTS: 82,66 € PRINTERHAGE_3D GETS REPAIRED BY TECHNICALEMPLOYEE_1

NEW ARRIVED ORDERS: 2

ORDER_NO62 TYPE: IMPLANT; URGENCY: NORMAL; MATERIAL: PPSU; CUSTOMER: ASSISTENZARZT, HOLZER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: REQUIRED

ORDER_NO63 TYPE: IMPLANT; URGENCY: NORMAL; MATERIAL: PEEK; CUSTOMER: UNFALLCHIRURG, BAUER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: REQUIRED

CHECKED ORDERS: 0

EMPTY

SEGMENTED ORDERS: 0

EMPTY

ORDERS WITH CREATED COMPUTER MODELS: 0

EMPTY

ORDERS WITH CREATED PRINT JOBS: 35

ORDER_NO16 TYPE: MODEL; URGENCY: NORMAL; MATERIAL: PC; CUSTOMER: ASSISTENZARZT, HOLZER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: NOT REQUIRED
COMPUTERMODEL_NO160016
PRINTJOB_NO160020 -> COMPLETION: 0%
TOTAL PROCESS TIME: 02:00:00
PROCESSING COSTS: 72,88 €

ORDER_NO17 TYPE: IMPLANT; URGENCY: NORMAL; MATERIAL: PETG; CUSTOMER: UNFALLCHIRURG, BAUER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: REQUIRED
COMPUTERMODEL_NO170018
PRINTJOB_NO170022 -> COMPLETION: 0%
TOTAL PROCESS TIME: 02:30:00
PROCESSING COSTS: 92,13 €

ORDER_NO19 TYPE: MODEL; URGENCY: NORMAL; MATERIAL: PP; CUSTOMER: ASSISTENZARZT, HOLZER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: NOT REQUIRED
COMPUTERMODEL_NO190019
PRINTJOB_NO190024 -> COMPLETION: 0%
TOTAL PROCESS TIME: 03:30:00
PROCESSING COSTS: 130,63 €

ORDER_NO20 TYPE: IMPLANT; URGENCY: NORMAL; MATERIAL: PA; CUSTOMER: UNFALLCHIRURG, BAUER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: REQUIRED
COMPUTERMODEL_NO200020
PRINTJOB_NO200025 -> COMPLETION: 0%
TOTAL PROCESS TIME: 02:00:00
PROCESSING COSTS: 72,88 €

ORDER_NO21 TYPE: TOOL; URGENCY: NORMAL; MATERIAL: TPC; CUSTOMER: ASSISTENZARZT, HOLZER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: REQUIRED
COMPUTERMODEL_NO210021
PRINTJOB_NO210026 -> COMPLETION: 0%
TOTAL PROCESS TIME: 02:30:00
PROCESSING COSTS: 92,13 €

ORDER_NO23 TYPE: MODEL; URGENCY: NORMAL; MATERIAL: PA12; CUSTOMER: ASSISTENZARZT, HOLZER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: NOT REQUIRED
COMPUTERMODEL_NO230024
PRINTJOB_NO230037 -> COMPLETION: 0%
TOTAL PROCESS TIME: 02:30:00
PROCESSING COSTS: 92,13 €

ORDER_NO24 TYPE: IMPLANT; URGENCY: NORMAL; MATERIAL: PC; CUSTOMER: UNFALLCHIRURG, BAUER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: REQUIRED
COMPUTERMODEL_NO240028
PRINTJOB_NO240038 -> COMPLETION: 0%
TOTAL PROCESS TIME: 02:30:00
PROCESSING COSTS: 92,13 €

ORDER_NO27 TYPE: IMPLANT; URGENCY: NORMAL; MATERIAL: TPC; CUSTOMER: ASSISTENZARZT, HOLZER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: REQUIRED
COMPUTERMODEL_NO270029
PRINTJOB_NO270039 -> COMPLETION: 0%
TOTAL PROCESS TIME: 03:00:00
PROCESSING COSTS: 111,38 €

ORDER_NO25 TYPE: IMPLANT; URGENCY: NORMAL; MATERIAL: PVDF; CUSTOMER: UNFALLCHIRURG, BAUER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: REQUIRED
COMPUTERMODEL_NO250031
PRINTJOB_NO250041 -> COMPLETION: 0%
TOTAL PROCESS TIME: 05:00:00
PROCESSING COSTS: 188,38 €

ORDER_NO28 TYPE: IMPLANT; URGENCY: NORMAL; MATERIAL: TPC; CUSTOMER: ASSISTENZARZT, HOLZER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: REQUIRED
COMPUTERMODEL_NO280032
PRINTJOB_NO280042 -> COMPLETION: 0%
TOTAL PROCESS TIME: 03:30:00
PROCESSING COSTS: 130,63 €

ORDER_NO31 TYPE: MODEL; URGENCY: NORMAL; MATERIAL: PVDF; CUSTOMER: ASSISTENZARZT, HOLZER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: NOT REQUIRED
COMPUTERMODEL_NO310033
PRINTJOB_NO310043 -> COMPLETION: 0%
TOTAL PROCESS TIME: 02:00:00
PROCESSING COSTS: 72,88 €

ORDER_NO32 TYPE: IMPLANT; URGENCY: NORMAL; MATERIAL: PPSU; CUSTOMER: UNFALLCHIRURG, BAUER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: REQUIRED
COMPUTERMODEL_NO320034
PRINTJOB_NO320044 -> COMPLETION: 0%
TOTAL PROCESS TIME: 02:30:00
PROCESSING COSTS: 92,13 €

ORDER_NO34 TYPE: IMPLANT; URGENCY: NORMAL; MATERIAL: PC; CUSTOMER: UNFALLCHIRURG, BAUER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: REQUIRED
COMPUTERMODEL_NO340035
PRINTJOB_NO340045 -> COMPLETION: 0%
TOTAL PROCESS TIME: 02:00:00
PROCESSING COSTS: 72,88 €

ORDER_NO35 TYPE: TOOL; URGENCY: NORMAL; MATERIAL: PC; CUSTOMER: UNFALLCHIRURG, BAUER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: REQUIRED
COMPUTERMODEL_NO350036
PRINTJOB_NO350046 -> COMPLETION: 0%
TOTAL PROCESS TIME: 02:30:00
PROCESSING COSTS: 92,13 €

ORDER_NO36 TYPE: IMPLANT; URGENCY: NORMAL; MATERIAL: PA12; CUSTOMER: ASSISTENZARZT, HOLZER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: REQUIRED
COMPUTERMODEL_NO360037
PRINTJOB_NO360048 -> COMPLETION: 0%
TOTAL PROCESS TIME: 02:30:00
PROCESSING COSTS: 92,13 €

ORDER_NO37 TYPE: IMPLANT; URGENCY: NORMAL; MATERIAL: PC; CUSTOMER: ASSISTENZARZT, HOLZER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: REQUIRED
COMPUTERMODEL_NO370038
PRINTJOB_NO370049 -> COMPLETION: 0%
TOTAL PROCESS TIME: 02:30:00
PROCESSING COSTS: 92,13 €

Order_no39

type: implant; urgency: normal; material: PC; customer: Unfallchirurg, Bauer
post-treatment: not required, sterilization: required
ComputerModel_no390039
PrintJob_no390050 -> completion: 0%
total process time: 02:00:00
processing costs: 72,88 €

Order_no38

type: implant; urgency: normal; material: ASA; customer: Unfallchirurg, Bauer
post-treatment: not required, sterilization: required
ComputerModel_no380040
PrintJob_no380051 -> completion: 0%
total process time: 05:00:00
processing costs: 188,38 €

Order_no41

type: implant; urgency: normal; material: PVDF; customer: Assistenzarzt, Holzer
post-treatment: not required, sterilization: required
ComputerModel_no410041
PrintJob_no410052 -> completion: 0%
total process time: 02:00:00
processing costs: 72,88 €

Order_no42

type: tool; urgency: normal; material: ABS; customer: Unfallchirurg, Bauer
post-treatment: not required, sterilization: required
ComputerModel_no420042
PrintJob_no420055 -> completion: 0%
total process time: 02:00:00
processing costs: 72,88 €

Order_no44

type: implant; urgency: normal; material: PVDF; customer: Unfallchirurg, Bauer
post-treatment: not required, sterilization: required
ComputerModel_no440044
PrintJob_no440056 -> completion: 0%
total process time: 02:30:00
processing costs: 92,13 €

Order_no46

type: tool; urgency: normal; material: PMMA; customer: Assistenzarzt, Holzer
post-treatment: not required, sterilization: required
ComputerModel_no460045
PrintJob_no460057 -> completion: 0%
total process time: 02:00:00
processing costs: 72,88 €

Order_no45

type: implant; urgency: normal; material: PETG; customer: Unfallchirurg, Bauer
post-treatment: not required, sterilization: required
ComputerModel_no450046
PrintJob_no450058 -> completion: 0%
total process time: 04:00:00
processing costs: 149,88 €

Order_no47

type: implant; urgency: normal; material: PMMA; customer: Unfallchirurg, Bauer
post-treatment: not required, sterilization: required
ComputerModel_no470047
PrintJob_no470059 -> completion: 0%
total process time: 02:00:00
processing costs: 72,88 €

Order_no48

type: tool; urgency: normal; material: ABS; customer: Assistenzarzt, Holzer
post-treatment: not required, sterilization: required
ComputerModel_no480048
PrintJob_no480061 -> completion: 0%
total process time: 04:00:00
processing costs: 149,88 €

Order_no49

type: implant; urgency: normal; material: PCTG; customer: Assistenzarzt, Holzer
post-treatment: not required, sterilization: required
ComputerModel_no490049
PrintJob_no490062 -> completion: 0%
total process time: 04:30:00
processing costs: 169,13 €

Order_no50

type: implant; urgency: normal; material: PC; customer: Unfallchirurg, Bauer
post-treatment: not required, sterilization: required
ComputerModel_no500050
PrintJob_no500065 -> completion: 0%
total process time: 02:30:00
processing costs: 92,13 €

ORDER_NO51 TYPE: IMPLANT; URGENCY: NORMAL; MATERIAL: ASA; CUSTOMER: UNFALLCHIRURG, BAUER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: REQUIRED
COMPUTERMODEL_NO510052
PRINTJOB_NO510066 -> COMPLETION: 0%
TOTAL PROCESS TIME: 02:00:00
PROCESSING COSTS: 72,88 €

ORDER_NO53 TYPE: TOOL; URGENCY: NORMAL; MATERIAL: PP; CUSTOMER: UNFALLCHIRURG, BAUER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: REQUIRED
COMPUTERMODEL_NO530053
PRINTJOB_NO530067 -> COMPLETION: 0%
TOTAL PROCESS TIME: 02:00:00
PROCESSING COSTS: 72,88 €

ORDER_NO54 TYPE: IMPLANT; URGENCY: NORMAL; MATERIAL: PPSU; CUSTOMER: ASSISTENZARZT, HOLZER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: REQUIRED
COMPUTERMODEL_NO540054
PRINTJOB_NO540068 -> COMPLETION: 0%
TOTAL PROCESS TIME: 02:00:00
PROCESSING COSTS: 72,88 €

ORDER_NO55 TYPE: IMPLANT; URGENCY: NORMAL; MATERIAL: PC; CUSTOMER: UNFALLCHIRURG, BAUER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: REQUIRED
COMPUTERMODEL_NO550055
PRINTJOB_NO550069 -> COMPLETION: 0%
TOTAL PROCESS TIME: 02:00:00
PROCESSING COSTS: 72,88 €

ORDER_NO56 TYPE: IMPLANT; URGENCY: NORMAL; MATERIAL: PMMA; CUSTOMER: UNFALLCHIRURG, BAUER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: REQUIRED
COMPUTERMODEL_NO560056
PRINTJOB_NO560070 -> COMPLETION: 0%
TOTAL PROCESS TIME: 02:30:00
PROCESSING COSTS: 92,13 €

ORDER_NO57 TYPE: IMPLANT; URGENCY: NORMAL; MATERIAL: ASA; CUSTOMER: ASSISTENZARZT, HOLZER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: REQUIRED
COMPUTERMODEL_NO570057
PRINTJOB_NO570071 -> COMPLETION: 0%
TOTAL PROCESS TIME: 02:30:00
PROCESSING COSTS: 92,13 €

ORDER_NO60 TYPE: MODEL; URGENCY: NORMAL; MATERIAL: PVDF; CUSTOMER: ASSISTENZARZT, HOLZER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: NOT REQUIRED
COMPUTERMODEL_NO600060
PRINTJOB_NO600075 -> COMPLETION: 0%
TOTAL PROCESS TIME: 03:00:00
PROCESSING COSTS: 111,38 €

ORDER_NO61 TYPE: IMPLANT; URGENCY: NORMAL; MATERIAL: PP; CUSTOMER: ASSISTENZARZT, HOLZER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: REQUIRED
COMPUTERMODEL_NO610061
PRINTJOB_NO610076 -> COMPLETION: 0%
TOTAL PROCESS TIME: 02:00:00
PROCESSING COSTS: 72,88 €

PRINTED ORDERS WAITING FOR CLEANING PART: 1

ORDER_NO11 TYPE: IMPLANT; URGENCY: NORMAL; MATERIAL: ASA; CUSTOMER: ASSISTENZARZT, HOLZER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: REQUIRED
COMPUTERMODEL_NO110011
PRINTJOB_NO110063 -> COMPLETION: 100%
PART_NO110041
[REJECTED PRINT JOBS:
PRINTJOB_NO110014
PRINTJOB_NO110031
PRINTJOB_NO110036
]
REJECTED PARTS:
PART_NO110021
PART_NO110024
PART_NO110034
TOTAL PROCESS TIME: 2.19:15:00
PROCESSING COSTS: 2 478,36 €

PARTS WAITING FOR POST-TREATMENT: 0

EMPTY

PARTS WAITING FOR STERILIZATION: 0

EMPTY

PARTS WAITING TO BE CHECKED: 0

EMPTY

PARTS WAITING TO BE PACKED: 0

EMPTY

FINISHED ORDERS: 23

ORDER_NO7
TYPE: IMPLANT; URGENCY: URGENT; MATERIAL: ABS; CUSTOMER: ASSISTENZARZT, HOLZER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: DONE
COMPUTERMODEL_NO70007
PRINTJOB_NO70010 -> COMPLETION: 100%
PART_NO70004
TOTAL PROCESS TIME: 1.08:15:00
PROCESSING COSTS: 1 216,88 €

ORDER_NO12
TYPE: MODEL; URGENCY: NORMAL; MATERIAL: PEEK; CUSTOMER: ASSISTENZARZT, HOLZER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: NOT REQUIRED
COMPUTERMODEL_NO120012
PRINTJOB_NO120015 -> COMPLETION: 100%
PART_NO120006
TOTAL PROCESS TIME: 2.01:00:00
PROCESSING COSTS: 2 421,04 €

ORDER_NO2
TYPE: MODEL; URGENCY: NORMAL; MATERIAL: PCTG; CUSTOMER: UNFALLCHIRURG, BAUER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: NOT REQUIRED
COMPUTERMODEL_NO20002
PRINTJOB_NO20004 -> COMPLETION: 100%
PART_NO20007
TOTAL PROCESS TIME: 4.16:01:50.6250000
PROCESSING COSTS: 3 751,34 €

ORDER_NO18
TYPE: IMPLANT; URGENCY: URGENT; MATERIAL: PPSU; CUSTOMER: ASSISTENZARZT, HOLZER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: DONE
COMPUTERMODEL_NO180017
PRINTJOB_NO180021 -> COMPLETION: 100%
PART_NO180009
TOTAL PROCESS TIME: 1.06:35:00
PROCESSING COSTS: 1 161,96 €

ORDER_NO1
TYPE: IMPLANT; URGENCY: NORMAL; MATERIAL: PMMA; CUSTOMER: UNFALLCHIRURG, BAUER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: DONE
COMPUTERMODEL_NO10001
PRINTJOB_NO10023 -> COMPLETION: 100%
PART_NO10010
REJECTED PRINT JOBS:
PRINTJOB_NO10001
PRINTJOB_NO10002
PRINTJOB_NO10003
PRINTJOB_NO10009
PRINTJOB_NO10017
REJECTED PARTS:
PART_NO10001
PART_NO10002
PART_NO10003
PART_NO10005
PART_NO10008
TOTAL PROCESS TIME: 4.18:29:52.2000000
PROCESSING COSTS: 4 174,68 €

ORDER_NO3 TYPE: IMPLANT; URGENCY: NORMAL; MATERIAL: PP; CUSTOMER: ASSISTENZARZT, HOLZER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: DONE
COMPUTERMODEL_NO30003
PRINTJOB_NO30005 -> COMPLETION: 100%
PART_NO30011
TOTAL PROCESS TIME: 1.06:35:00
PROCESSING COSTS: 1 159,88 €

ORDER_NO4 TYPE: IMPLANT; URGENCY: NORMAL; MATERIAL: PVDF; CUSTOMER: UNFALLCHIRURG, BAUER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: DONE
COMPUTERMODEL_NO40004
PRINTJOB_NO40006 -> COMPLETION: 100%
PART_NO40013
TOTAL PROCESS TIME: 1.07:45:00
PROCESSING COSTS: 1 204,35 €

ORDER_NO22 TYPE: IMPLANT; URGENCY: URGENT; MATERIAL: PEEK; CUSTOMER: ASSISTENZARZT, HOLZER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: DONE
COMPUTERMODEL_NO220022
PRINTJOB_NO220028 -> COMPLETION: 100%
PART_NO220016
REJECTED PRINT JOBS:
PRINTJOB_NO220027
REJECTED PARTS:
PART_NO220014
TOTAL PROCESS TIME: 1.12:45:00
PROCESSING COSTS: 1 503,92 €

ORDER_NO5 TYPE: IMPLANT; URGENCY: NORMAL; MATERIAL: TPC; CUSTOMER: ASSISTENZARZT, HOLZER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: DONE
COMPUTERMODEL_NO50005
PRINTJOB_NO50007 -> COMPLETION: 100%
PART_NO50015
TOTAL PROCESS TIME: 1.06:55:00
PROCESSING COSTS: 1 174,96 €

ORDER_NO6 TYPE: IMPLANT; URGENCY: NORMAL; MATERIAL: PVDF; CUSTOMER: ASSISTENZARZT, HOLZER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: DONE
COMPUTERMODEL_NO60006
PRINTJOB_NO60008 -> COMPLETION: 100%
PART_NO60017
TOTAL PROCESS TIME: 1.09:05:15.5380000
PROCESSING COSTS: 1 246,20 €

ORDER_NO9 TYPE: IMPLANT; URGENCY: NORMAL; MATERIAL: PA; CUSTOMER: ASSISTENZARZT, HOLZER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: DONE
COMPUTERMODEL_NO90009
PRINTJOB_NO90012 -> COMPLETION: 100%
PART_NO90019
TOTAL PROCESS TIME: 1.09:15:00
PROCESSING COSTS: 1 256,81 €

ORDER_NO14 TYPE: MODEL; URGENCY: NORMAL; MATERIAL: PEEK; CUSTOMER: UNFALLCHIRURG, BAUER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: NOT REQUIRED
COMPUTERMODEL_NO140014
PRINTJOB_NO140018 -> COMPLETION: 100%
PART_NO140012
TOTAL PROCESS TIME: 2.02:10:00
PROCESSING COSTS: 2 467,89 €

ORDER_NO26 TYPE: IMPLANT; URGENCY: URGENT; MATERIAL: PCTG; CUSTOMER: UNFALLCHIRURG, BAUER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: DONE
COMPUTERMODEL_NO260023
PRINTJOB_NO260029 -> COMPLETION: 100%
PART_NO260022
TOTAL PROCESS TIME: 1.06:55:00
PROCESSING COSTS: 1 172,26 €

ORDER_NO10 TYPE: IMPLANT; URGENCY: NORMAL; MATERIAL: PETG; CUSTOMER: UNFALLCHIRURG, BAUER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: DONE
COMPUTERMODEL_NO100010
PRINTJOB_NO100013 -> COMPLETION: 100%
PART_NO100020
TOTAL PROCESS TIME: 1.08:45:00
PROCESSING COSTS: 1 230,94 €

ORDER_NO29 TYPE: IMPLANT; URGENCY: URGENT; MATERIAL: PMMA; CUSTOMER: UNFALLCHIRURG, BAUER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: DONE
COMPUTERMODEL_NO290025
PRINTJOB_NO290032 -> COMPLETION: 100%
PART_NO290025
TOTAL PROCESS TIME: 1.10:25:00
PROCESSING COSTS: 1 300,66 €

ORDER_NO8 TYPE: IMPLANT; URGENCY: NORMAL; MATERIAL: PCTG; CUSTOMER: UNFALLCHIRURG, BAUER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: DONE
COMPUTERMODEL_NO80008
PRINTJOB_NO80030 -> COMPLETION: 100%
PART_NO80023
REJECTED PRINT JOBS:
PRINTJOB_NO80011
REJECTED PARTS:
PART_NO80018
TOTAL PROCESS TIME: 2.12:20:00
PROCESSING COSTS: 2 278,77 €

ORDER_NO30 TYPE: IMPLANT; URGENCY: URGENT; MATERIAL: TPC; CUSTOMER: ASSISTENZARZT, HOLZER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: DONE
COMPUTERMODEL_NO300026
PRINTJOB_NO300033 -> COMPLETION: 100%
PART_NO300027
TOTAL PROCESS TIME: 1.08:35:00
PROCESSING COSTS: 1 231,51 €

ORDER_NO13 TYPE: IMPLANT; URGENCY: NORMAL; MATERIAL: PA; CUSTOMER: ASSISTENZARZT, HOLZER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: DONE
COMPUTERMODEL_NO130013
PRINTJOB_NO130016 -> COMPLETION: 100%
PART_NO130026
TOTAL PROCESS TIME: 1.07:35:00
PROCESSING COSTS: 1 198,48 €

ORDER_NO33 TYPE: IMPLANT; URGENCY: URGENT; MATERIAL: PMMA; CUSTOMER: UNFALLCHIRURG, BAUER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: DONE
COMPUTERMODEL_NO330027
PRINTJOB_NO330035 -> COMPLETION: 100%
PART_NO330030
REJECTED PRINT JOBS:
PRINTJOB_NO330034
REJECTED PARTS:
PART_NO330028
TOTAL PROCESS TIME: 2.16:30:00
PROCESSING COSTS: 2 407,93 €

ORDER_NO43 TYPE: MODEL; URGENCY: URGENT; MATERIAL: PVDF; CUSTOMER: ASSISTENZARZT, HOLZER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: NOT REQUIRED
COMPUTERMODEL_NO430043
PRINTJOB_NO430053 -> COMPLETION: 100%
PART_NO430032
TOTAL PROCESS TIME: 2.02:50:00
PROCESSING COSTS: 1 776,98 €

ORDER_NO40 TYPE: IMPLANT; URGENCY: URGENT; MATERIAL: ASA; CUSTOMER: UNFALLCHIRURG, BAUER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: DONE
COMPUTERMODEL_NO400030
PRINTJOB_NO400060 -> COMPLETION: 100%
PART_NO400035
REJECTED PRINT JOBS:
PRINTJOB_NO400040
PRINTJOB_NO400054
REJECTED PARTS:
PART_NO400031
PART_NO400033
TOTAL PROCESS TIME: 1.18:21:19.7760000
PROCESSING COSTS: 1 534,87 €

ORDER_NO15 TYPE: MODEL; URGENCY: NORMAL; MATERIAL: ASA; CUSTOMER: ASSISTENZARZT, HOLZER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: NOT REQUIRED
COMPUTERMODEL_NO150015
PRINTJOB_NO150047 -> COMPLETION: 100%
PART_NO150036
REJECTED PRINT JOBS:
PRINTJOB_NO150019
REJECTED PARTS:
PART_NO150029
TOTAL PROCESS TIME: 4.23:18:52.7820000
PROCESSING COSTS: 3 963,67 €

ORDER_NO58 TYPE: IMPLANT; URGENCY: URGENT; MATERIAL: PCTG; CUSTOMER: ASSISTENZARZT, HOLZER
POST-TREATMENT: NOT REQUIRED, STERILIZATION: DONE
COMPUTERMODEL_NO580058
PRINTJOB_NO580072 -> COMPLETION: 100%
PART_NO580038
TOTAL PROCESS TIME: 1.06:15:00
PROCESSING COSTS: 1 147,37 €

-------------------------INPUTS--------------------------

PRE-SIMULATION TIME: 48 HOURS

WORKING SHIFT
SHIFT START: 7.00 HOUR
SHIFT END: 15.00 HOUR

EMPLOYEES
NUMBER OF TECHNICIANS: 2 #
SALARY TECHNICIANS: 35,00 €/H
NUMBER OF RADIOLOGISTS: 1 #
SALARY RADIOLOGISTS: 45,00 €/H

WORKSTATIONS
NUMBER OF COMPUTERS: 2 #
NUMBER OF WORK BENCHES: 2 #

COSTS
MANUFACTORING OVERHEADS: 10 %
MATERIAL OVERHEADS: 10 %
FIXED ORDER COSTS: 7,00 €
TRANSPORTATION COSTS: 2,50 €

MATERIAL
TIME FOR PLACING ORDER: 10.00 HOUR
TIME WHEN ORDER ARRIVES: 8.00 HOUR
ORDER SIZE: 2 UNITS

PRINTER
AVG. PRINTER REPAIR TIME: 90 H
REJECT PART AFTER REPAIR: TRUE

ORDERS
TIME BETWEEN ORDERS: 450 MIN
PROBABILITY FALSE ORDER: 50 %
PROBABILITY TYPE IMPLANT: 77 %
PROBABILITY TYPE MODEL: 15 %
PROBABILITY TYPE TOOL: 8 %
PROBABILITY URGENCY LOW: 20 %
PROBABILITY URGENCY HIGH: 20 %
PROBABILITY POST-TREATMENT: 100 %
PROBABILITY INTERRUPTIBLE: 0 %

COMPUTER MODEL
AVG. VOLUME IMPLANT: 25 G
AVG. VOLUME MODEL: 280 G
AVG. VOLUME TOOL: 40 G

PART
PROBABILITY PART FAULT: 15 %

ACTIVITY DURATIONS
ORDER CHECK: 15 MIN
CONSULT CUSTOMER: 15 MIN
SEGMENTING: 30 MIN
MODELLING: 60 MIN
CREATING PRINT JOB: 15 MIN
PREPARE PRINTER: 5 MIN
PRINTING TIME PER WEIGHT: 10 MIN/G
CLEAN PART: 5 MIN
CHECK PRODUCED PART: 15 MIN
POST-TREATMENT: 20 MIN
STERILIZATION: 1440 MIN
PACK PRODUCED PART: 15 MIN
SEND PART: 0 MIN

C TCO model V4.0 based on Gartner Group Inc

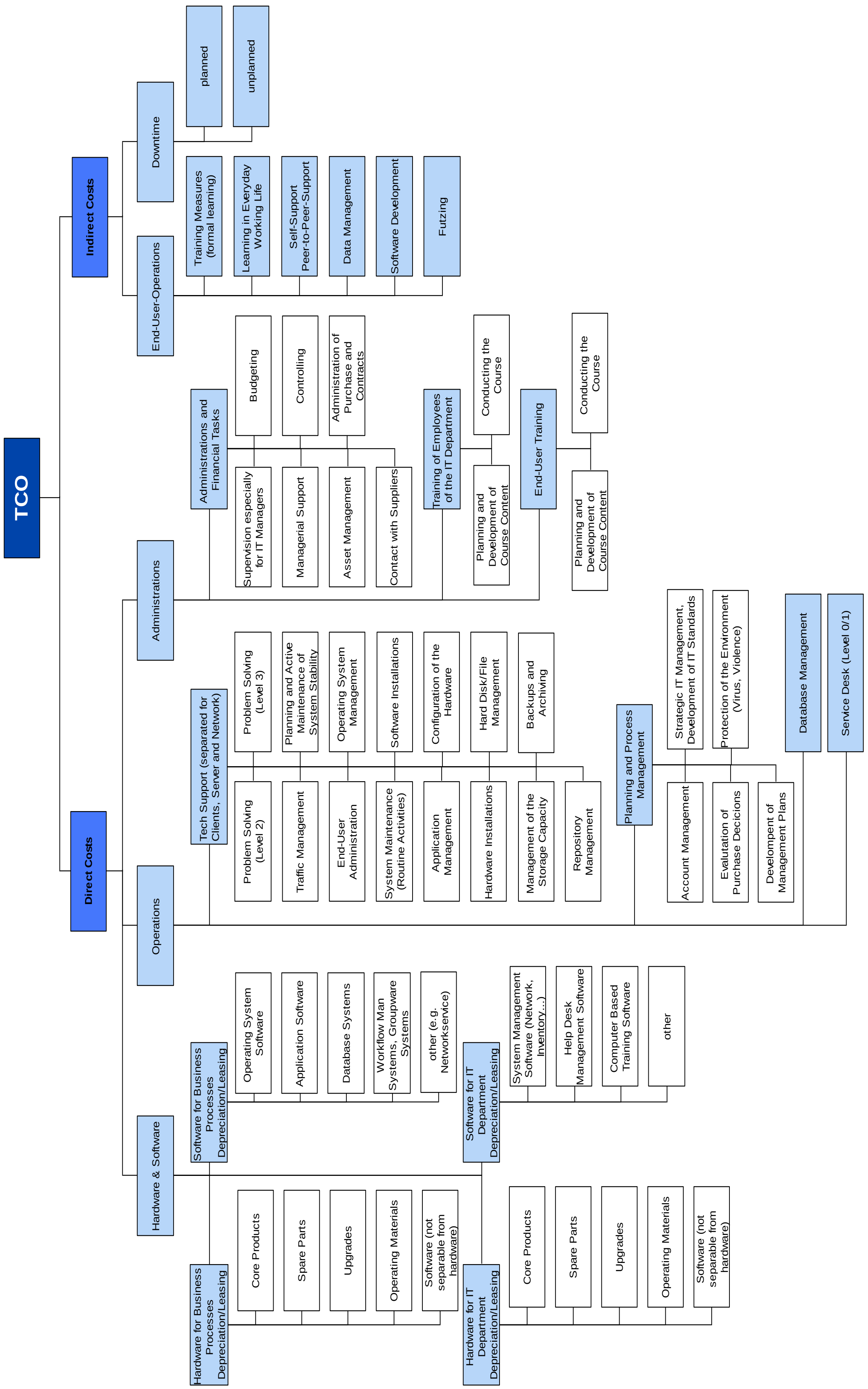

Figure Attachment 1 TCO-model 4.0 (illustrated by author based on Gartner Group Inc. (2020))