

# Simulation-based optimization methods for operative production control and scheduling, using discrete digital factory models

Application in the field of plastic processing

DIPLOMARBEIT

zur Erlangung des akademischen Grades

**Diplom-Ingenieur**

im Rahmen des Studiums

**Wirtschaftsingenieurwesen Informatik**

eingereicht von

**Manfred Neubacher, Bakk. techn.**

Matrikelnummer 0206653

an der  
Fakultät für Informatik der Technischen Universität Wien

Betreuung  
Betreuer: Ao. Univ.-Prof. Dipl.-Ing. Dr.techn. Felix Breitenecker  
Mitwirkung: Univ.Lektor Dipl.-Ing. Dr.techn. Thomas Löscher

Wien, 27.10.2011

\_\_\_\_\_  
(Unterschrift Verfasser)

\_\_\_\_\_  
(Unterschrift Betreuer)

Manfred Neubacher  
Hettenkofergasse 18-22/3/5  
1160 Wien

„Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.“

Wien, am 27. Oktober 2011

---

Manfred Neubacher

## Abstract

This thesis describes simulation-based optimization, where an optimizer is coupled with a simulator. The work presented here was performed in the field of the plastic processing industry and was part of a FFG (Austrian Research Promotion Agency) funded project within the program line “Factory of Tomorrow”. The development of the practical part of this work was done in a co-operation between the Technical University of Vienna and the Profactor GmbH located in Steyr, Austria.

Based on the process and data analyses, a simulation model was built using the simulation library SiRO. As optimization tool the HeuristicLab framework for heuristic and evolutionary algorithms was used. The optimization generates complex priority rules by using an *offspring selection genetic algorithm*. These complex priority rules are applied situationally for all decision points at each simulation replication.

The simulation delivers indices of the operating data from each replication to a rating model, where these indices are transferred into a common unit and then merged into a fitness value. The formulation of the rating model represents the company objectives (objective function). To construct a rating model, a rating framework was developed. After having performed the simulation and rating, the fitness value was delivered to the optimization. By comparing the different fitness values of different priority rules, it is possible to infer the quality of the generated rules. The best complex priority rule is obtained and based on this, a Gantt chart is provided to illustrate the obtained work schedules. A very interesting result of this study showed that the felt stress level of workers correlates with the backlog of work.

## Kurzfassung

Diese Diplomarbeit beschreibt simulationsbasierte Optimierung, bei der ein Simulator mit einem Optimierer gekoppelt wird. Die Arbeit wurde im Bereich der Plastik verarbeitenden Industrie durchgeführt und war Teil eines FFG (Österreichische Forschungsförderungsgesellschaft) geförderten Projektes der Programmlinie „Fabrik der Zukunft“. Der praktische Teil dieser Arbeit wurde im Rahmen einer Kooperation zwischen der Technischen Universität Wien und der Profactor GmbH in Steyr, Österreich durchgeführt. Aufbauend auf einer Prozess- und Datenanalyse wurde ein Simulationsmodell erstellt. Hierfür wurde die Simulationsbibliothek SiRO verwendet. Für die Optimierung wurde das HeuristicLab Framework für heuristische und evolutionäre Algorithmen verwendet. Unter Verwendung eines *offspring selection genetic algorithm* wurden komplexe Prioritätsregeln generiert, welche situativ für alle Entscheidungspunkte jedes Simulationslaufes angewandt werden.

Die von der Simulation gewonnenen Betriebsdaten werden an ein Bewertungsmodell als Kennzahlen übergeben, auf gemeinsame Einheit überführt und zu einem Fitnesswert zusammengeführt. Das Bewertungsmodell repräsentiert, durch die jeweilige Formulierung, die Unternehmensziele (Zielfunktion). Ein Rating Framework wurde entwickelt, um die Erstellung von Bewertungsmodellen zu vereinfachen. Nach Ausführen der Simulation und des Bewertungsprozesses wird der Fitnesswert an die Optimierung übergeben. Durch den Vergleich der Fitnesswerte verschiedener Prioritätsregeln kann auf die Qualität der erzeugten Regeln geschlossen werden. Basierend auf der besten gefundenen Prioritätsregel werden die erzeugten Arbeitspläne als *Gantt chart* bereitgestellt. Ein sehr interessantes Resultat dieser Studie zeigte, dass das gefühlte Stressniveau der Arbeiter mit dem jeweiligen Arbeitsrückstand korreliert.

## **Acknowledgements**

I wish to thank Professor Felix Breitenecker and his group for their ideas, interesting discussions and the possibility to develop this master thesis. I would like especially to thank Dr. Thomas Löscher for his help and contribution.

I also would like to extend my gratitude to Dr. Markurs Vorderwinkler and to all the members of the simulation based design & optimization group from Profactor GmbH, who have supported me within the project. Especially M.Sc. Helga Heiß for her important inputs and patience.

In addition, I wish to thank all the group members from the institutions involved in this project: Chair of Production and Operations Management, University of Vienna; Heuristics and Evolutionary Algorithms Laboratory, Upper Austria University of Applied Sciences; as well as the industries partner ASMA GmbH, ABF-Industrial Automation GmbH, Schneegans-Silicon GmbH and Arbeitsleben Gruber KG, for the interesting discussions in the meetings.

Thanks to the International Office, especially Dr. Mitar Pitzek, for a grant that allowed me to develop part of this work abroad. Many thanks to Professor Juan Ramonet and Professor Silvia Ramos from the hosting university (Faculty of Engineering, University of Buenos Aires, Argentina) for their enrichment ideas and interesting talks.

A lot of thanks to my family, who gave me the possibility to study and always supported me with my career.

Finally, I owe thanks to my lovely girlfriend Rita, for her love and understanding. Without her encouragement this work would not exist. I also want to thank her for many advices and the correction of this work.

# Index

<b>List of abbreviations.....</b>	<b>viii</b>
<b>List of figures and tables.....</b>	<b>x</b>
<b>1 Introduction.....</b>	<b>1</b>
1.1 Motivation.....	1
1.2 Problem description of a performed SBO study .....	3
1.3 Contents preview.....	3
<b>2 Production Planning and Control .....</b>	<b>4</b>
2.1 Manufacturing system .....	4
2.2 Planning and control.....	5
2.2.1 Production planning.....	6
2.2.2 Production control.....	8
2.2.3 Objectives of production planning and control .....	9
2.3 Difference between scheduling and dispatching .....	11
2.4 Trends in production management, theories and techniques.....	12
<b>3 Modelling and Simulation.....</b>	<b>16</b>
3.1 Background .....	16
3.1.1 System .....	16
3.1.2 Model .....	17
3.2 Simulation .....	18
3.2.1 Concepts related to modelling and simulation .....	19
3.2.2 Why do we use simulation? .....	20
3.2.2.1 Advantages of simulation.....	20
3.2.2.2 Disadvantages of simulation.....	20
3.2.3 Categories of simulation .....	21
3.3 Classification of models: Different ways of studying a system .....	22
3.4 Simulation study .....	23
3.5 A simple simulation example .....	26
<b>4 Simulation Software .....</b>	<b>29</b>
4.1 Classification of simulation and historical review .....	29
4.2 Random numbers.....	32
4.3 Selected simulation software .....	35
4.3.1 PowerSim.....	36

4.3.2 Plant Simulation .....	36
4.3.3 GPSS.....	36
4.3.4 SiRO .....	37
4.3.4.1 An exemplary implementation using SiRO.....	40
4.3.5 Simio.....	43
4.4 Case study: Comparison implementation using different simulation tools .....	44
<b>5 Rating (of simulation replications) .....</b>	<b>47</b>
5.1 What is meant by rating .....	47
5.2 The rating framework.....	47
5.2.1 Summary of the steps of a rating process .....	48
5.3 Formulation of the rating model .....	53
5.3.1 Different ratings.....	53
<b>6 Simulation-based optimization: basics, methods and applications.....</b>	<b>54</b>
6.1 Background .....	54
6.2 Methods of simulation-based optimization .....	56
6.3 Optimizers - Optimization packages .....	64
6.4 Scheme of a simulation-based optimization study .....	66
6.5 Application areas for simulation-based optimization .....	67
<b>7 Prototype application of a simulation-based optimization study.....</b>	<b>69</b>
7.1 Problem identification .....	69
7.2 Tools and methods.....	72
7.3 Modelling and implementation .....	75
7.4 Coupling the tools (WCF-Service) .....	78
7.5 Problem formulation from HeuristicLab.....	79
7.6 The rating process.....	81
7.7 Test cases.....	82
<b>8 Resume and Outlook .....</b>	<b>88</b>
8.1 The use of simulation in the industry – outlook .....	90
8.2 Outlook of the project .....	91
<b>Annex .....</b>	<b>92</b>
A. Simulation software survey.....	93
B. SiRO: Exemplary implementation of the dining philosophers problem in C# from Profactor.....	96
C. Queueing theory: Model notation.....	97
D. Basic statistics for simulation .....	98

E. Parameters of the OSGA.....	100
F. Submitted ARGESIM comparisons to the SNE Journal: C6 Emergency Department - Follow-up Treatment.....	101
<b>References .....</b>	<b>105</b>



## List of abbreviations

AHP	Analytic Hierarchy Process
CCR	Capacity-Constraining Resource
CR	Critical Ratio
CRM	Customer Relationship Management
CT	Cycle Time
DES	Discrete-event Simulation
EDD	Earliest Due Date
ERP	Enterprise Resource Planning
EUROSIS	European Multidisciplinary Society for Modelling and Simulation Technology
FIFO	First in First out
FoT	Factory of Tomorrow
GA(s)	Genetic Algorithm(s)
GPSS	General Purpose Simulation System
GUI	Graphical User Interface
HEAL	Heuristic and Evolutionary Algorithms Laboratory
i.i.d.	independent and identically distributed
LIFO	Last in First out
LP	Linear Programming
MES	Manufacturing Execution System
MRP-II	Manufacturing Resource Planning II
MST	Minimum Slack Time

OS	Offspring Selection
OSGA	Offspring Selection Genetic Algorithm
OvS	Optimization via Simulation (used as synonym of SBO)
PCS	Probability of Correct Selection
POM	Production and Operations Management
PPC	Production Planning and Control
PRNG	Pseudo Random Number Generator
RCPS	Resource Constraint Project Scheduling
RCPSP	Resource Constraint Project Scheduling Problem
RNG	Random Number Generator
R&S	Ranking and Selection
RSM	Response Surface Methodology
SA	Simulated Annealing
SBO	Simulation-based Optimization
SPO	Sample Path Optimization
SPT	Shortest Processing Time
TH	Throughput
TOC	Theory of Constraints
TS	Tabu Search
UI	User Interface
VDI	Association of German Engineers (German: Verein Deutscher Ingenieure)
WCF	Windows Communication Foundation
WIP	Work in Process
WSC	Winter Simulation Conference

## List of figures and tables

### Figures

<b>Figure 1.1:</b>	SBO using a rating framework to calculate the fitness value. ....	1
<b>Figure 2.1:</b>	Illustration of a manufacturing system. ....	5
<b>Figure 2.2:</b>	Reference views of the Aachener PPC-Model.....	6
<b>Figure 2.3:</b>	Structure of a PPC system. ....	7
<b>Figure 2.4:</b>	Main objectives of PPC systems. ....	10
<b>Figure 2.5:</b>	ABC analysis combined with XYZ analysis. ....	12
<b>Figure 3.1:</b>	A model is an abstract mapping of a system. ....	17
<b>Figure 3.2:</b>	Different ways of studying a system. ....	22
<b>Figure 3.3a:</b>	Steps 1 to 7 of a simulation study.....	24
<b>Figure 3.3b:</b>	Steps 8 to 12 of a simulation study.....	25
<b>Figure 3.4:</b>	Graphical representation in AnyLogic of the simple (M/M/1) simulation example. ....	26
<b>Figure 3.5:</b>	Queue length of clients waiting to be served. ....	28
<b>Figure 3.6:</b>	Time which a client spends in the bank. ....	28
<b>Figure 3.7:</b>	Statistical output of the server (bank assistant). ....	28
<b>Figure 4.1:</b>	Summary of simulation tools for production and logistics. ....	31
<b>Figure 4.2:</b>	Transformation of a uniform random number. ....	34
<b>Figure 4.3:</b>	Assigning random seeds in a consistent way. ....	39
<b>Figure 4.4:</b>	Dining philosophers.....	41
<b>Figure 4.5:</b>	SiRO implementation of the Simulation class. ....	42
<b>Figure 4.6:</b>	Illustration of the model components in the Plant Simulation. ....	45
<b>Figure 5.1:</b>	A multi-stage rating model to calculate the fitness value. ....	48
<b>Figure 5.2:</b>	Example values of a transfer function representing adherence to delivery dates. ....	51
<b>Figure 5.3:</b>	Classifying the indices according the significance in the fitness value. ....	52
<b>Figure 6.1:</b>	Mechanism of the sample-path optimization method.....	60
<b>Figure 6.2:</b>	Cycle of a simulation-based optimization study. ....	66
<b>Figure 6.3:</b>	Application areas for simulation-based optimization. ....	68

<b>Figure 7.1a:</b>	Illustration of different products I. ....	70
<b>Figure 7.1b:</b>	Illustration of different products II. ....	71
<b>Figure 7.2:</b>	Orders queuing to be processed. ....	73
<b>Figure 7.3:</b>	Example of a complex priority rule. ....	75
<b>Figure 7.4:</b>	Communication between simulator and optimizer. ....	79
<b>Figure 7.5:</b>	Grammar formulation of the priority rule. ....	80
<b>Figure 7.6:</b>	Exemplary rating model.....	81
<b>Figure 7.7:</b>	Establishing of a database connection. ....	82
<b>Figure 7.8:</b>	Screenshot of the Gantt chart representing the workers.....	83
<b>Figure 7.9:</b>	Animation of a simulation run. ....	84
<b>Figure 7.10:</b>	Screenshot of the logging during the study run. ....	84
<b>Figure 7.11:</b>	Prefix notation of the best found complex priority rule. ....	85
<b>Figure 7.12:</b>	Complex priority rule with decreased allowed complexity. ....	86
<b>Figure 8.1:</b>	General overview. ....	89
<b>Figure A.1:</b>	A handler is added to an event.....	96
<b>Figure A.2:</b>	The event handler method. ....	96
<b>Figure A.3:</b>	The Chopstick class. ....	96
<b>Figure A.4:</b>	Exponential distribution $\lambda = 0,5$ . ....	98
<b>Figure A.5:</b>	Triangular distribution.....	99
<b>Figure A.6:</b>	Binomial distribution. ....	99

## **Tables**

<b>Table 4.1:</b>	Overview of selected simulation tools with purpose and category. ....	36
<b>Table 4.2:</b>	Times needed to calculate the result. ....	46
<b>Table 5. 1:</b>	Possible index groups building. ....	50
<b>Table 6.1:</b>	Overview of optimization approaches used in SBO. ....	57
<b>Table 6.2:</b>	Optimization packages for SBO. ....	65
<b>Table A.1:</b>	Parameters of the OSGA. ....	100

*“La duda es uno de los  
nombres de la inteligencia”*

**Jorge Luis Borges**

# 1 Introduction

---

## 1.1 Motivation

Simulation is a tool extensively applied in operations research. Since the rapidly increasing computer power in the last decade, more problems for operative production control and scheduling have become feasible to solve. Many times there is unused potential of optimizing the processes of a factory. The processes can be improved using operations research techniques like simulation and optimization among others. One interesting approach is Simulation-based optimization (SBO) where an optimizer is coupled with a simulator. They can be coupled in different ways. However, the most common one is where the objective function of classical optimization is evaluated by a simulation. To do so, the simulation has to calculate a fitness value that is then provided to the optimizer. In order to calculate a fitness value a rating framework is used in this work. This rating framework merges, transfers and weights the relevant indices into a common, comparable unit. Figure 1.1 illustrates the interaction of that, using a rating framework in a SBO study.

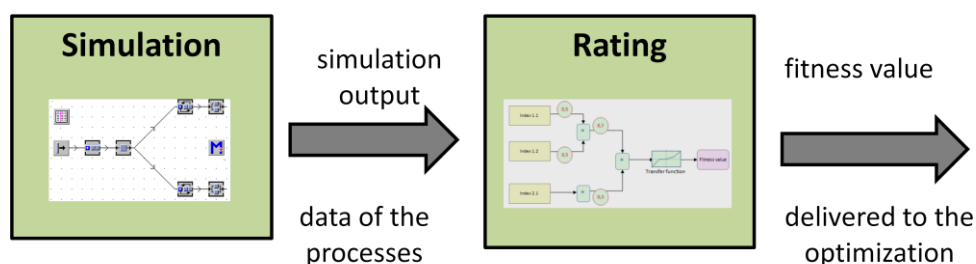


Figure 1.1: SBO using a rating framework to calculate the fitness value.

To perform a SBO study various optimization methods are established. Therefore, in recent years several optimizers (mainly commercial) that can be coupled with simulation were developed. These include optimizer like OptQuest<sup>1</sup>, ISSOP<sup>2</sup>, ILOG<sup>3</sup>, AutoStat<sup>4</sup> and HeuristicLab<sup>5</sup>. Most of times in commercial software packages metaheuristic techniques are used to perform the optimization. However, the research of optimization methods in SBO focuses more on the stochastic nature of discrete-event simulation (DES) [13].

There is a broad variety of literature in the area of SBO. Very interesting reviews on this topic were published by Fu [13], [14]. Moreover, Banks gives in his book “Handbook of Simulation” a good overview about SBO [15].

There are several application areas of SBO, one of this can be found in the field of factory simulation. Factory simulation can be used in different stages of the processes of a factory. Involving the planning, the implementation and the operative production phase [16]. Therefore, simulation is done to obtain knowledge of the complex interaction between the individual objects, which forms the field of study. In addition, the impacts of *what – if* questions can be shown easily by the means of simulation.

It is interesting to note that the optimization of factory processes is quite diverse in the different phases. For production planning and control (PPC), the main optimization target is based on job sequencing and determining optimal lot sizes. This lot sizing and scheduling is done in order to fulfil the company objectives, but on the other hand it influences other aspects, which are often not tangible directly, including:

- Economical aspects like resource utilization, stocks and inventory, timeliness.
- Work ergonomics and work psychological aspects of workers like stress levels, work monotony, pause planning, etc.
- Emission aspects like waste of raw materials, additional cleaning, heating or cooling.

In most production systems, several operations are of stochastic nature. These can be machine failures, product rejections rates or just stochastic process and set-up times, for instance. If the system to be analyzed is of a certain complexity, containing interactions between the system elements, simulation is an appropriate tool to use.

---

<sup>1</sup> OptQuest: [www.opttek.com/Products/OptQuest.html](http://www.opttek.com/Products/OptQuest.html)

<sup>2</sup> ISSOP: [www.dualis-it.de/issop](http://www.dualis-it.de/issop)

<sup>3</sup> ILOG: [www-01.ibm.com/software/websphere/ilog/](http://www-01.ibm.com/software/websphere/ilog/)

<sup>4</sup> AutoStat: [www.autosim.com](http://www.autosim.com)

<sup>5</sup> HeuristicLab: <http://dev.heuristiclab.com>

## 1.2 Problem description of a performed SBO study

During the development of this work a SBO study was performed within a project team. The project (project name: Procomposite) was founded by the Austrian Research Promotion Agency (FFG<sup>6</sup>) within the program line Factory of Tomorrow (FoT<sup>7</sup>). The field of study is the ASMA GmbH located in Weitra, Austria. This company is working in the area of plastic processing. The main production is based on plastic casting. The operative production is quite flexible at the moment and in many circumstances done by equipment operators themselves, based on the available information (e.g., priorities, due dates, order specifications, and so on). Just several basic indices (e.g., the turnover per productive working hour as well as the deviation of the order production times) are taken into account. In the performed SBO study the optimization approach is mainly based on order sequencing and lot sizes.

## 1.3 Contents preview

This work is organized as follows: The theoretical background and all necessary tools are presented at the beginning of this work. Chapter 2 introduces the basic aspects of PPC together with production management. At the end of this chapter some theories and techniques related to PPC are presented. In chapter 3 several general definitions are given to understand what is meant by modelling and simulation. The advantages and disadvantages, as well as, the categories of simulation are explained. Ending this chapter with a simple simulation example. The following chapter 4 provides an overview of state of the art simulation software focusing on DES software. In addition, software is classified and compared. A rating framework to evaluate the simulation output is presented in chapter 5 of this work. What is meant by rating, the theoretical background and a rating framework is shown. Chapter 6 treats how simulation is coupled with optimization. The used optimization methods are presented together with optimizers in SBO. The interaction between simulation, rating and optimization is explained and several applications of SBO are listed. In chapter 7 a performed SBO study from the plastic industry is presented in a detailed way. The development of the simulation and rating model is described. Beginning from data analysis to database connection, the implementation and finally results are presented. Further this section describes the optimization problem formulation using the HeuristicLab framework. The final chapter 8 gives a summary and an outlook.

---

<sup>6</sup> The Austrian Research Promotion Agency (FFG): <http://www.ffg.at/>

<sup>7</sup> Factory of Tomorrow: <http://www.fabrikderzukunft.at/index.htm>

*“Being busy does not always mean real work. The object of all work is production or accomplishment and to either of these ends there must be forethought, system, planning, intelligence, and honest purpose, as well as perspiration. Seeming to do is not doing.”*

**Thomas A. Edison**

## 2 Production Planning and Control

---

### 2.1 Manufacturing system

Manufacture has its origin from the two Latin words “*manus*” which means hand and “*factus*” which means make [17]. This word has changed from its original meanings of “hand made” to a modern use. A definition of that is:

“A manufacturing system is an objective-oriented network of processes through which entities flow” [18].

This definition is very rich, containing several important terms, which will be found all over the work again. Thus, a manufacturing system has an objective. In most cases this is related to economic (money) concerns. However, other factors might influence too, as we will see in this work. A manufacturing system is divided into processes that form a network. Entities represent all the information, as well as the manufacturing parts. The flow describes how the entities pass through the networks and interact together. Figure 2.1 illustrates a manufacturing system.

Technical understanding and expertise are needed in order to make decisions about the design and operation of manufacturing systems. The ability to fulfil the objectives is required as well. Consequently, different disciplines are needed in a manufacturing system. Thus, mechanical engineering designs the products, industrial engineering



handles concerns about the flow of the entities in the network and the management is responsible for the objective and goals [18].

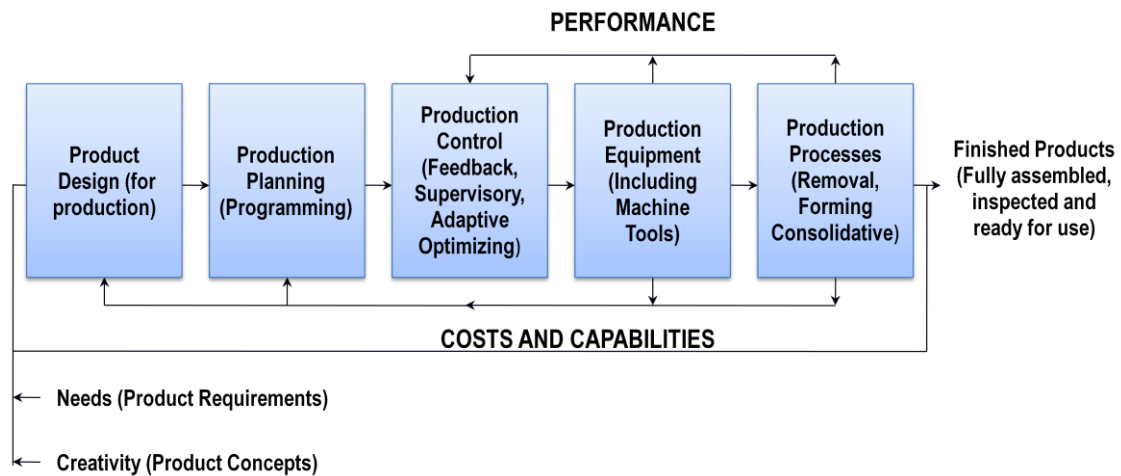


Figure 2.1: Illustration of a manufacturing system, according to Chryssolouris [5].

## 2.2 Planning and control

### *Aachener PPC-Model*

The Aachener PPC-Model was developed by the Research Institute for Industrial Management (FIR) at the RWTH Aachen University<sup>8</sup>. The FIR has years of experience and has been doing investigation in the field of PPC. In the German area the Aachener PPC-Model is widely recognized. This model was developed in order to provide an efficient support of complex order process and at the same time the development of PPC systems. The Aachener PPC-Model distinguishes 4 different reference views of PPC (the task reference view, the process architecture reference view, the function reference view and the process reference view). These four views and their interactions can be observed in Figure 2.2.

The following explanations of PPC are based on the Aachener PPC-Model. PPC systems can be divided into the two big branches, namely production planning and production control. Figure 2.3 shows the distinction with the associate steps. The 6 illustrated steps in figure 2.3 are passed through successively.

<sup>8</sup> FIR: Research Institute for Industrial Management (German: Forschungsinstituts für Rationalisierung) at the RWTH Aachen University. <http://www.fir.rwth-aachen.de/>

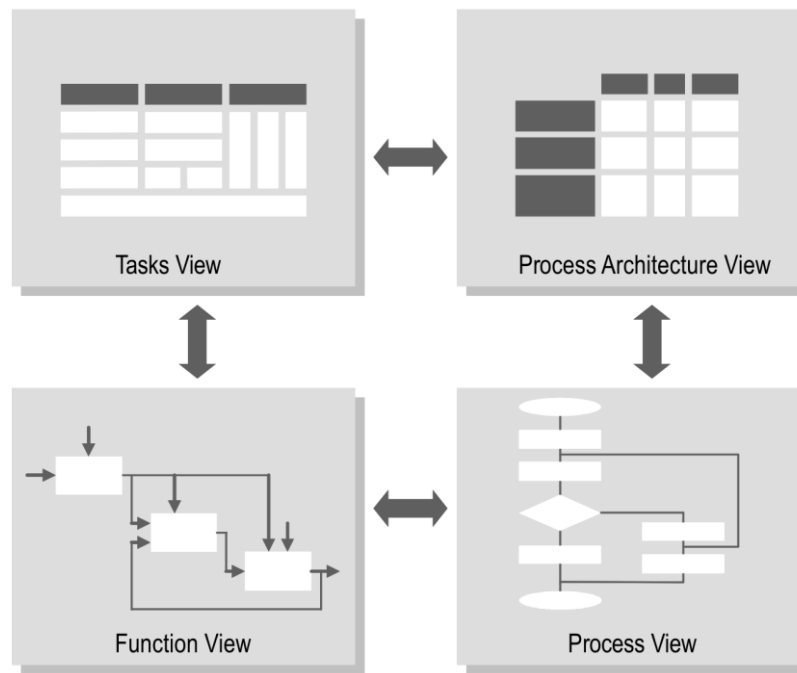


Figure 2.2: Reference views of the Aachener PPC-Model, according to Schuh [6].

### 2.2.1 Production planning

Production planning includes questions like what product, in what quantity and when it should be produced [17], [19], [11]. As shown in figure 2.3 production planning consists of three steps.

#### Production program planning

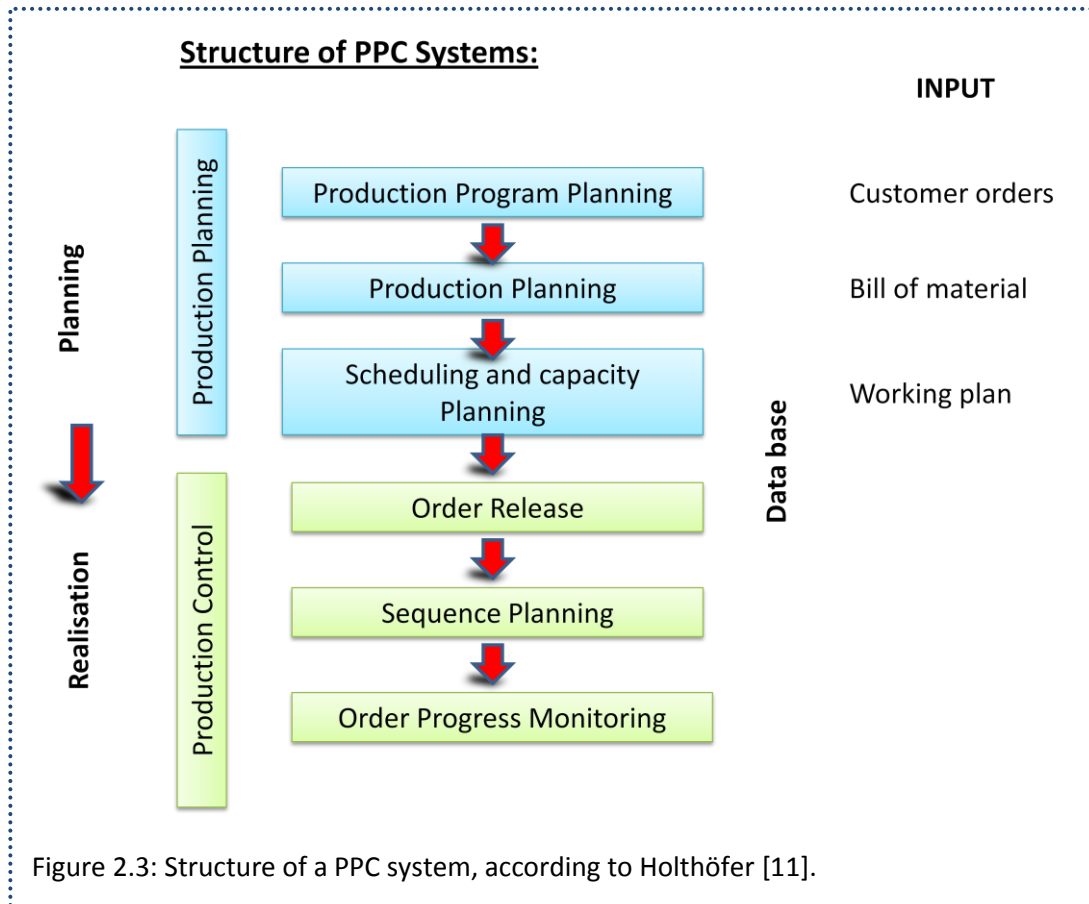
In the production program planning is determined what products have to be produced in a period of time. The primary demand is calculated according the existing stocks, the estimated disposal and the minimum level of inventory. The result of this step is a *master production schedule* [11].

#### Quantity planning

The task of the quantity planning is to determine the necessary quantities of material and resources to fulfil the master production schedule [19]. By using bills of material the exact quantities can be determined.

### Scheduling and capacity planning

With throughput scheduling, a plan start date is defined. In the following, the required capacity is calculated for each resource [19].



Furthermore, production planning can be divided according to the time range in a strategic, tactic and operative planning [20].

### Strategic (long-range) planning

The main task of strategic planning is to create and maintain competitive products. The time frame of strategic planning is years.

Scenario techniques are used to predict the influence of different input dimensions, which may influence the strategic planning. The aim is to simulate the possible scenarios that may occur in the future [21].

Possible input dimensions of different scenarios may be of general nature or more related to a company. They may include elements like:

- Global economic development
- Sales estimations
- Costs of any kind (material, labour, machines, energy and so on)
- Prices of the products
- Production layout changes (number of machines and workers, qualifications and so on)

#### Tactic (medium-range) planning

The main tasks of tactic planning are decisions concerning the products, the resources capacity, as well as the personal capacity. Further, the organisation of the production is part of tactic planning. The time frame is from month to years.

#### Operative (short-range) planning

Based on the underlying strategic and tactic planning, the operative planning uses the available resources to fulfil them. The time frame here is from hours to weeks.

### **2.2.2 Production control**

Production control takes the inputs from the production planning. Production control handles concerns about the available resources to execute the plan [17]. Thus, we step from the planning phases to the realisation phases. Production control is divided into order release, sequence planning and order progress monitoring.

#### Order release

In this stage the orders are released to be produced by assigning them to the corresponding working stations, where they wait to be processed.

In this stage also the lot sizes are determined. The calculation of the lot sizes includes several questions like the set-up costs and inventory costs. In general the objective is to minimize these costs [19].

#### Sequence planning

After the order release, the production sequence of the orders has to be determined. This can be done in different ways [19]:

### *Control stations*

The assignments of the working steps are visualized commonly by Gantt charts. A production planner can change the assigned working steps interactively.

### *Priority rules*

There exist various types of priority rules. The easiest is the first in first out (FIFO), where the processing sequence is in that way as the orders arrive. Other priority rules are for instance: last in first out (LIFO), longest waiting time and highest capital commitment. According to Holthöfer [11] simulation is a good tool to verify the impacts and consequences of different priority rules. Chapter 7 gives a more detailed view on priority rules.

### *Algorithmic optimization*

A more complex way to obtain the production sequence makes use of optimization algorithms. Therefore, an objective function is formulated, which has to be optimized. Often (meta)heuristic techniques, like search or evolutionary and genetic algorithms are used to solve more complex problems. It is also possible that algorithmic optimization is coupled with simulation. In this work it is shown how complex priority rules can be obtained using genetic algorithms.

### Order progress monitoring

In this stage the nominal data are compared to the actual data. This is done through operating data logging, where the relevant information is recorded. If the deviation between nominal and actual data is too big, it might be necessary to re-plan the operations [11].

## **2.2.3 Objectives of production planning and control**

Customers and companies have different objectives. These objectives also have changed during the last decades. In the past, the efforts were concerned to high utilization of resources. Nowadays, the efforts have been shifted typically to concerns like short cycle times, low work in process (WIP) and high adherence to delivery dates.

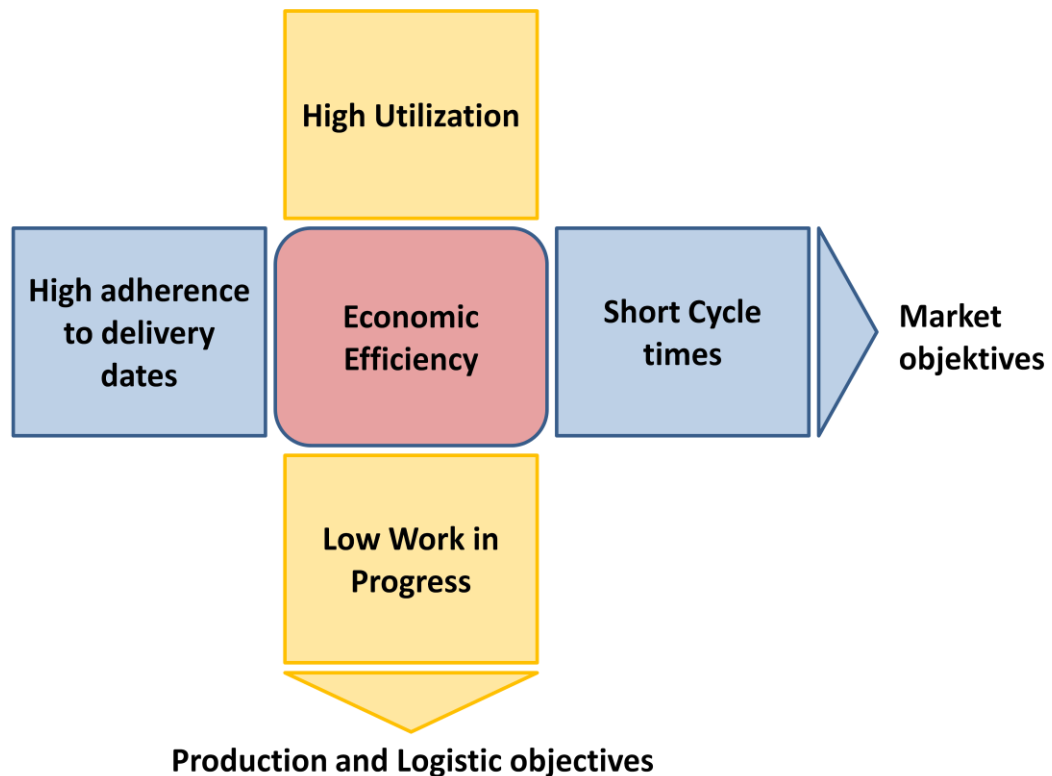


Figure 2.4: Main objectives of PPC systems [9], [12].

From the customers view (market objectives), the ordered products should be delivered at the desired date. The delivery time should be short as well. From the companies view (production and logistic objectives), the WIP should be low in order to gain short cycle times and low capital commitment. A further objective of companies is that the utilization rate of the resources is high enough [12].

Many times all these objectives cannot be fulfilled simultaneously because they are conflictive. As a consequence, often the WIP is high or the productivity is low. As in most manufacturing companies the staff and resource utilizations have priority; thus, the companies have taken the decision of having high levels of WIP. Furthermore, the trend to small lot sizes can be found [9]. To solve these conflicts, techniques of operative production control and scheduling are often used. Due to that in most cases the optimization problem is of NP-hard<sup>9</sup> nature, it cannot be solved using exact analytical techniques. One common solution to solve this and to obtain operative production plans are heuristics.

<sup>9</sup> NP-hard: "The complexity class of decision problems that are intrinsically harder than those that can be solved by a nondeterministic Turing machine in polynomial time. When a decision version of a combinatorial optimization problem is proved to belong the class of NP-complete problems, then the optimization version is NP-hard" Source: US National Institute of Standards and Technology

### Little's Law – Definitions:

In order to understand well the concept of PPC, several definitions are needed. Very important is the relation between WIP, cycle time (CT) and throughput (TH), which is recognized as the Little's Law<sup>10</sup> [18]:

$$WIP = TH \times CT \quad (2.1)$$

This law also implies that a cycle time reduction can be achieved by reducing the WIP or by increasing the TH.

$$CT = \frac{WIP}{TH} \quad (2.2)$$

As in Little's Law no other factors are included than WIP, CT and TH, this leads to that higher WIP together with higher CT will result in the same throughput level.

$$TH = \frac{WIP}{CT} \quad (2.3)$$

## **2.3 Difference between scheduling and dispatching**

Dispatching and dispatching algorithms are also commonly called “scheduling” and “scheduling algorithms”. The similar use of these terminologies is unnecessarily confusing. Hence, it is important to have in mind the distinction of them [22].

Scheduling: By using scheduling it is determined in advance when and which job will use which factory resources. If a manufacturing system is deterministic, scheduling has advantages over dispatching.

Dispatching: By using dispatching it is determined how to use factory resources just if a resource is available. Thus, when a resource is available a situational dynamic selection from a pool of available jobs is performed. If a manufacturing system is not deterministic, then dispatching has advantage.

---

<sup>10</sup> Little's Law: Named after John D. C. Little. He gave the mathematical proof of that.

## 2.4 Trends in production management, theories and techniques

There exist different planning and control methods. Jodlbauer proposes a combination of the ABC analysis<sup>11</sup> with the classical XYZ analysis<sup>12</sup> to determine which planning and control method is the appropriate [4]. The combination of them can be seen as a matrix in figure 2.5. In the following several of these methods are presented.

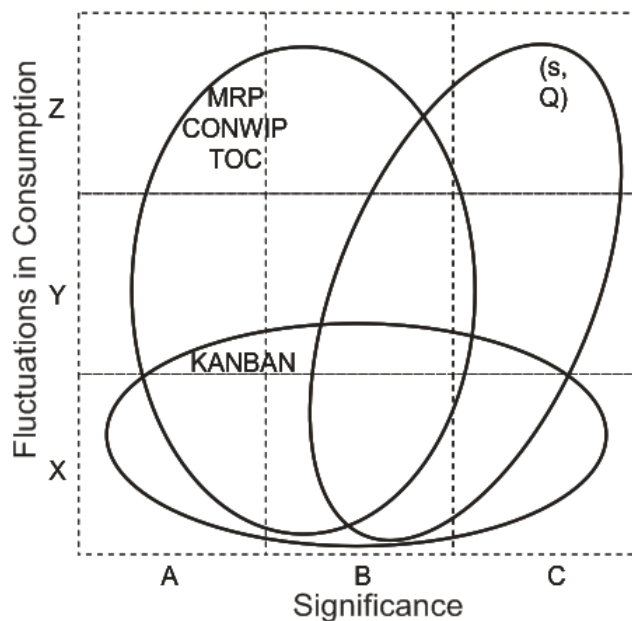


Figure 2.5: ABC analysis combined with XYZ analysis [4].

### Information systems: MRP / MRP-II / ERP and MES

McKay and Wires [23] give a good overview of information systems used in the industry. Information systems are basically used to handle the huge amount of data that are needed for production control affairs. This data includes information about products, suppliers, customers, maintenance, inventory levels and historical data among others.

Material requirements planning (MRP) and its extension, manufacturing resource planning-II (MRP-II), handle these data. In MRP-II are included also functions that relate resources directly to the production process. If MRP-II is further extended with more

<sup>11</sup> The ABC analysis goes back to Pareto and is similar to the pareto principle. Three classes namely A,B and C are made. The A parts class represents 18% of the products but 80% of the value. The B parts class represents 26% of the products and 15% of the value. And finally the C parts represent 56% of the products and represents just 5% of the value.

<sup>12</sup> The XYZ analysis classifies the fluctuations in consumption during an annual period. X parts are those who have marginal annual fluctuations. Z parts are those who have high annual fluctuations and Y parts are between them.



features for instance human resource management, we speak now of enterprise resource planning (ERP). On the market several ERP products are available. However, SAP<sup>13</sup> is the market leader.

To an ERP system, a manufacturing execution system (MES) may be coupled. MES manage the production processes in the factory. In MES, scheduling tools are often integrated.

### KANBAN

KANBAN is an adequate tool when the annual fluctuation in consumption is marginal (compare figure 2.5). KANBAN goes back to the beginning of the Toyota production system. The word *kanban*, coming from the Japanese, can be translated as signboard. The KANBAN system was developed by Taiichi Onho. He gained to minimize the WIP as well as to reduce inventory costs. KANBAN makes use of visual signals with two basic principles behind this system, which are:

- Just to produce products that need to be replaced because they are sold.
- Just to produce products that are demanded by the customers [24].

### Theory of Constraints

The Theory of Constraints (TOC) was developed by Eliyahu M. Goldratt<sup>14</sup>. Goldratt described it for the first time in his business novel *The Goal* (1984) [25]. The basic idea of TOC is that in any manufacturing process the capacity is limited by its capacity-constraining resources (CCR) or simple constraints [19], [26]. Therefore, the throughput of a system cannot be higher than its weakest or limiting part, the constraint. However other process parts could produce faster.

The TOC consists of a five-step process with main consideration on the constraint(s). These five steps developed by Goldratt are [25]:

- I. Identify the constraint or the constraints of the system.
- II. Use all the capacity of the constraint(s).
- III. Subordinate all other processes above the constraint(s).
- IV. Increase the capacity of the constraint(s) in order to increase the throughput.
- V. Continuous improvement of the system.

---

<sup>13</sup> SAP is a product developed by SAP AG, Germany: [www.sap.com](http://www.sap.com)

<sup>14</sup> Eliyahu M. Goldratt (1947 – 2011) was an Israeli physicist. He developed the TOC and other derived tools from it. Further he wrote several books as business novels in management science like *The Goal* (1984), *It's Not Luck* (1994), *Critical Chain* (1997) and *Necessary But Not Sufficient* (2000).

### *I. Identify the system constraint(s)*

All improvements should start with the weakest link, because this brings the best opportunity of improvement. The constraints in a producing system include flow problems, quality problems, capacity problems or policy problems [27].

Through capacity requirements calculation of the resources the capacity constraint(s) can be identified. Thus, the constraint(s) are the resources with the highest calculated utilizations [19]. For relative small problems the constraint(s) also can be easily identified with linear programming (LP)<sup>15</sup> [26].

### *II. Use all the capacity of the constraint(s)*

In this step the focus is on the constraint(s). The throughput that flows through the constraint(s) has to be increased. If the constraint is in a manufacturing process that limits the output of the entire process, it is a so called bottleneck [27]. Further, Levinson writes that unused capacity of constraints is lost forever and the opportunity to sell can never be regained [26].

### *III. Subordinate all other processes above the constraint(s)*

All non-constraints or non-bottlenecks should produce at the same speed as the constrained resources do [27]. If the non-constraints produce slower, then the constraints capacity is unused (what this implies was already discussed in the step before). On the other hand, if the non constraint(s) produce faster than the constraint(s) the stock and the WIP increase.

### *IV. Increase the capacity of the constraint(s) in order to increase the throughput*

If necessary, additional capacity to the constraint(s) have to be assigned. This can be for instance extra work hours, extra shifts, more personal or more machines.

### *V. Continuous improvement of the system*

After having implemented the steps from 1 to 4 the constraint(s) might be eliminated. But also new constraint(s) can be occurred. Therefore, the system has to be check continuously of new constraints(s) by starting the iteration at step 1 again.

---

<sup>15</sup> In linear programming an objective function  $Z$  has to be *maximized* or *minimized*. The constraints of all the resources are represented by linear equations. The marginal benefits represent the capacity – constraints and on the other hand if a resource does not have marginal benefit it's a non constraint.

In The Goal [25] Goldratt describes the balanced capacity production line. With an example using matchsticks and dice he shows that random variation reduces the throughput of a system. On the other hand the inventory increases. This is because the theoretical rate cannot be reached. This matchsticks and dice example can be simulated. A simulation implementation of this is provided by Levinson Productivity Systems [28].

*“A system is a big black box  
Of which we can't unlock the locks,  
And all we can find out about  
Is what comes in and what goes out  
Perceiving input-output pairs,  
Connected by parameters,  
Permits us, sometimes, to relate  
An input, output and a state.  
If this relation's good and stable  
Then predict we may be able,  
But if this fails us - heaven forbid!  
We'll be compelled to force the lid!”*  
**Kenneth Boulding**

## 3 Modelling and Simulation

---

### 3.1 Background

#### 3.1.1 System

The word system origins from the Greek *systema*; *systema* comes from *synistanai* which combines *syn* meaning “with or together”, and *histanai* which means “to set”. Moreover, Law and Kelton give a clear definition of a system [1]:

“A system is defined to be a collection of entities, e.g., people or machines, that act and interact together toward the accomplishment of some logical end.”

It is possible to find several definitions of system, although in all of them the fundamental term is *interaction*. As shown in figure 3.1, a system is characterized through its system boundaries, the input and output, the system elements with their logical flows, states and transitions.

### 3.1.2 Model

The philosopher Klaus Dieter Wüsteneck has provided a general definition of the model concept. His German original formulation from the 60s is the following:

*“Ein Modell ist ein System, das als Repräsentant eines komplizierten Originals auf Grund mit diesem gemeinsamer, für eine bestimmte Aufgabe wesentlicher Eigenschaften von einem dritten System benutzt, ausgewählt oder geschaffen wird, um letzterem die Erfassung oder Beherrschung des Originals zu ermöglichen oder zu erleichtern, beziehungsweise um es zu ersetzen.”*

Its English translation is the following:

“A model is a system that is used, selected, or produced by a third system to enable the understanding of or the command over the original system, or to replace the original system. Model system and original system share certain characteristics that are of relevance to the task at hand.”<sup>16</sup>

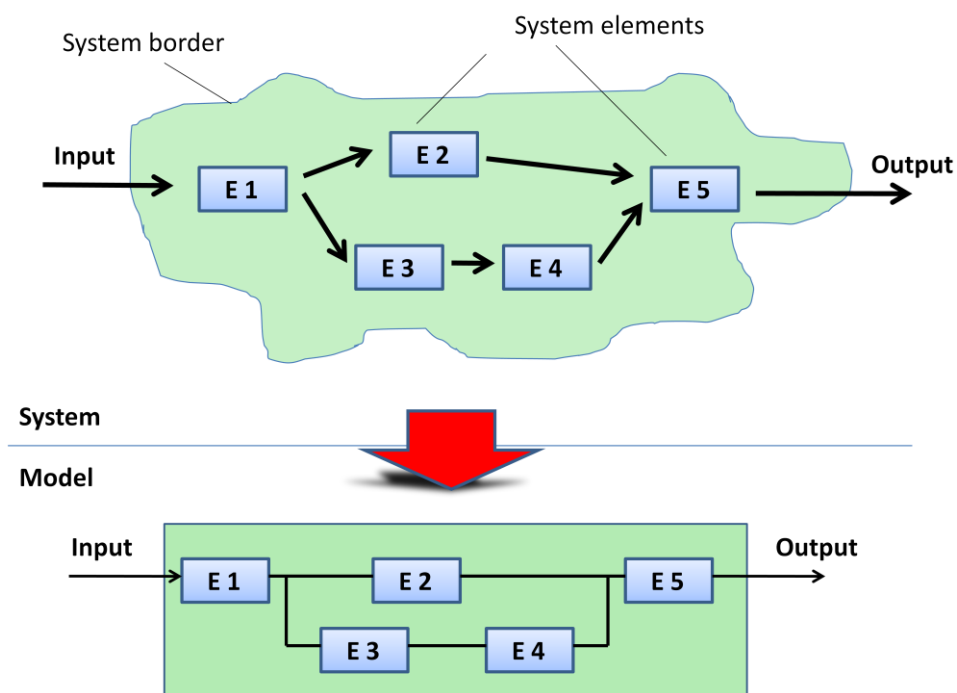


Figure 3.1: A model is an abstract mapping of a system. This figure was modified from Kühn [3].

<sup>16</sup> English translation from [29] W. Marquardt, et al., "Overview on OntoCAPE," *OntoCAPE*, 2010.

A general model theory [30], which has been widely recognized, was introduced in 1973 by Herbert Stachowiak<sup>17</sup>. According to Stachowiak, this model theory is not limited to the application of one specific discipline and is defined by three characteristics:

#### Mapping

A model is a mapping of the reality, a representation of many objects, which may be models themselves.

#### Contraction

A model is a contraction of the reality without gathering all attributes of the reality. It includes those which might be relevant to the modeller according to the modelling purpose.

#### Pragmatism

A model is pragmatic within a certain time span and a specific purpose. The model is geared to the purpose to answer a question. It might also be seen as the interpretation of the model.

Kühn further proposes that the perception of reality is not objective [3]. As far as different observers can interpret the system differently and consequently model different models of the same system. Furthermore, it has to be mentioned that modelling has limits.

### **3.2 Simulation**

A lot of times the term simulation is used without understanding its meaning completely. Simulation (from its Latin origin *simulare*) means to imitate, clone, or reproduce the operation of a system. In general, if one speaks about simulation people think about the behaviour of a car or an airplane which are simulated in a wind tunnel, or a traffic simulation where the cars drive around. People also do simulation daily by themselves to organize their lives. For example someone in the morning goes through the daily tasks and simulates them in different ways.

---

<sup>17</sup> Herbert Stachowiak (1921 – 2004).

A formal definition of simulation is given by The Association of German Engineers (German abbreviation: VDI) Guideline 3633<sup>18</sup>, [31]:

“Simulation is the preproduction of a system with his dynamic processes in a model which allows experiments, to get expertises which are transferable to the reality”

In addition, Shannon has presented another formal definition of simulation [32] which is also widely used:

“Simulation is the process of designing a model of a real system and conducting experiments with this model for the purpose either of understanding the system or of evaluating various strategies (within the limits imposed by a criterion or set of criteria) for the operation of the system.”

In literature more definitions of simulation can be found, to mention them all here is not considered reasonable. Simulation is not limited to the use of computers. However, in this work simulation is referred to as computer-aided simulation.

### **3.2.1 Concepts related to modelling and simulation**

#### Activity

An activity is a set of operations, which are carried out at one point in time. An activity starts and ends with an event.

#### Process

A process is a sequence of activities belonging together over a period of time. This is for instance a manufacturing process.

#### Simulation run

A simulation run or replication is the execution of a simulation model, with the aim to simulate the behaviour of a system over time.

---

<sup>18</sup> The VDI Guideline 3633 contains 12 parts and is entitled “Simulation of systems in materials handling, logistics and production”.

### **3.2.2 Why do we use simulation?**

In general we use simulation when a system tends to be too complex. Moreover, many times an analytical solution cannot be found. It is possible that the system states changes over time, that there are interactions between systems or that the system (or part of it) is of stochastic nature. In these cases the use of simulation would be appropriate.

Simulation has many applications in different areas. Just to name a few of them: health care, manufacturing applications, supply chain management, transportation, military applications, business process modelling, and risk analysis among others.

#### **3.2.2.1 Advantages of simulation**

Simulation has a lot of strength and advantages. The most important are:

- When a stochastic system which is too complex to be analyzed satisfactorily by the kinds of mathematical models, the simulation appears as useful tool [33].
- The ability of visualizing a model of a studied system brings advantages, allowing a quick overview of the processes. This, can often impact positively clients or management during the model presentation. The visualization may either be in 2D or 3D.
- It is possible to predict events that underlie randomness by using random numbers.
- Using simulation brings advantages because it is often not possible to perform experiments with the real system. Simulation can be used to perform experiments. These experiments can be done in an already operating system; in a system that is under construction or in the planning stage.
- Layout analysis or so called “what – if questions”, where system configurations and parameters are changed, can be done easily. This can also be done in the planning phase in order to analyse the behaviour of possible future layouts.
- Simulation is not restricted to real time. In simulation a long period (even years) can be simulated in a short time. The simulation time can be accelerated or slowed down if necessary.

#### **3.2.2.2 Disadvantages of simulation**

Besides the advantages, simulation has disadvantages as well. Some of them are:

- Simulation is very time consuming and often expensive. If the same result can be obtained using an analytic model, simulation should not be used.



- The output of stochastic models is not an exact solution. As in most simulation models, which represent some random input the obtained output is also random itself. Therefore the result has to be interpreted, considering that this result underlies stochastic aspects. To minimize this randomness, a simulation model has to be replicated  $n$  times.
- An important question in SBO studies is to determine the number of simulation replications. Law and Kelton write that this number of replications should depend on variance estimation. To do so, a few initial replications should be performed. Consequently, more replications are needed for higher variances [1].
- If a model does not represent the system in a valid way, the output will not be valid neither. Therefore, an incorrect model will lead to the failure of the simulation study.
- Often the results of a simulation study are not questioned.

### **3.2.3 Categories of simulation**

#### **Discrete-event simulation**

In DES the system state changes occur instantaneously at random points in time. This state changes are a result of the occurrence of discrete events [33].

Some examples for DES are:

- The operations of a bank where clients arrive with a certain frequency to be attended by a teller.
- A manufacturing system where merchandise is produced.

The events in these cases would be for instance the arrival of clients or merchandise, the service of an operator, unpredictable or planned interruption and so on.

#### **Continuous simulation**

When the state changes of the system occur continuously over time the simulation is continuous [33]. For example, a belt conveyer moving products or an airplane flying continuously over time. System dynamics<sup>19</sup> or physical applications are often described by continuous simulation to understand the behaviour of a system over time. Continuous simulation tends to be relatively complex. It is often possible to approximate the behaviour

---

<sup>19</sup> System Dynamics was introduced by Jav W. Forrester in the 1950s.

of a continuous system by using DES. E.g., a belt conveyer is divided into finite parts, where the state changes now occurs in a discrete way. To simulate continuous systems often mathematical software like Mathematica, Matlab or Dymola are used.

### Hybrid simulation

Hybrid simulation is a mixed continuous and DES where the state changes occur not only at a specific point in time but continuously over time as well. An example for hybrid simulation could be an oil tanker terminal, where the tankers arrive at specific points in time and the oil flows continuously.

## 3.3 Classification of models: Different ways of studying a system

Figure 3.2 shows how to study a system experimentally in different ways. First of all there is the possibility to experiment with the actual system. In most cases this is not feasible to do. This is because experiments are often too expensive and complex, the system does not exist yet or there is just no way to effect experiments with the actual system. In all of these circumstances, a model of the system is needed to perform the required experiments.

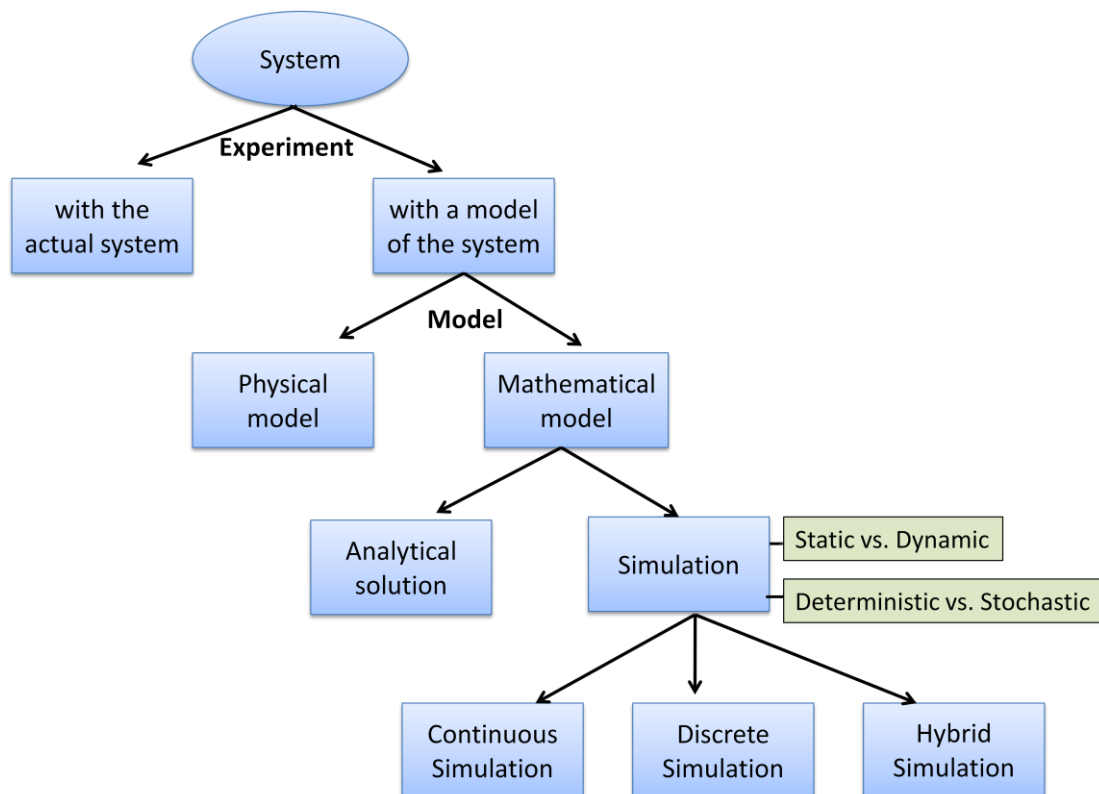


Figure 3.2: Different ways of studying a system. Modified and extended from Law and Kelton [1].

These models can either be physical models or mathematical models. Physical models are those that represent the reality by something tangible, constructed in scale or have at least an analogue behaviour than the reality. According to Law and Kelton physical models are also called iconic models [1]. On the other hand, mathematical models represent the reality in an abstract way by diverse manners. In static models, the state of systems is invariable over time. In contrast, in dynamic models the state of system changes over time. If the model solution is unique and always the same predetermined one, the model is deterministic. When there are random elements in a system the model is of stochastic nature.

### 3.4 Simulation study

An outline of a major simulation study is given. Similar descriptions of the steps of a simulation study can be found in current literature, however some minor differences are discussed (compare Law and Kelton [1], Hiller and Lieberman [33], VDI Guideline 3633 [31], and Banks, Carson, Nelson and Nicol [2]). In the following, the steps of a simulation study according to Banks *et al.* are presented [2]. They recommend a procedure to perform a successful simulation study, which includes twelve steps. These twelve steps are illustrated in figure 3.3a and 3.3b. Sometimes it may be necessary to move back to previous steps and perform them again.

- (1) **Problem formulation.** The simulation team together with the client must first formulate the questions to be studied. If this formulation is done by just one part it is necessary to ensure that problem is well understood by the other part. In some cases after a simulation study has been started it might be necessary to reformulate the problem due to new circumstances.
- (2) **Setting of objectives and overall project plan.** Banks *et al.* [2] also call this step “prepare a proposal” and recommend an additional external or internal consultant. Here the objectives of the simulation study have to be defined. According to the aims of each company and their simulation goals the simulation objectives are different in each simulation study. In the project plan, a time framework, as well as the required resources (e.g., personal and technologies) have to be defined.
- (3) **Model conceptualization.** A concept of the system to be studied is needed for the modelling process. It is recommended to begin modelling first in a simple way and after that, the complexity can be raised. Furthermore, it has been proposed to check the accuracy of the model by walking through the established conceptual model [33].

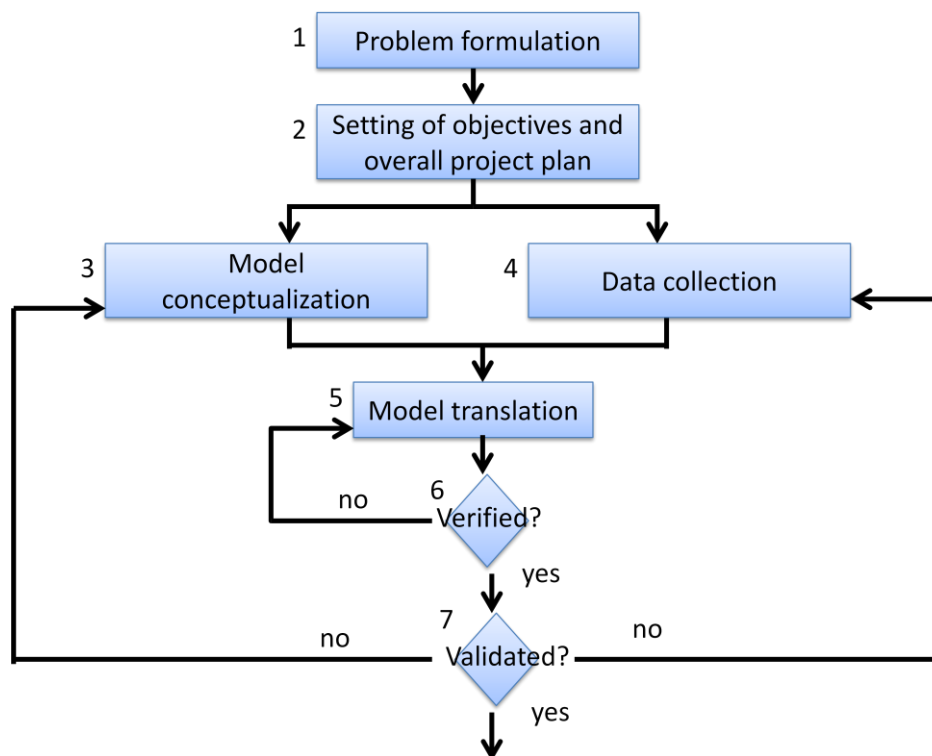


Figure 3.3a: Steps 1 to 7 of a simulation study, according to Banks [2].

- (4) **Data collection.** Data collection and the quality of the collected data are essential for a successful simulation study. The VDI Guideline 3633 [31] classifies this data into groups, which are: system data, organisational data, and technical data. System data includes working plans, spare parts lists, due times and order quantities. Organisational data includes work time organisation, resources and process organisation. Technical data includes factory structures, production data, capacities and failure rates, among others.

A common problem related to data collection is described by Banks. It usually happens that clients promise that the required data for the simulation study is available. Although, after delivering the data, it comes out that this data does not fit the requirements and is quite different from the promised one. An illustrative example of this has been reported for an airline, where the company have said “we have every bit of data that you want over the last five years.” However, after the data was handed in, it was found out that this data was just an average of “talk time” by year of each employee of the call centre [34].

- (5) **Model translation.** Based on step three, where the conceptual model was defined, and step four, where the data was collected, the data has to be brought

into a hard coded computer program. In this step an appropriate simulation tool has to be selected. Chapter 4 of this work introduces into selecting simulation tools in a more detailed way.

- (6) **Verified?**. After performing the hard coded model it has to be debugged. This is done to see if the model operates in an appropriate way. If necessary rollback to the model translation and the verification has to be repeated. Law and Kelton describe eight different verification techniques of computer simulation programs [1].
- (7) **Validated?**. After having the conceptual model translated into a computer program and debugged, it has to be proven whether the model represents the real world system in a reliable way. There are several different validation techniques [1], [33].
- (8) **Experimental design**. In this step decisions concerning the system configuration of the simulation have to be made. Those are the length of each simulation run, the initial state of the system, if a warm up period is required and the number of different simulation runs [1], [33].

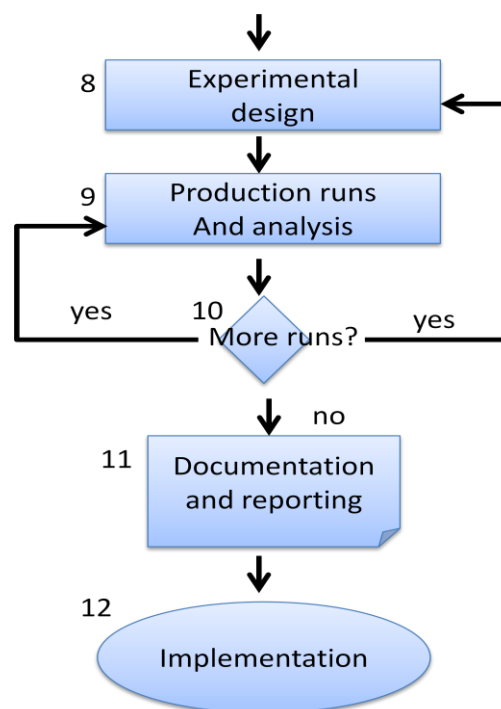


Figure 3.3b: Steps 8 to 12 of a simulation study, according to Banks [2].

- (9) **Production runs and analysis**. The simulation runs have to be made according to the experimental design. Having the result of each simulation replication they

need to be analysed. Different system configurations of different simulation runs have to be compared to each other.

- (10) **More runs?**. Here, it is analyzed and determined whether more simulation runs are necessary. If yes, it steps back.
- (11) **Documentation and reporting**. As in other studies and projects, documentation is necessary for various reasons. Further, the results of the simulation study have to be presented to the executive management and the involved personal of the client company.
- (12) **Implementation**. Having performed the eleven steps of a simulation study and provided the results, the simulation has to be implemented in the operative system.

### 3.5 A simple simulation example

To illustrate the processes of a simulation study, a simple exemplary system of study is presented. This system contains the four basic elements of DES:

A **source** where the entities are created and enter the system.

A **queue** where the entities queue to be processed.

A **server** where the entities are processed.

A **sink** where the entities are terminated and leave the system.

Such a so called single-server queuing system (Source – Queue – Server – Sink) can represent for instance a production line (where the products have to pass through a machine) or clients of a bank (who are served by bank assistances). This system may be either deterministic or of stochastic nature. The queuing theory provides notations to describe this. E.g., (M/M/1) represents a Markovian (stochastic) single-server queuing system. In annex of this work more details on queuing theory notations can be found.

This exemplary simulation study is done using the AnyLogic simulator from XJ Technologies<sup>20</sup>. For this small simulation study the case of the above mentioned bank system is presented. AnyLogic provides a graphical editor. The structure of the model using the enterprise library from AnyLogic in Version 6.5 can be seen in the figure 3.4.

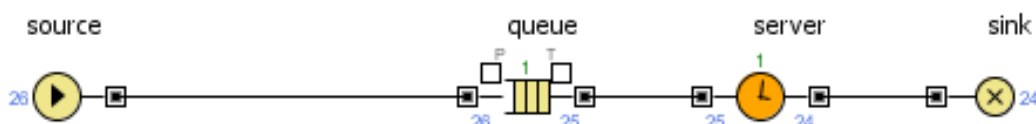


Figure 3.4: Graphical representation in AnyLogic of the simple (M/M/1) simulation example.

<sup>20</sup> XJ Technologies – AnyLogic <http://www.xjtek.com/>

For this example of a bank system (M/M/1), it is assumed that every 2 ½ minutes a client enters to the bank. The arrival time of clients have independent inter-arrival times. As we will see in the following of this work, a big advantage of simulation is the ability to represent input and output data by statistical distributions. Thus, in the source the arrival rate of 2 ½ minutes per clients is described by a Poisson distribution with parameter  $\lambda = 2,5$ .

If the entities representing the clients are ones generated in the source, they follow their way through the system to the queue. If the server is not busy and they queue empty the client passes directly to the server. If not, the client has to queue. These queuing systems also may have capacity limits. In the case of a bank, this might be the physical space within the bank. In the case of a manufacturing process, this could be the storage space in front of a machine.

A bank assistant serves the queuing clients. The service is assumed to take 2 minutes. For the service time a variation of + / - 1 minute is assumed. This is done using a uniform distribution with a lower boundary of 1 minute and an upper boundary of 3 minutes.

After being served, the clients leave the bank through the sink. The sink is also often used to collect statistical data from the entities before they leave the system.

Assuming the following: The system was in an empty state, for instance in the morning when the bank opens. The simulation length was defined to be 60 minutes. In these 60 minutes 26 clients entered the bank and 24 were served and left the bank.

If in average every 2 ½ minutes a client enters and the service time is 2 minutes +/- 1 minute, on the first view there should not be people queuing, because the average service time is 20% inferior than the arrival of the clients. The output analysis of this simulation shows, that the queue goes up to two clients waiting to be served twice. The average time which a client is in the system (in the bank), is about 3 minutes and 10 seconds. On the other hand, the utilization of the bank assistant is 85,7 % of the available service time. This output analysis provided by AnyLogic is shown graphically below (Figure 3.5, 3.6 and 3.7).

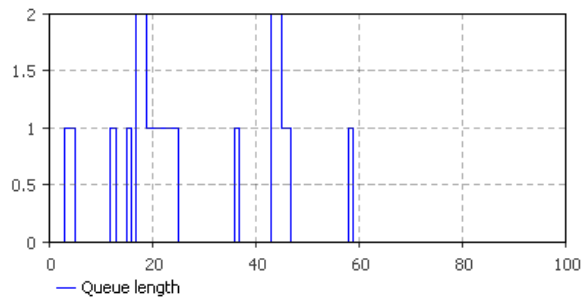


Figure 3.5: Queue length of clients waiting to be served.

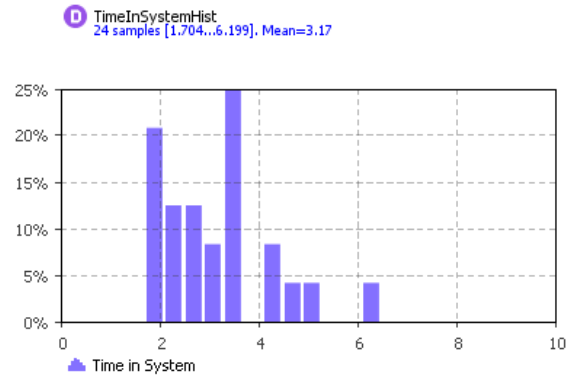


Figure 3.6: Time which a client spends in the bank.

To find the optimal system configuration is part and duty of a simulation study. In this case, this would be to take the decision between the clients waiting time and employees employed.

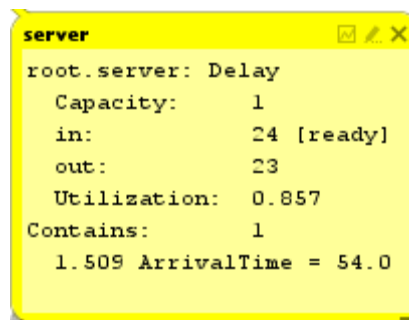


Figure 3.7: Statistical output of the server (bank assistant).



### *All science is modelling*

*“Alle Wissenschaft hat Erfahrungen zu ersetzen oder zu ersparen durch Nachbildung oder Vorbildung von Tatsachen in Gedanken. Diese Nachbildungen sind leichter zur Hand als die Erfahrung selbst und können dieselbe in mancher Beziehung vertreten.”*

**Ernst Mach, 1883**

## 4 Simulation Software

---

Taking into account the vast variety of different simulation software in the market, electing the appropriate one is a very important decision for the success of a simulation study. Simulation software can be classified in different ways. First of all, whether the software is of general-purpose or specific-purpose. Secondly, it can be distinguished between simulation languages, simulators and general programming languages.

### **4.1 Classification of simulation and historical review**

The historical development of simulation and simulation software can be seen in different ways. Pidd and Goldsman have presented two interesting historical reviews [35], [36]. According to Pidd the development of DES software has passed through different phases. On the other hand, Goldsman distinguishes the developments in periods. There is a so called formative period from 1945 to 1970 where main developments for simulation have been made. During this period, the first general-purpose simulator GSP (General Simulation Program) was developed and the first textbook in simulation, *The Art of Simulation*, was published (in 1963). Also Geoffrey Gordon appears in this period, with the development of the General Purpose Simulation System (GPSS) between 1960 and 1961. In the following period, the expansion period, the focus laid on an enhancement of simulation tools. For instance, important tools as GASP IV, SLAM and SIMAN were developed. Recently, the phases of Visual Interactive Simulation (VIS) followed by Visual Interactive Modelling Systems (VIMS) have been passed through.

#### **4.1.1 Simulation software**

In “The 2009 simulation software survey” Swain has listed 50 commercial products for DES from about 27 different companies [37]. In this survey, information about the different software is provided:

- The developing company of the simulation software
- Typical applications and the markets for the software
- System requirements
- Model building facilities, whether graphically (by icons or drag and drop), programming or both
- Model building
- Input and output analysis of data
- The possibility of experimental design
- Model packaging and code reuse
- Animation
- Support
- Price
- New features since the last survey from 2007

This survey is not a complete list of available DES software on the market. However, it gives a good overview of them, in combination with the specific information of the provided software features. A summary of the 2009 simulation software survey can be found in the annex A of this work.

Besides the commercial software packages, there exists also a huge number of free or open source simulation software tools. These cover the range from small simulation libraries, with limited simulation possibilities, to full simulation packages and simulators with data analysis and animation. A list of such free and open source simulation software tools can be found at the webpage of The European Multidisciplinary Society for Modelling and Simulation Technology (EUROSIS) [38]. This listing includes software for 3D modelling, environmental modelling, traffic simulation, as well as, discrete-event simulators.

Above and beyond commercial and free/open source software, the available simulation software can be classified. Wenzel classifies them in her work into two groups of

simulation tools, depending whether they are for general usability or oriented to a special application [7]. Figure 4.1 illustrates this classification.

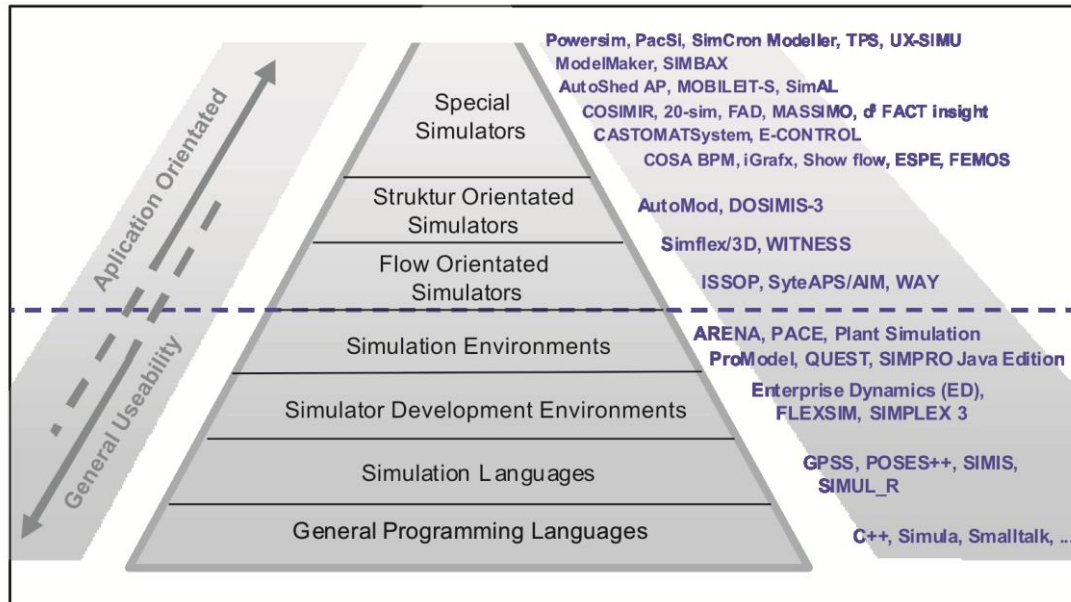


Figure 4.1: Summary of simulation tools for production and logistics. Modified from Wenzel [7].

On the bottom of the triangle general programming languages like C#, C++ or Java are localized. There exist simulation libraries and object-orientated modelling tools for them. Then the simulation languages can be found, which are computer languages especially designed to perform simulation. Followed by simulation environments and simulators on the top of the triangle. The simulation environments, simulation languages and general programming languages are of general usability. The simulators are application orientated.

### Simulation on spreadsheets

For completeness, simulation on spreadsheets is included in this work. In a lot of current literature, simulation on spreadsheets is classified as a fourth group of simulation tools in addition to simulation languages, simulators and general programming languages [33]. Simulation on spreadsheets is widely used, because it is quite easy to use. Just adequate spreadsheet software (e.g. Microsoft Excel, or Open Office Calc) is needed to perform the simulation. However, their power is very limited in comparison to other simulation tools. Furthermore, there are simulation plugins for spreadsheet software.

#### 4.1.2 Features of simulation software:

The specific features required for a certain simulation study may vary due to the circumstances of each simulation problem. Law and Kelton categorize the needed and expected features for simulation software into the following groups [1]:

- General capabilities
- Hardware and software considerations
- Animation
- Statistical features
- Customer support and documentation
- Output reports and plots

#### 4.2 Random numbers

All simulation models need random numbers to perform the stochastic randomness of the system. This randomness is placed in probability distributions which use the random numbers. There exist different random number generators (RNGs) to generate such random numbers. Besides real random numbers and hardware RNGs (that generate true random numbers from a physical process) exist software RNGs as well. The real random numbers and hardware RNGs are non-deterministic while software RNGs are deterministic.

According to L'Ecuyer [39] random number generation for simulation applications is composed of two steps:

Step 1) Uniform distributed random numbers are generated in the interval  $(0,1)$ .

They are independent and identically distributed (i.i.d.).

Step 2) To generate random variates from distributions, the i.i.d. random variates  $U(0,1)$  are transformed.

##### Real random numbers

An example for real random numbers is a fair die in a fair environment. However, in the real world, such a fair die in a fair environment can hardly be found. The outputs of a fair die can be seen as independent random variables. Technically, these outputs can be modelled as a sequence of variables that are distributed uniformly using a set of integers from 1 to 6 (assuming the die has the numbers from 1 to 6). As it is difficult to obtain the

huge number of random numbers that are needed in a simulation study, the RNG is hardware or (mostly) software based.

#### Non-deterministic random number generators

Non-deterministic RNGs are physical devices that generate the random numbers from an underlying physical process. In simulation the performed experiments must be repeatable. With non-deterministic RNGs the different simulation runs cannot be repeated, or the used random numbers have to be recorded.

#### Deterministic random number generators

Deterministic RNGs are also called pseudo random number generators (PRNGs) and their outputs are consequently named pseudo random numbers. The RNGs which are used for simulation are in general deterministic.

A mathematical definition of a RNG is given by L'Ecuyer [39], where a RNG can be defined as a composition  $g(S, \mu, f, U, g)$  with:

$S$  : a finite set of states

$\mu$  : a probability distribution based on  $S$  given the initial state or seed  $s_0$

$f$  : the transition function  $f : S \rightarrow S$   $s_0$

$U$  : a finite set of output symbols

$g$  : the output function  $g : S \rightarrow U$

The output at step  $n$  is  $U_n 2^b = g(S_n) \in U$ , being the  $U_n$  the random numbers generated by a RNG.

Due to that the set of states in  $S$  is finite; the RNG might come back to an already used state. This happens when the period length  $p$  of a RNG is reached. The length of  $p$  is limited by  $p \leq 2^b$ , being  $b$  the bits representing the number of states. Hence, when the length  $p$  is reached the RNG might repeat the sequence of the generated random numbers or just change it using a number different from zero. Historically the cycle length  $p$  was around  $2^{31}$ .

A seed is the initial value where the random number sequence starts. Using the same initial value (seed) at different times or on different computers, the RNG will produce the same random number sequence.

There are different deterministic RNGs. The most used is the linear congruential generator, which was introduced in the 1950<sup>th</sup> [1]. This RNG is defined by a recursive formula:

$$Z_i = (aZ_{i-1} + c)(\text{mod } m) \quad (4.1)$$

where:

$m$  : the modulus

$a$  : the multiplier

$c$  : the increment

$Z_0$  : the seed

Besides the linear congruential generator, the mixed and the multiplicative congruential generators are also of common use.

### Non-uniform random variates

Generating non-uniform random variates from probability distributions, like exponential, normal, chi-square, is done by transformation of the uniform random numbers. Such transformation of uniform random numbers to generate non-uniform random variate is illustrated in figure 4.2.

Transformation:

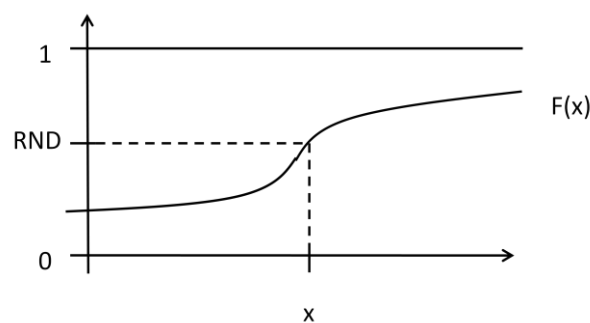


Figure 4.2: Transformation of a uniform random number.

### Streams

Another important feature of RNGs is the ability to provide different streams. Kelton proposes that these streams can be subdivided in sub-streams. These sub-streams

separate independent activities in simulation, through indexing and assigning independent *chunks* of random numbers multi-dimensionally. Furthermore, streams are used in variance reduction [40].

### Quality of RNGs

In order to have reliable results, the use of truly and statically tested random numbers is necessary. The re-use of random number sequences can threaten the validity of simulation results. This can be explained by the fact that when random numbers are repeated, they are not “random” anymore. For instance, if a die is thrown  $n$  times, the random number sequence is of length  $m$ , being  $n < m$ . But, when  $n > m$  the random numbers are repeated, and they are not random any longer [41].

As a consequence, if the generated random numbers are invalid the output also becomes invalid. To avoid this it is essential that the length of random numbers sequences is long enough. This can be achieved by using RNGs with implemented cycle lengths of  $10^{57}$  and even more [40].

Many simulation tools, including commercial ones, still use poor RNGs with historical cycle length of around  $2^{31}$ , which must be discarded. Kelton goes very hard in charge with those companies which still use such insufficient RNGs, that can reach the maximum cycle length in a few minutes [40].

## **4.3 Selected simulation software**

In this subsection, different simulation tools are presented to point out the differences between them. These tools are either of general purpose or specific purpose. In addition, they are representatives from the three categories of simulation (simulation languages, simulators, and general programming languages). The table 4.1 gives an overview of selected simulation tools and classifies them.

Table 4.1: Overview of selected simulation tools with purpose and category.

Simulation Tool	Purpose	Category
PowerSim	Special purpose	Simulator (Special simulator)
Plant Simulation	General purpose	Simulation environment
GPSS	General purpose	Simulation language
SiRO	General purpose	General programming language (Simulation library)
Simio	General purpose	Simulator (Development environment)

#### 4.3.1 PowerSim

In a PowerSim<sup>21</sup> application, the simulator may be coupled with a geographical information system. With this geographical information it is possible to study, for instance, the impacts of heavy rainfall or the expansion of a gas cloud after a gas leakage.

#### 4.3.2 Plant Simulation

Plant Simulation<sup>22</sup> is a graphical component simulation environment. Plant Simulation provides several standard classes, which includes all basic components for the model construction. Through the modelling editor, the components are placed and connected with paths. As the basic compartment of the pre-defined standard classes may not be enough for all circumstances Plant Simulation provides a scripting language (SimTalk) for self defined methods [16].

#### 4.3.3 GPSS

The General Purpose System Simulation (GPSS) goes back to Gordon, a former IBM employee [42]. For many members of the simulation community Gordon is the father of DES. GPSS is an antiquarian simulation language. However, it is still represented and is used on the market. There are free and open source, as well as commercial GPSS simulation packages on the market. Since the development of the original GPSS in the year 1961, several developments and extensions of the original GPSS had been done.

<sup>21</sup> Powersim: [www.powersim.com](http://www.powersim.com)

<sup>22</sup> Technomatix Plant Simulation: [www.emplant.de](http://www.emplant.de)



Nowadays exist for instance GPSS/H, GPSS World, WebGPSS and a Matlab GPSS toolbox [43].

Using GPSS, so called blocks are used to construct a simulation model. These blocks are (in most GPSS versions) text-based, and consist of a label, a verb (with the block name) and operator(s). The label is just a naming of a block to distinguish them. The verb is the actual GPSS command having predefined operators that defines, for instance queues, servers, functions and so on. Each block can have various operators that specify them. The structure of GPSS blocks is as follow:

LABEL	BLOCKNAME	OPERATOR(S)
-------	-----------	-------------

Besides functions, tables, matrices, variables exist around 40 executable blocks.

A short example illustrates the use of GPSS: A source that generates 500 orders uniformly distributed between 60 and 90 seconds with priority 15 is shown. The first order is generated 2 hours after the simulation run started:

Order1	GENERATE	60,90,7200,500,15
--------	----------	-------------------

#### 4.3.4 SiRO

The abbreviation SiRO stands for **S**imulation, **R**ating and **O**ptimization. SiRO has been developed since 2008 at Profactor GmbH located in Steyr, Austria. It is Profactor's simulation tool for performance-critical applications. According the classification given in figure 4.1, SiRO is on the bottom line of the pyramid, because it makes use of a general programming language. To reduce the effort to build simulation models, the SiRO framework provides an object-oriented simulation library. This library is available as a Java and C# edition as well. At the moment both editions comport identically (SiRO Version 2.1). A simulation model, using SiRO, is implemented directly in the programming frameworks of Java and C# (e.g. Eclipse and MS Visual Studio). A graphical modelling editor is not available.

##### Applications and features of SiRO:

SiRO's basic field of application is in manufacturing and producing industries, for example in the automotive and plastic sector. More applications of SiRO can be found in supply chain management as well as in the transport and pharmaceutical industry [8].

In the following some features of SiRO are provided:

- Makes strong use of generics.
- Using general programming languages together with a simulation library provides many technological advantages. However, the programming effort is higher than using simulation languages and simulators.
- Allows the use of simulation-based optimization and has the ability to be coupled with other optimizers.
- Put logic where it belongs by separation of events and event handling.
- Provides the possibility to represent stochastics and to disable them.
- Uncomplicated simulation model extension.

### Events and Event handlers

For the event handling SiRO distinguish between *events* and *event handlers*. A definition of what is meant by event and event handler in SiRO is given in the following:

“**An event** is a named, abstract entity, instance of which can occur repeatedly.”

“**An event handler** is basically just a method, executing some logic as a consequence to the event occurrence and dependant on the event data [8].”

An event can have several event handlers. The events can be scheduled from zero time (where the simulation clock stands at the moment) to any point in time in the future. After having successively processed all events of a point in time, the simulation clock steps forward to next point in time which contains events. With `AddEvent` and `RemoveEvent` SiRO provides two functions for the event scheduling.

### Random numbers

SiRO contains a PRNG that makes use of a central seed. This seed can be set by default or by the user. SiRO provides by using this central seed, a consistent way to assign random numbers. The assigned seeds for the simulation objects are based on the *id* of the objects, thus the order of which they have been constructed is not relevant. Providing this feature makes simulation runs and simulation experiments reproducible.

Figure 4.3 shows how the multiplication of the seed index and the model seed form the assigned seed for the model object. Within the object various RNGs are used to provide the necessary random numbers. These random numbers acquire their seeds from the object seed source, which depend on order [8].

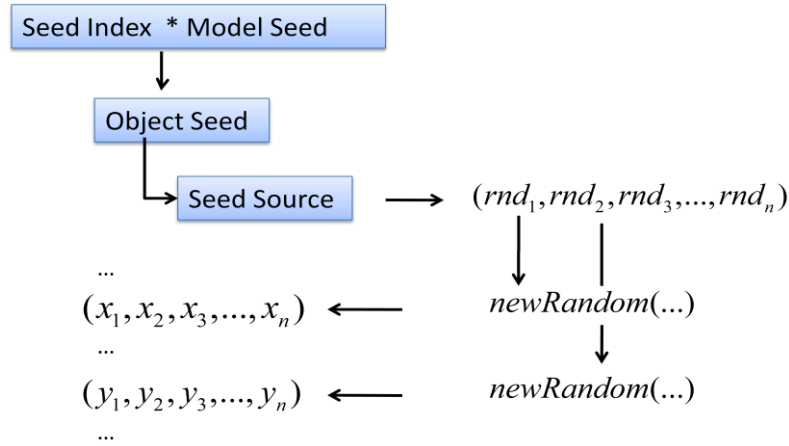


Figure 4.3: Assigning random seeds in a consistent way, according to Löscher and Gruber [8].

### Distributions

SiRO provides all necessary distributions that are expected from a simulation framework. For the representation of stochastics, in the `profactor.math.dll`, the following distributions are implemented: Bernoulli-, Beta-, Binomial-, Geometric-, Constant-, Erlang-, Gamma-, Gaussian-, LogLogistic-, LogNormal-, Uniform-, NegBinomial-, NegExponential-, Pearson-, Poisson-, Triangular- and Weibull distribution.

In the annex of this work some information about statistics for simulation is given together with the illustration of several distributions.

### Templates

The SiRO framework provides a variety of templates in the simulation libraries. Using templates reduces the modelling effort notably. Among others, exist the following templates:

- `Queue<T>` is a generic queue.
- `Sink<T>` is a generic sink, where the entities leave the system. In the sink also the data collection of the received entities is done.
- `Source<T>` is a generic source, which can produce entities of any kind.
- `Delay<T>` is a typical generic delay, where the duration can be specified.
- `Product` provides the basic characteristics of a product.
- `Machine` provides the basic comportment of a machine.

- MobileObject represents the movable entities that flow along their way through the system.
- ResourceObject are resources that can be seized and released.

### Model building

All the physical elements of a system are represented as model objects. With the function AddObject the model objects are added to the simulation model. By linking the objects together a simulation model is created. This is done with the function OnBuildModel.

### Résumé of SiRO

The development of SiRO at Profactor GmbH is an ongoing research project, with a lot of potential. At the moment, there are several important fields which are being improved, like the rating framework and coupling SiRO to optimizers. Further, an accurate model visualisation being developed. Also the template library has to be extended in order to make modelling easier and faster for modellers. SiRO is available commercially at Profactor<sup>23</sup>.

#### 4.3.4.1 An exemplary implementation using SiRO

The dining philosophers problem is presented in the following to illustrate an exemplary implementation using SiRO. The problem implementation was done by the Profactor GmbH. A Tutorial of this implementation using SiRO is also available at Profactor.

#### Problem specification:

The dining philosophers problem is a well known problem in computer science. Different approaches to model and simulate the problem are possible, for instance Petri Nets and DES.

Five philosophers are sitting on a round table. Each philosopher has a plate with food in front of himself. Hence, that a philosopher can eat, two chopsticks are needed. But there are just five chopsticks available on the table, which are placed between the plates. Consequently, just a limited number of philosophers can eat at the same time (Figure 4.4).

---

<sup>23</sup> Profactor GmbH - SiRO: [www.profactor.at/production/produkte/siro.html](http://www.profactor.at/production/produkte/siro.html) , last visit: 07.09.2011.

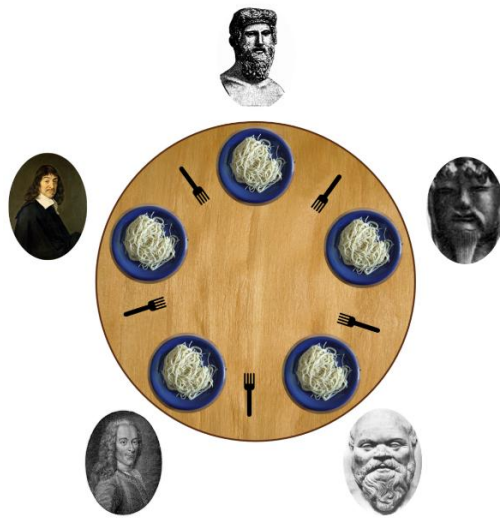


Figure 4.4: Dining philosophers<sup>24</sup>.

*States of the philosophers:* They can either be thinking or eating. Further they can get hungry and stuffed.

*State transitions:* Four state transitions are possible. From thinking to hungry, from hungry to eating, from eating to stuffed and from stuffed to thinking.

*Resources:* The five chopsticks, which are shared by the philosophers. If a philosopher gets hungry he takes first the left chopstick if this is available. Then, he takes the right chopstick if this is available. If a chopstick is not available the philosophers wait until it becomes available.

*Possible deadlock:* In that case, that all five philosophers are hungry at the same time and all have been taken the left chopstick and wait for the right one. Thus, the philosophers have to die.

A more detailed specification of the dining philosophers problem can be found on the ARGESIM webpage in Comparison 4 [44]. There also can be found different implementations (source code and a short description) of this problem using other simulation tools.

### Implementation

The proceedings of the implementation of the dining philosophers problem using the SiRO simulation libraries are described in the following. Having added the two libraries

<sup>24</sup> Source: [www.wikipedia.org](http://www.wikipedia.org) - Dining philosophers problem, last visit 20.08.2011.

profactor.siro.dll and profactor.math.dll to a new project four classes are used for the problem implementation.

*Program class:* The program class contains the main method. In this implementation a form is used where logged events and other relevant information are printed out. Further the model is built and executed.

*Simulation class:* The simulation class contains the simulation model, where the model elements are added. The five chopsticks are added as *resource objects* and the five philosophers are added as *objects* to the simulation model. This class is also the main container for the simulation experiments. In this case just a model seed can be customized to perform simulation experiments using different model seed values. Thus, different outputs and possible deadlock times can be observed. The simulation class also provides data inputs, like database connections. But those are not used in this example. In the following figure 4.5 the implementation of the simulation class is shown:

```
using System;
using Profactor.SimEngine.Engine;
using Profactor.SimEngine.Utilities;
using System.Collections.Generic;
namespace DiningPhilosophers.Model
{
    public class Simulation : DefaultSimulation
    {
        public Simulation(int seed) : base(seed)
        {
        }

        public override void OnBuildModel() {
            base.OnBuildModel();

            for(int i = 0; i < 5; i += 1) {
                Model.AddResourceObject(new Chopstick(i));
                Model.AddObject(new Philosopher(i));
            }
        }
    }
}
```

Figure 4.5: SiRO implementation of the Simulation class.

*Chopstick class:* The chopstick class provides the implementation of the five chopsticks which are *resource objects*. This class basically contains a constructor of a chopstick together with an index which is assigned to each chopstick.

*Philosopher class:* When the model is initialized two random numbers which are uniform distributed are generated. They are used to map the random behaviour of the philosophers, for the time they get hungry again after having eaten and the time they get stuffed after having started to eat.

The possible states of the philosophers are enumerated. The philosophers can be thinking, hungry, eating and stuffed. Further a state undefined is implemented if the actual is not defined. In addition the state transitions are described.

There exists just one event, the “left chopstick is picked up” event. For this event an event handler which executes the logic is added. This event handler basically tries to take the right chopstick. And finally, if the five philosophers have the left chopstick in their hand it comes to a deadlock.

An exemplary implementation of other classes of the problem can be found in the annex of this work.

#### 4.3.5 Simio

Simio<sup>25</sup> (**s**imulation modelling framework based on **i**ntelligent **o**bjects) is a very recent and interesting development from Simio LLC located in Pittsburgh, Pennsylvania, USA. The founders of Simio LLC have already developed simulation software in the past (e.g., ARENA<sup>26</sup>). Simio is an object-based modelling environment, which support both 2D and 3D models. According to Pegden, in Simio the objects are process-based. Furthermore, Pegden suggests that this process-based approach is more beneficial than code-based approaches are. Instead of function calls, process-based objects have so called *add-ons* to execute the logic. The described advantage of this approach is that the framework is easy to use for the modeller and brings them more flexibility [45].

The general idea behind Simio of using objects to build a model is similar to the described approach of SiRO. The difference is that not much programming and scripting knowledge is needed because of a graphical modelling editor is available. The development stage of Simio is in a quite advanced stage and will gain more importance in the future.

---

<sup>25</sup> Simio LLC: [www.simio.com](http://www.simio.com)

<sup>26</sup> ARENA: [www.arenasimulation.com](http://www.arenasimulation.com)

## 4.4 Case study: Comparison implementation using different simulation tools

To compare different simulation software in practice, a case study implementation using different software has been done. Diverse aspects of the different simulation software are compared. However a main focus will be on performance. In simulation studies runtime concerns may be critical, as it will be shown in the further work. The implementation of Comparison 6 – “The Emergency Department – Follow-up Treatment”, of the ARGESIM comparisons was elected [44]. This is done in three different simulation software tools, representing three different categories:

- Plant Simulation (Simulation environment)
- GPSS (Simulation language)
- SiRO (Simulation library)

### Problem description:

This problem addresses DES, where the discrete processes of a hospital are studied. Arriving patients make their ways through the system according their type of treatment needed and other probabilistic assignments. The patients arrive exponentially distributed and the different treatment times are triangular distributed. The entire problem formulation can be downloaded from the ARGESIM webpage [44].

### Implementations using different simulation software tools:

Simulation language GPSS: As GPSS simulator SNAKE<sup>27</sup> is used. The model is created by using a set of about 90 to 100 GPSS blocks. The resources of the hospital are represented by so called *Facilities* (single server) and *Storages* (parallel servers). The distributions are implemented by functions that use the underlying pseudo random number streams. For the output analysis the GPSS simulation software SNAKE provides adequate text-based outputs on tables and matrices.

Simulator Plant Simulation: All model elements together with the necessary methods, tables, event controller and experiment manager were added to create a simulation model. Figure 4.6 illustrates this.

---

<sup>27</sup> SNAKE (Simulation New Advanced Kit – Enterprise edition)  
<http://materias.fi.uba.ar/7526/programas.html> , last visit 07.09.2011.



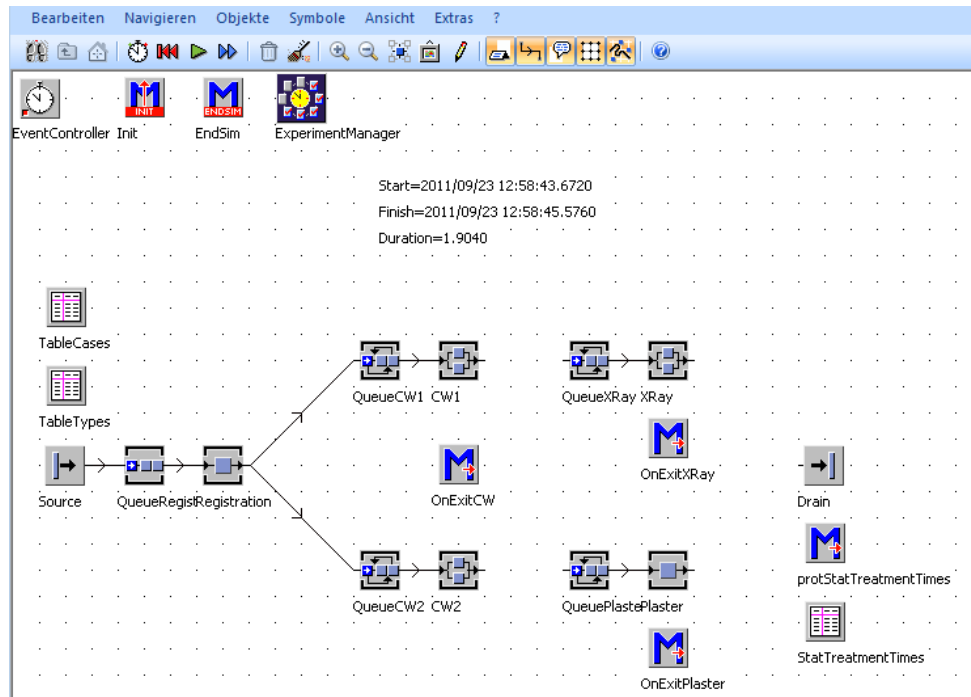


Figure 4.6: Illustration of the model components in the Plant Simulation. Screenshot of the graphical modelling editor.

Simulation library SiRO: The comparison was implemented using the Java version of SiRO. As the simulation model makes strong use of generics it can be extended easily (e.g., if more or less doctors should work). The model contains nine classes to represent the system of the emergency department. For the event handling three different events are used.

### Résumé:

The provided features of the different simulation software were satisfying. The implementation of the comparison was possible with all the three simulation tools. As expected, all simulation software have provided an adequate way to assign random numbers and random distribution varieties of the problem.

The advantage of an object-oriented simulation library is that the simulation model is easy to extend. However, the modelling process is notably more elaborative. On the other hand, the advantage of a general simulation language is that a special tailored and well tested language is available. Nevertheless, the ability to include more complex model interactions, where special behaviours are needed, is limited. Finally, the advantage of using a simulation environment with a graphical modelling editor is that modelling effort is

reduced. Furthermore, not much programming knowledge is required. Plant Simulation provides a script language to put required logic into the simulation model objects.

Table 4.2 summarises the times needed to perform a simulation run of the problem implementation using the different simulation software.

Table 4.2: Times needed to calculate the result.

Simulation tool	Duration to perform a simulation run
GPSS (SNAKE)	6 seconds
Plant Simulation	1,9 seconds
SiRO	90 milliseconds

The studies were performed on a Pentium 4; 2,6 GHz Laptop. In all simulation tools the logging was enabled. The GPSS simulator SNAKE took around 6 seconds to perform a simulation run. The simulation environment Plant Simulation, with a duration of 1,9 seconds was slower than the fastest performance delivered by the in Java implemented SiRO model, which took 90 ms. Two articles of this ARGESIM comparison have been submitted to the SNE Journal<sup>28</sup>. In the category “software benchmarks” different simulation software are compared. In the annex of this work these two articles can be found, including the simulation runtimes using SiRO and SNAKE.

---

<sup>28</sup> SNE – Simulation News Europe is an simulation journal edited and published by ARGESIM

*“If you have an apple and I have an apple and we exchange apples then you and I will still each have one apple. But if you have an idea and I have an idea and we exchange these ideas, then each of us will have two ideas.”*

**George Bernard Shaw**

## 5 Rating (of simulation replications)

---

### 5.1 What is meant by rating

As already mentioned in previous chapters, a simulation run or experiment delivers as result output data. In the case of a producing company this output may represent the operating data of the production process. This operating data has to be analyzed and evaluated after having performed the simulation run. This analysis and evaluation process can be called rating. Therefore, the rating gets the operating data from the simulation and describes the objective function  $Z$  of an optimization problem. This objective function  $Z$  is represented by a fitness value, which has to be:

maximized:  $Z \rightarrow \max$

or

minimized:  $Z \rightarrow \min$

### 5.2 The rating framework

In order to calculate a fitness value, a multi-stage online rating framework is presented in the following. The fitness value is obtained from simulated indices by transferring, merging and weighting them into comparable units using multi-stage rating. These indices are

based on the operating data of the company, previously obtained as simulation output. The number of stages is not limited; the illustration in figure 5.1 shows however an example of a 2 stage rating model, which gives the principles of the rating framework.

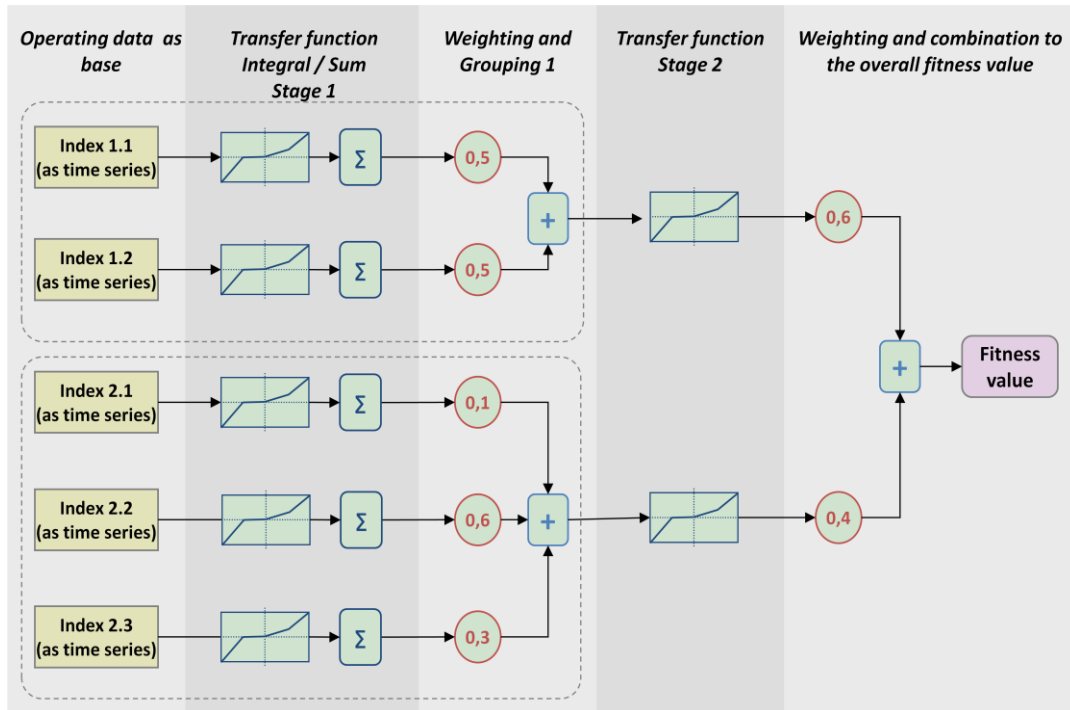


Figure 5.1: A multi-stage rating model to calculate the fitness value. According to M. Vorderwinkler, Procomposite – FoT.

### 5.2.1 Summary of the steps of a rating process

- A) First the relevant indices which should be included in the rating model (and consequently in the fitness value) have to be determined.
- B) Several indices that represent similar objectives may form index groups.
- C) The single indices and, in further stages, the index groups have to be transferred into a common comparable unit.
- D) The indices and index groups if any, have to be weighted.
- E) The single indices and, in further stages, the index groups have to be merged.
- F) In multi stage rating this process of transferring, merging and weighting can be done  $n$  times until the final fitness value is obtained.
- G) The fitness value is provided to the optimization.

A) The relevant indices which should be included in the rating model have to be determined

Before starting with the rating process, relevant information, which forms part of the company strategies, have to be identified. From this information the respective indices have to be derived. To form these determined indices, the delivered operating data is provided from the simulation as time series.

The aim is to build a holistic rating model that includes the relevant indices in order to fulfil the company objectives. One difficulty is that often companies don't know which indices are relevant and which are not.

According to Eidenmüller there are three dimensions of information that can be considered to determine the indices [46]:

- *Objective dimension:* These include operational, quantifiable and realisable objectives, which can be well defined. They are for instance CO<sub>2</sub> and other emissions, the productive output, unit costs and so on.
- *The classical indices:* These include indices which are in relation with the objective dimensions. For instance the energy use, the effective availability of machines, manufacturing costs and so on.
- *Measured variables:* They are in relation with the classical indices. Measured variables are recorded, for instance the cycle time of a machine, the point in time material enters and leaves, the absence rate of workers and so on.

B) Several indices that represent similar objectives may form index groups

In order to make use of the multi-stage rating it is possible to build groups of indices. Similar indices can be gathered in a group. Possible index groups could be elected for technical, economical, environmental and sustainability indices. Table 5.1 shows a short overview of possible groups building.

Table 5. 1. Possible index groups building.

Technical indices	Economical indices	Environmental and other sustainability indices
Efficiency of machines	Production costs	Emissions
Efficiency of workers	Storage and transportation costs	Energy and resource use
Number of defect products	Cycle times	Load of workers
	Adherence to delivery dates	Work monotony
	Throughput	
	Customer satisfaction	

C) The single indices and in further stages the index groups have to be transferred into a common unit

In this step of the rating process the indices are transferred into a common unit, which represents the penalty. This transformation of the indices is done using a (non)linear transfer function (represented as a polygon line). For the penalty, any unit can be chosen, as long as they are comparable, for instance point values as scores or monetary units.

*Description of transfer functions*

A transfer function assigns a value (that represents the penalty as scores or monetary units) to an index. The x-axis represents the value of the index. The y-axis represents the amount of the penalty. The base points are interpolated and form the transfer function. A central question is how the slope behaves when the index value changes. Selecting the base points together with the slope in an appropriate way is essential. Small changes of the transfer functions can lead to big differences in the rating result. Thus, the function has to be carefully elected if not, the optimization will be yield to not desired results. An exemplary transfer function for the adherence to delivery dates is shown in figure 5.2.

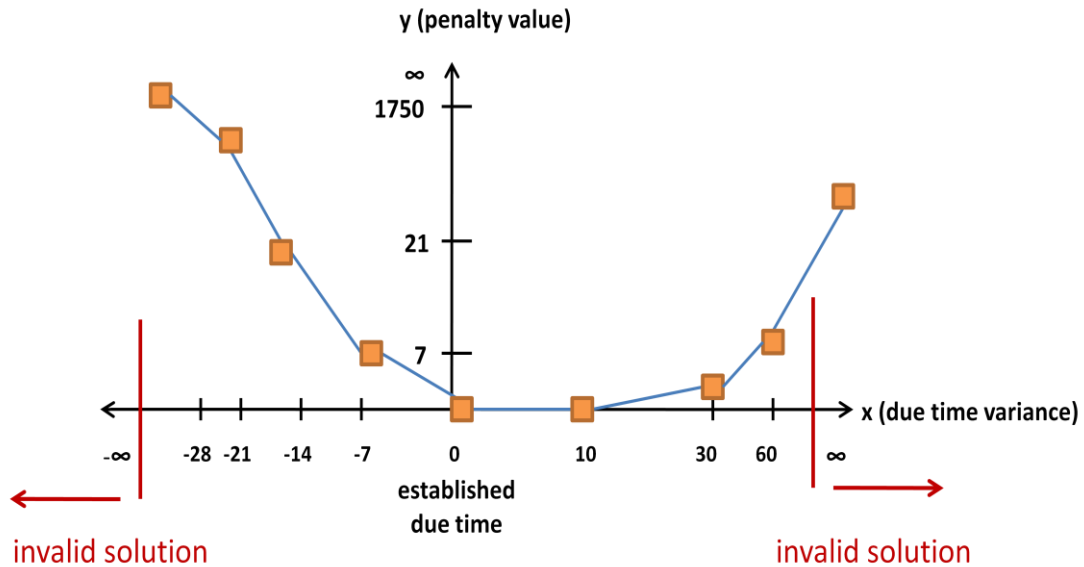


Figure 5.2: Example values of a transfer function representing adherence to delivery dates.

In this illustration the x-axis represents the adherence of the delivery dates (also called tardiness) in days. Positive adherences to delivery dates (the product is finished earlier than the customer desired) are represented as positive values. Negative x-axis values are represented as negative adherence to delivery dates. If the adherence is higher than a predefined value the solution gets invalid.

#### D) The indices and index groups if any, have to be weighted

The different indices and index groups have to be weighted. This weight specifies the percentage of an index in the fitness function or within an index group.

A question here is how to weight them. This weighting process may be simple if there are just a few indices to be considered. But, if there are big numbers of indices and index groups which have to be weighted the process gets more difficult. In that case the use of appropriate software is helpful. However, in practice many times the weighting of the indices is done based on experiences or estimations by the factory operators. At Profactor GmbH an Excel spreadsheet is used where the indices are first weighted one by one with each other. This spreadsheet can be seen as a preliminary stage to the developed rating framework. Here it is specified whether an index is more, equal or less important than another index. This is done for all index combinations and at the end the corresponding weighting of all indices is obtained. Figure 5.3 shows the spreadsheet for the weighting.

#	Indices	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Weight	Factor
1	Index																0	
2	Index																0	
3	Index																0	
4	Index																0	
5	Index																0	
6	Index																0	
7	Index																0	
8	Index																0	
9	Index																0	
10	Index																0	
11	Index																0	
12	Index																0	
13	Index																0	
14	Index																0	
15	Index																0	
Sum:																	0	1

Figure 5.3: Classifying the indices according the significance in the fitness value. Source: Profactor GmbH.

This process is similar to the well known method of analytic hierarchy process (AHP). One commercial software that implements the concept of AHP is Expert choice<sup>29</sup>.

E) The single indices and in further stages the index groups have to be merged

Index series without time reference can be brought together by adding them up. On the other hand, index series with time reference can be brought together by integration over the timeline.

F) In multi stage rating this process of transferring, merging and weighting can be done  $n$  times until the final fitness value is obtained

Using the whole functionality of the rating framework the steps of transferring, merging and weighting can be performed  $n$  times until the final fitness value is obtained.

G) The fitness value is provided to the optimization

When the simulation run is done and the fitness value is calculated it is provided to the optimizer.

<sup>29</sup> Expert choice: [www.expertchoice.com](http://www.expertchoice.com)



## 5.3 Formulation of the rating model

Based on the conceptual developments of the descriptions above, a rating framework was implemented to allow the formulation of a rating model. The result was a software library in .NET. This library was mainly designed to be used with the simulation library SiRO. However, it is not limited to this.

The modeller has the opportunity to link dynamically several elements of the rating framework to a rating model together. These elements are represented by nodes which are linked together by paths. In the library the following nodes are provided:

- Input nodes: are used to represent the simulation output (operating data) as indices. The input nodes are the initial point of the rating model formulation.
- Accumulator nodes: are used to add identical indices.
- Sum nodes: are used to add different indices.
- Integration nodes: are used to integrate time dependent indices.
- Differential nodes: are used to calculate the slope of an index. Using this node it is possible to determine if and how fast an index decreases or increases (e.g., the stock level).
- Transfer function nodes: are used when an index (or indices groups) has to be transferred.
- Weight nodes: are used to give an index or indices groups the desired relevance within the rating model.
- Statistical nodes: are used for the statistical analysis of the rating model output.

In addition to the nodes, there are other auxiliary elements like two dimensional points, numbers, units, time spans and statistical elements provided.

In order to discard invalid solutions during runtime, the rating framework detects invalid solutions and delivers that information to the optimizer as an invalid fitness value.

### 5.3.1 Different ratings

Different rating scenarios can be evaluated and compared afterwards. This can be done easily by changing the structure of the rating model. Here, different indices may be included or excluded, the weighting and grouping of the indices can be changed and the transfer functions can be adapted. It is also possible to start a SBO study with a few indices and extend the amount of indices over time.

*“The wise man does not say everything he thinks,  
but always thinks everything he says.”*

**Aristotle**

## 6 Simulation-based optimization: basics, methods and applications

---

### 6.1 Background

With the appearance of digital computers of high performance - beginning in the early 1970s - and the ability to conduct simulation experiments, the technique simulation was converted into an important application in the area of operations research [33].

A lot of efforts have been made to couple simulation with optimization (or optimization with simulation). In the annual Winter Simulation Conference (WSC)<sup>30</sup> in 2011, simulation optimization gets an own track for the first time [47].

Further Gartner, a major analyst and research company in this area, ranks the Top 10 Strategic Technologies annually. In 2010 Gartner ranked *Advanced Analytics* on the second position, which includes optimization and simulation:

“Optimization and simulation is using analytical tools and models to maximize business process and decision effectiveness by examining alternative outcomes and scenarios, before, during and after process implementation and execution. This can be viewed as a third step in supporting operational business decisions. Fixed rules and prepared policies gave way to more informed decisions powered by the right information delivered at the right time, whether through customer relationship management (CRM) or enterprise resource planning (ERP) or

---

<sup>30</sup> The Winter Simulation Conference is an annual international forum for system simulation. Its main focus is on discrete-event simulation and combined discrete-continuous simulation.

other applications. The new step is to provide simulation, prediction, optimization and other analytics, not simply information, to empower even more decision flexibility at the time and place of every business process action. The new step looks into the future, predicting what can or will happen [48].”

In literature SBO is sometimes also called simulation optimization or optimization via simulation (OvS) (compare Hong [49]). Ólafsson and Kim [50] define simulation optimization as:

“The process of finding the best values of some decision variables for a system where the performance is evaluated based on the output of a simulation model of this system.”

In recent years SBO has done considerable developments and gained more and more importance. A good review on SBO can be found from Fu [13]. The Handbooks in Operations Research and Management Science provides a series entitled “Simulation” [51]. Further relevant literature that threats SBO have been published by Law and Kelton [1], Banks [15] and Gosavi [52]. Relevant application of SBO can be found in the following of this chapter.

The combination of DES and optimization techniques was picked up in several approaches. According to März *et al.* [9] the simulation can be coupled to the optimization in various ways:

- The optimization is integrated in the simulation. This is done to make predictions of the future state of the system. The simulation runs are performed with the actual state of the system.
- The simulation can be seen as the objective function in classical optimization problems. This is the classical simulation-based optimization approach and according to März, this way of coupling simulation and optimization is the most used one.
- The simulation result is the initial solution for the optimization. The difference to the previous point is that here, there is just the communication from the simulation to the optimization and not vice versa.
- The optimization results are used for the configuration of the simulation. This is done to check whether the obtained optimization results are feasible in the real world or not.

## 6.2 Methods of simulation-based optimization

There has been a discrepancy between the academic research and the implemented methods in commercial packages. The academic research tries to attend to the stochastic nature of DES. While, on the other hand, the used methods in commercial packages are implementations of general metaheuristic techniques, which are designed for deterministic optimization. However, gradient search, random search and mathematical programming are also modified approaches from deterministic optimization [13].

According to Keskin *et al.* [53] and Fu [13] SBO uses mainly optimization approaches that include methods like random search, response surface methodology, stochastic approximation / gradient-based procedures, ranking and selection, sample path optimization and metaheuristics. The table 6.1 summarizes the main SBO methods, which will be explained in more detail in the following.

### Random Search

Random search moves randomly from a point to another point. According to Andradóttir, a generic, four steps version of a random search algorithm to solve a SBO problem of the form:

$$\min_{\theta \in \Theta} f(\theta) \quad (6.1)$$

being:  $\Theta$  the feasible region and  $f : \Theta \rightarrow \mathbb{R}$  the objective function

is the following:

**“Step 0 (initialize):** Choose the initial sampling strategy  $S_1$  and let  $n = 1$

**Step 1 (sample):** Select  $\theta_n^{(1)}, \dots, \theta_n^{(M_n)} \in \Theta$  according to the sampling strategy  $S_n$

**Step 2 (simulate):** Estimate  $f(\theta_n^{(i)})$  for  $i = 1, \dots, M_n$  using simulation

**Step 3 (update):** Use the simulation results obtained so far in Step 2 to compute an estimate of the optimal solution  $\theta_n^*$  and to choose an update sampling strategy  $S_{n+1}$ . Let  $n = n + 1$  and go to Step 1” [54].

Random search methods can use different sampling strategies  $S_n$  and / or different sequences of estimates.

Table 6.1: Overview of optimization approaches used in simulation-based optimization. Main characteristics, the practical use and a view of some literature considered as relevant for each method.

METHOD	MAIN CHARACTERISTICS	MAINLY USED IN	RELEVANT LITERATURE
<b>Random search</b>	A search where no information is used. The search moves randomly from a current point to another point.	Research approaches in simulation-based optimization addressing the stochastic nature of simulation. Mainly not implemented in commercial software packages.	Andradóttir [54].
<b>Response surface methodology</b>	Metamodel-based optimization method. An approximation of the response that describes the objective function based on a set of input variables over an experimental region.		Mayers <i>et al.</i> [55], Barton <i>et al.</i> [56], Zang [57].
<b>Stochastic approximation - Gradient-based procedures</b>	According the estimated value of the gradient the move is done locally.		Andradóttir [58], Fu [59].
<b>Ranking and selection</b>	For a small number of combinations, where all possibilities are ranked and then best is selected.		Kim and Nelson [60], Law and Kelton [1].
<b>Sample path optimization</b>	For each simulation path the optimal solution is calculated by approximation over a sample of the path.		Rubinstein [61], Robinson [62], Deng [10].
<b>Metaheuristics</b>	The mainly used metaheuristics includes genetic algorithms, tabu search, simulated annealing, scatter search, artificial neural networks, among others.	Implemented in commercial software packages. Derived from deterministic optimization methods (compare with table 6.2).	Michalewicz [63], Ólafsson [64].

## Response surface methodology

The origin of the response surface methodology (RSM) goes back to a work of Box and Wilson [65]. In simulation RSM is metamodel based by using linear and quadratic model approximations. Mainly regression and neural networks are used to get this metamodels [13], [56].

Giving: **The input:** A set of variables  $\theta = (\theta_1, \theta_2, \dots, \theta_n)$  and

**The output:** The objective function (the response function  $f(\theta)$ )

The aim of RSM is to determine the *approximation* of the objective function (*response*) and how it is influenced when the set of input variables changes. These differences are used to optimize (maximize or minimize) the response  $f(\theta)$  of the input variables over an experimental region. The advantage of RSM is that a relative small number of experiments (simulation runs) is needed to approximate the response function  $f(\theta)$  [13], [56].

### First and second order equations: Approximation of the Response

Being:  $y$  the response variable and  $\beta_j$  the  $j^{th}$  unknown coefficient,  $\theta_j$  the  $j^{th}$  input variable and  $\varepsilon$  a random variable that represents the “error”  $\varepsilon \sim i.i.d.N(0, \sigma^2)$ .

In general the RSM consists of two phases. Phase 1 uses first order polynomial models to reach a region of the experimental space that is close to the optimum. Then in phase 2 second order models are used [56].

An example of a first order regression model is:

$$Y(\theta) = \beta_0 + \sum_{j=1}^n \beta_j \theta_j + \varepsilon \quad (6.2)$$

And consequently a second order regression model is:

$$Y(\theta) = \beta_0 + \sum_{j=1}^n \beta_j \theta_j + \sum_{i=1}^n \sum_{k=1}^n \beta_{ik} \theta_i \theta_k + \varepsilon \quad (6.3)$$

An example using RSM in simulation can be found in the work entitled “Simulation-based optimization of dispatching rules for semiconductor wafer fabrication system scheduling by the response surface methodology” [57].

## Gradient-based procedures

The idea behind gradient-based procedures is to move locally, according to estimates of the gradient, into the most promising direction [13]. Therefore, for the optimizing problem shown in (6.1) the value of the gradient  $g(\theta) = \nabla f(\theta)$  is estimated. For the estimation a

gradient estimator  $\hat{g}(\theta)$  (which uses forward differences in this case) is used:

$$\hat{g}_i(\theta) = \frac{\hat{f}(\theta + ce_i) - \hat{f}(\theta)}{c} \quad (6.4)$$

Well known techniques for gradient estimation are the likelihood ration/score function and the perturbation analysis [58], [13].

## Ranking and Selection

Ranking and Selection (R&S) methods in SBO are designed to select the best of  $k$  systems by comparing populations. These populations are a fixed and finite number of simulated alternatives. The R&S methods address the stochastic setting of a problem. The selection of the best system can be of different classes: selection of the best, comparison with a standard, multinomial selection and Bernoulli selection. A confidence level, a so called probability of correct selection (PCS) is used, to ensure that the selected system is the best one within the given probability  $(1 - \alpha)$ . Two popular R&S approaches are the indifference-zone formulation and the subset selection formulation [60].

## Sample path optimization

According to Fu [13] Sample path optimization (SPO) is used when gradient-based methods e.g., stochastic approximation are not applicable. The goal of this method is to approximate the simulation result using sample observations of each path. For this, a deterministic algorithm is used in order to solve a SPO problem [10]:

$$\min_{x \in \mathbb{R}^n} \hat{f}^N(x) \quad (6.5)$$

which approximates the original problem (6.1).

Figure 6.1 illustrates the SPO method. Beginning from the path root  $x_0$  and a given number of samples ( $N$ ) the optimal solution  $\{x^{*,N}\}$  is an approximation that converts to

the real solution  $x^{*,\infty} = x^*$ . SPO is convergence proof. Often SPO is also called Monte Carlo sampling approach or sample average approximation [10].

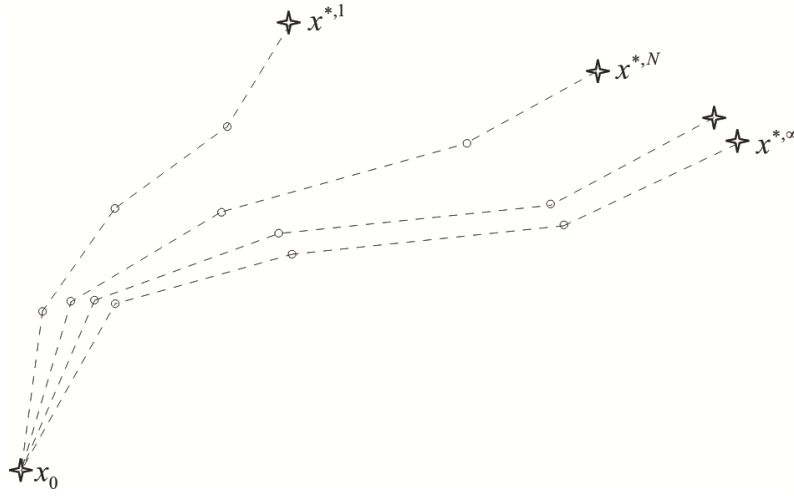


Figure 6.1: Mechanism of the sample-path optimization method, according to Deng and Ferris [10].

## Metaheuristics

Although, there exists important research in the methods mentioned before, in practice the most used optimization approach in SBO is metaheuristic optimization. Voss defines a metaheuristic as follows [66]:

“A metaheuristic is an iterative master process that guides and modifies the operations of subordinate heuristics to efficiently produce high-quality solutions. It may manipulate a complete (or incomplete) single solution or a collection of solutions at each iteration. The subordinate heuristics may be high (or low) level procedures, or a simple local search, or just a construction method.”

In almost all commercial simulation packages, metaheuristics are implemented as optimization methods. These are mostly evolutionary algorithms especially genetic algorithm, followed by the local search metaheuristics simulated annealing and tabu search. Other metaheuristics methods used in SBO include techniques like scatter search, threshold accepting (which is a variant of simulated annealing) and the nested partition method [13], [53], [64].



As the presented metaheuristics are iterative algorithms, they need a termination condition that defines how and when the algorithm stops. Possible termination conditions are:

- Reaching a maximum number of iterations
- Reaching a time limit
- Finding an acceptable solution or a global optimum if this is known
- Premature convergence

The focus in this work is on metaheuristics as optimization method to be coupled with simulation, mainly with applications in the manufacturing industry. In the following several selected optimization techniques are presented:

### Tabu search

Tabu search (TS) is a very popular approach based on the local search method. TS goes back to Fred Glover with the publications Tabu Search Part I in 1989 [67] and Part II in 1990 [68] as well as the work in 1997 entitled: Tabu Search [69].

The basic **procedure of tabu search** is the following:

---

```

begin
    initialize tabu list
    generate randomly initial solution  $S_C$ 
    evaluate  $S_C$ 
    repeat
        generate all neighbourhood solutions of solution  $S_C$ 
        find the best solution  $S_X$  in the neighbourhood
        if  $S_X$  is not tabu solution then  $S_C = S_X$ 
        else if "aspiration criteria" is fulfilled then  $S_C = S_X$ 
        else
            find best not tabu solution in the neighbourhood  $S_{nt}$ 
             $S_C = S_{nt}$ 
        Update tabu list
    until (terminate-condition)
end

```

---

In TS a memory, a so called tabu list, is used where the already visited solutions are stored. The solutions are forbidden (tabu) as new solutions, as long as they are in the

tabu list. This tabu list is used to escape local optima, by avoiding cycles which would occur and force the exploration of new areas of the search space as well. This is done by accepting the best not tabu solution, also if this solution is inferior to the actual solution, and consequently leaving the local optimum. There are some considerations that have to be taken into account using tabu lists. The length of the tabu list is critical. A too small length will lead to cycles, while a too long one will restrict the search [64].

#### *Aspiration criteria*

The idea behind aspiration criteria is that a very interesting solution from the tabu list is overridden and the move can be made to this solution [63].

#### Simulated annealing

In 1983 Kirkpatrick *et al.* first proposed simulated annealing (SA) to applications of combinatorial optimization, as one of the earliest metaheuristic [70]. SA is memory less. The idea behind SA is based on the analogical physical process from the metallurgy field [63]. Meaning the thermodynamic process of heating a raw material until it melts, followed by temperature reduction to grow a crystal. This temperature reduction should not be done too quickly, if not the crystal gets irregularities. A detailed listing of the analogues of the physical process and the optimization problem can be found in Michalewicz [63].

The **procedure of simulated annealing** is the following:

---

```

begin
     $t = 0$ 
    initialize  $T$ 
    select a current string  $v_c$  at random
    evaluate  $v_c$ 
    repeat
        repeat
            select a new point  $v_n$  in the neighbourhood of  $v_c$ 
            if  $\text{eval}(v_c) < \text{eval}(v_n)$  then  $v_c = v_n$ 
            else if  $\text{random}[0,1) < e^{\frac{\text{eval}(v_n) - \text{eval}(v_c)}{T}}$  then  $v_c = v_n$ 
        until (termination-condition)
         $T = g(T, t)$ 
         $t = t + 1$ 
    until (halting-criterion)
end

```

---

The annealing process:

In SA the probability to accept another (even worse) solution is defined:

$$P = \frac{1}{1 + e^{\frac{eval(v_c) - eval(v_n)}{T}}} \quad (6.6)$$

The temperature parameter  $T$  changes over time. Starting with a high value of  $T$  (here with similarities to random search) and decreasing it gradually. When  $T$  decreases the probability of accepting a worse solution is lower (compare 6.6). A suitable value for  $T$  has to be elected, not too high and not too low. Besides the initial temperature, the parameters for the cooling ratio  $g(T, t)$ , the termination-condition and the halting-criterion have to be set up [63].

### Genetic algorithm

Genetic algorithm (GA) is a search heuristic of the group of evolutionary algorithms.

The basic structure of the **procedure of evolutionary algorithms** is the following:

---

```
begin
   $t = 0$ 
  initialize  $P(t)$ 
  evaluate  $P(t)$ 
  while (not termination-condition) do
    begin
       $t = t + 1$ 
      select  $P(t)$  from  $P(t - 1)$ 
      alter  $P(t)$ 
      evaluate  $P(t)$ 
    end
  end
```

---

GA goes back to Holland, who has studied besides of combinatorial optimization the self-adeptness in biological processes [71]. GA tries to imitate the evolution of natural selection and natural genetics, where other (better) populations are developed. Therefore, GA generates at each iteration a population of solutions, which are called individuals. Matching of these solutions is done using the basic operators of GA survival, crossover and mutation. Using survival a solution is carried out to the next iteration without change.

Crossover combines two parent solutions and mutation changes parts of a solution [63], [64].

There exist several extensions of genetic algorithms. *Offspring selection* (OS) is one that extends the algorithm with an additional selection process after the recombination. This is done to ensure the success of the recombination. Besides the fitness of the parents also the fitness of the offspring is evaluated and compared with the parents fitness. Thus, by using *offspring selection genetic algorithm* (OSGA)<sup>31</sup> just those offspring, which have higher performance than their parents are taken to the next generation. A new parameter, a so called *success ratio* ( $SuccessRatio \in [0,1]$ ) is established in order to ensure that just those parents solutions which have a high number of successful offspring (*success ratio*) proceed. At the beginning the offspring just have to outperform the weaker parent. This is increased during the evolution to the better parent. One advantage of using OS is to avoid premature convergence using an advanced selection mechanism [72], [73].

#### The simulation noise

In general metaheuristic techniques do not address the simulation noise from simulation-based optimization [64]. Thus, the calculated values of the evaluation or target function

$F(x)$  and  $F(y)$  are just an estimation  $\hat{F}(x)$  and  $\hat{F}(y)$  of the underlying simulation.

Therefore  $\hat{F}(x) < \hat{F}(y)$  does not guarantee  $F(x) < F(y)$ . This leads to the question how this simulation noise can be handled when metaheuristics are used. For instance, in tabu search recently visited solutions are not revisited again, also when they are set tabu wrongly. Moreover, in GA unfit solutions may reoccur due to simulation noise, however here randomness may be beneficial to a certain level. Thus, this question of how simulation noise can be addressed (preferable with little computational effort) is not entirely solved yet and will need further research [64].

### **6.3 Optimizers - Optimization packages**

On the market exist several optimization packages (commercial and not) that can be coupled to a simulation. Law and Kelton [1] provide a table where commercial optimization packages are listed together with simulation software products (Table 6.2).

---

<sup>31</sup> A specification as well as the algorithm parameters from HeuristicLab can be found in <http://dev.heuristiclab.com/trac/hl/core/wiki/OSGA>

Table 6.2: Optimization packages for SBO, according to Law and Kelton [1] and extended.

<b>Optimization Package</b>	<b>Vendor</b>	<b>Simulation Software (Platform)</b>	<b>Mainly Used Search Procedures</b>
<b>AutoStat</b>	AutoSimulations, Inc.	AutoMod, AutoSched	Evolution strategies, genetic algorithms
<b>OptQuest</b>	Optimization Technologies, Inc.	Arena, MicroSaint, QUEST, Taylor, Enterprise Dynamics	Scatter search, tabu search, neural networks, genetic algorithms
<b>OPTIMIZ</b>	Visual Thinking International Ltd.	SIMUL8	Neural networks
<b>SimRunner</b>	PROMODEL Corp.	MedModel, ProModel, ServiceModel	Evolution strategies, genetic algorithms
<b>Optimizer</b>	Lanner Group, Inc.	WITNESS	Simulated annealing, tabu search
<b>ISSOP</b>	Dualis GmbH IT Solutions	Plant Simulation, Arena, Enterprise Dynamics, MATLAB	Evolution strategies, genetic algorithms, quasi gradient-methods
<b>HeuristicLab</b>	Heuristic and Evolutionary Algorithms Laboratory	SiRO, AnyLogic, among others (.NET based)	Evolution strategies, genetic algorithms, and other metaheuristics

In addition, some simulation packages have already included optimizers. However, they are not as powerful as optimization packages. For instance, Plant Simulation has an experiment manager that automatically generates scenarios using GA and neural networks. Another tool is IBM's ILOG which comes with several products in the field of optimization, including the JRules BRMS<sup>32</sup> that provides simulation.

### Heuristic Lab

HeuristicLab is an optimization framework for heuristic and evolutionary algorithms [74]. It has been developed since 2002 by the Heuristic and Evolutionary Algorithms Laboratory

<sup>32</sup> JRule BRMS: [www-01.ibm.com/software/integration/business-rule-management/jrules](http://www-01.ibm.com/software/integration/business-rule-management/jrules)

(HEAL). This framework is open-source and .NET based. HeuristicLab provides a plugin system, which allows the extension of further algorithms and optimization problems. One goal of HeuristicLab is the improvement of the reusability and flexibility of optimization algorithms, however the framework provides a lot more functions. Through a graphical user interface (GUI) parameterization and customization of an optimization strategy can be done. For that, several optimization strategies (including SA, TS and GA) are provided within the framework. A detailed specification as well tutorials of HeuristicLab can be found on the HEAL web page<sup>33</sup>.

## 6.4 Scheme of a simulation-based optimization study

The following figure 6.2 shows a cycle which describes the scheme of the SBO study. The classical way of SBO is presented, where the simulation can be seen as the objective function in classical optimization problems. A rating framework as in chapter 5 is used.

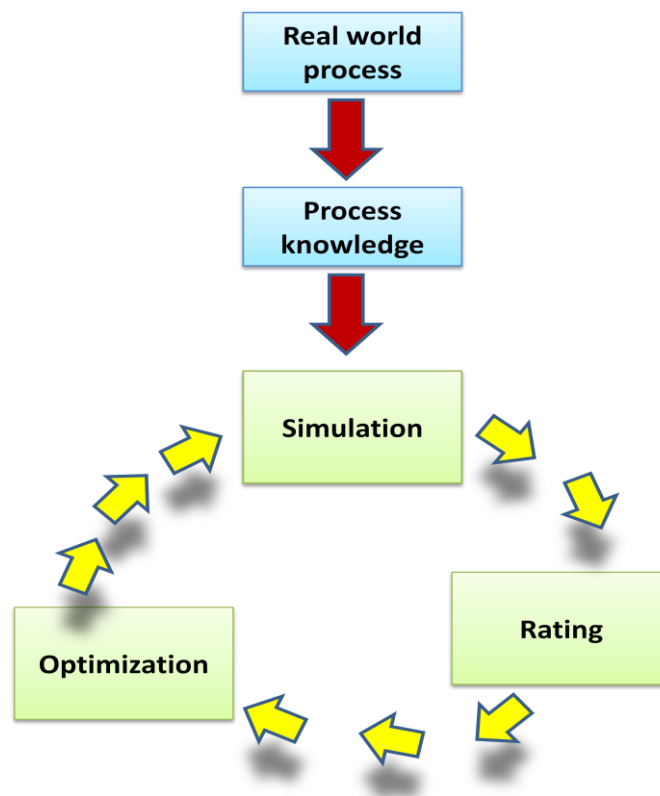


Figure 6.2: Cycle of a simulation-based optimization study.

<sup>33</sup> HeuristicLab: <http://dev.heuristiclab.com/trac/hl/core>

Based on a real word process, which may be of stochastic nature or not, the necessary process knowledge has to be acquired. With the knowledge of the underlying process it is possible to create a simulation model, which represents the real world process accurately. After having performed a simulation run, the simulation returns a result, which may be either output parameters, statistical or other output data. This output is the result of the input data and the probably stochastic nature of the process.

In a next step the obtained result has to be rated. On behalf of that, a rating framework is used to bring all the output data of the simulation together to a fitness value. Of major importance is that the underlying rating model returns an accurate fitness value.

And finally there is the optimization part. The optimization gets the fitness value and takes it as a measure of the quality of the last performed optimization, which can be a set of lists, priority rules or something else. Thus, the optimizer takes into account if there is an improvement in the result (the fitness value) or not, and starts a next iteration beginning from the simulation again.

Before the first simulation run, a random initial solution is elected for the optimization. This is the best way to start.

After having performed  $n$  Simulation runs and the SBO study has finished, analysis of the obtained information can be done. This may include the utilization rates of resources (either machines or workers), the length of queues and queuing times, and more specific data. Moreover bottleneck analysis, layout and structure analysis can be performed.

## **6.5 Application areas for simulation-based optimization**

The use of SBO can be beneficial in various areas. According to Fu the main application of SBO is in the areas of manufacturing, business processes, and computer and communications networks [13]. Figure 6.3 shows an overview of applications in the manufacturing area.

The applications of SBO in manufacturing can be found in the short-, in the medium- and in the long term as well. In the short term, these are basically concerns about lot sizes and order sequences. Together with work schedules and the order release the SBO in short term (hours, days) tries to increase the planning precision. In the medium (weeks, month) and in the long term (month, years) the aim is to increase the planning reliability. This is achieved with concerns on order planning, inventory and buffer levels and strategic planning where the number of machines, personal qualifications, concepts and structures of the factory are established.

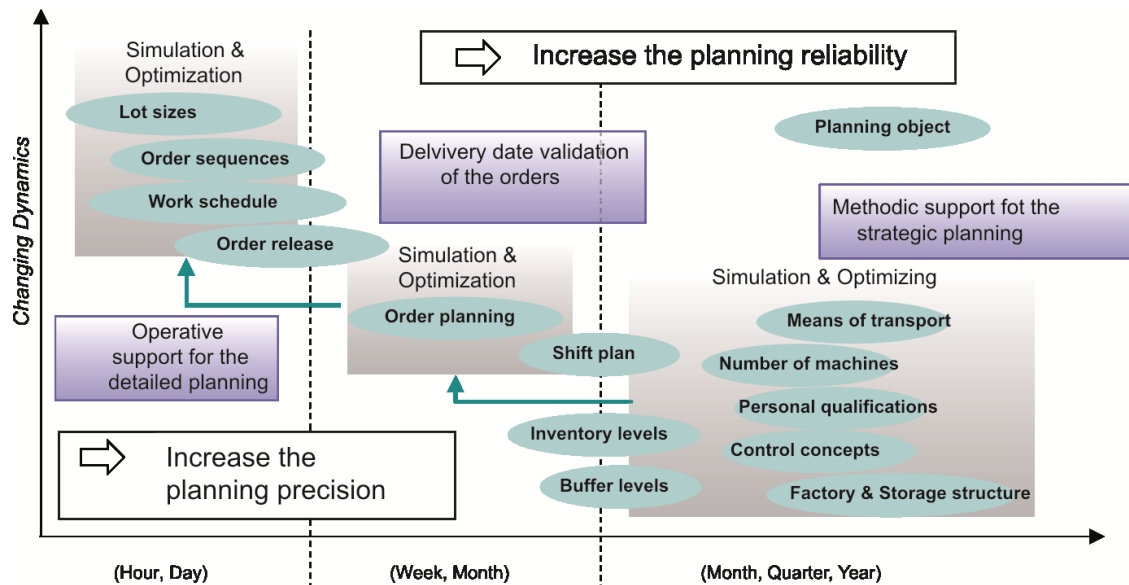


Figure 6.3: Application areas for simulation-based optimization, modified from März [9].

### Selected applications and works in the field of SBO

In literature exist plenty of applications where simulation is coupled with optimization. Just to give some demonstrative examples a few of them are mentioned in the following:

- A technical university and a co-operation partner from the ship building developed an approach to couple a discrete-event process simulation with metaheuristic optimization. They used Tecnomatix Plant Simulation as simulation tool and ISSOP from Dualis-IT as optimization tool [21].
- Supply Chain Systems have been modelled using SBO [75], [76].
- Several successful performed SBO case studies from the industry can be found in März *et al.* [9].
- In the following chapter 7, a SBO study in the area of manufacturing is presented in a detailed way.



*„If there's more than one possible outcome of a job or task, and one of those outcomes will result in disaster or an undesirable consequence, then somebody will do it that way."*

**Murphys Law**

## 7 Prototype application of a simulation-based optimization study

### A case study from the plastic processing industry (casting)

---

#### 7.1 Problem identification

##### Project partners

In this project<sup>34</sup> (project name: Procomposite) several institutions are involved. The Profactor GmbH as a private research institution, as well as two public research institutions, namely the Chair of Production and Operations Management (POM)<sup>35</sup> from the University of Vienna and the HEAL from the Upper Austria University of Applied Sciences participate. Further, the Schneegans-Silicon GmbH<sup>36</sup>, the ABF-Industrial Automation GmbH<sup>37</sup> are industry partners within the projects. The ASMA GmbH<sup>38</sup>, a company in the plastic processing industry is the field of study. Additionally, Arbeitsleben Gruber KG<sup>39</sup>, a company which works in the area of work ability and psychology is involved.

---

<sup>34</sup> A short synopsis of the project can be found on the web page of the programme:  
<http://www.fabrikderzukunft.at/results.html/id5764>

<sup>35</sup> Chair of Production and Operations Management: <http://prolog.univie.ac.at/>

<sup>36</sup> Schneegans-Silicon GmbH: <http://www.schneegans.com/start.php>

<sup>37</sup> ABF Industrial Automation: <http://www.abf.at/>

<sup>38</sup> ASMA: <http://www.asma.at/>

<sup>39</sup> Arbeitsleben Gruber KG: <http://www.arbeitsleben.at/cms/>

This project is part of an international EUREKA<sup>40</sup> project (project number E!3571). The Austrian contribution was financed by a grant from the FFG within the programme line FoT. The final report will be soon publically available on the web page of FoT programme [77].

### **The company**

The production, of the already mentioned ASMA GmbH located in Weitra, Austria, is the field of study. Confidential data from the company, its client and their processes is omitted or sometimes given in a simplified or generalized form here.

ASMA employs more than 100 people. Around fifty percent of them work directly in the production process.

The products, the produced quantities (from 1 to 100.000), the production processes, the materials as well as the application areas of the products are quite diverse. ASMA mainly processes plastics in all kinds. Their produced products can be classified into four product groups: wheels and rollers, moulded parts, coatings and specialties [78].

The produced products are used in areas like metal processing, glass production, packaging, transport and handling, winter service, infrastructure and other specialties. In the following some of the products from the ASMA GmbH are illustrated.



Figure 7.1a: Illustration of different products I.<sup>41</sup>

---

<sup>40</sup> EUREKA is an initiative for application oriented research and development in Europe. Further information about EUREKA can be found on the EUREKA network: <http://www.eurekanetwork.org/>

<sup>41</sup> Source: ASMA GmbH.



Figure 7.1b: Illustration of different products II.<sup>42</sup>

The used production technology includes different machinery of injection moulding as well as casting by hand. The used materials are varieties of different polyurethane formulations for different applications (e.g., Asmaprene, Vulkollan, Asmaprene) [78].

### **The production process description and aims of the study**

According to the product, the casting process can be done by machines or by hand and either with hot or cold plastic. In some circumstances the product can be produced by more than one machine or in more ways, but in general the machine sequence is order dependent. Thus, this problem is a *job shop* problem. If the routing of the orders is arbitrary, we talk about an *open shop* and if the routing is identical for all the jobs it is a *flow shop* [79].

The reduction of cycle times together with the reduction of tardiness was identified by the company as a main concern. Historically, the average cycle times are around 3 to 4 weeks. The effective production time is just a few hours, but between process steps the most products have to rest overnight either in heating coils or in cooling rooms. A specific aim is to reduce these cycle times by around 25%. This decrease would notably improve the competitiveness of the company.

The production process of the product includes several stages (e.g., the casting process, drying, finishing with quality control, among others). Each of these stages is formed by several work steps, for instance the casting process includes 13 work steps that are considered in the simulation model.

The result of the analysis of the processes and the corresponding work steps were specified in a document of over 30 pages. This document shows the big dimension of the problem that has to be studied.

During the development of a simulation study several problems can occur. Due to various circumstances, it is often difficult to make a realistic estimation of how long a simulation

---

<sup>42</sup> Source: ASMA GmbH.

study will take. A critical point might be the quality of the provided data (exhaustive data) from the company.

Another aspect of this project is that a holistic view on the production process is included. Therefore, the working plans should consider work ergonomics and other sustainable aspects like resource use, emissions and waste among others. The project partner Arbeitsleben Geißler KG prepared a working paper of around 40 pages, which consider the questions of labour psychology in the production process. This working paper is based on the principles of the human criteria for work processes published by Hacker [80].

The overall aim of the project is to provide ASMA a prototype implementation, which can be validated in real production conditions and further used in the daily operative production process.

## **7.2 Tools and methods**

### **Selection of the tools**

To perform a SBO study, adequate simulation, optimization and rating tools have to be chosen. It has to be taken into account, that the complexity of this *job shop* problem is *NP-hard*. As simulation tool, SiRO from Profactor is used. As optimization tool, HeuristicLab from HEAL is used. The rating framework which is used for the evaluation of the simulation output was introduced in chapter 5. This study follows the presented scheme of SBO shown in chapter 6.

### **Optimization methods**

Two different optimization methods are performed within the project to obtain optimal production schedules. One is the method of the Resource Constraint Project Scheduling Problem (RCPSP). Using Resource Constraint Project Scheduling (RCPS) the start time for each activity is determined according to the resource constraints. In this case, the optimization results are used for the configuration of the simulation, to check whether the obtained optimization results are feasible or not. Then, from the output a fitness value is calculated. This RCPS approach using adaptive large neighborhood search (ALNS) was done by the project partner POM. This optimization approach is not primary part of this work, but is mentioned for completeness. An interesting review about RCPS can be found from Brucker *et al.* [81].

The second optimization method is a priority rule based dispatching strategy. Therefore, several priority rules are provided in the simulation. On the other hand, also complex priority rules can be formulated. To obtain such complex priority rules, an offspring selection genetic algorithm (OSGA) is used. This approach was done by the project partner HEAL using the HeuristicLab framework. In this work the main focus is on coupling the simulation with the optimization with such complex priority rule.

### Priority rules

As priority rules, both scheduling and dispatching rules are understood. Priority rules have been used successfully for PPC. Their performances were intensively studied by means of simulation experiments.

In the following the principles of priority rules are presented: All jobs which are ready to be performed are placed into queues before the corresponding working station. In these queues they wait to be processed. Figure 7.2 illustrates this queuing process (compare also chapter 2 and chapter 3 of this work).

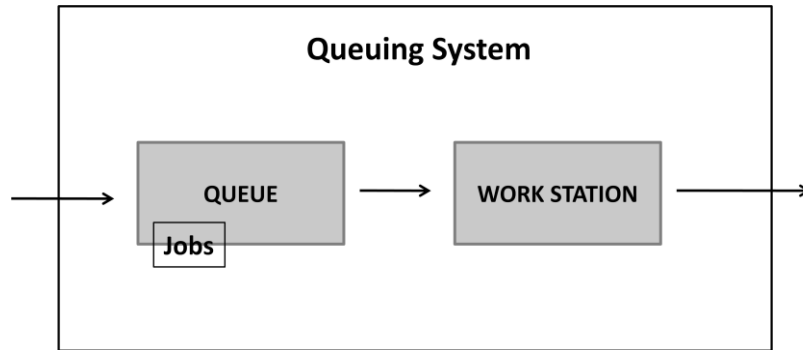


Figure 7.2: Orders queuing to be processed.

After being processed in the work station, the jobs follow their way through the production system until they leave (this happens when their schedule is entirely processed).

According to their priorities, the jobs are selected from the queue one by one. To do so, every one of the  $i$  jobs ( $i = 1, \dots, n$ ) queuing for the  $k$  work station ( $k = 1, \dots, m$ ) gets a priority value  $Z_i(t)$  assigned. The priority value  $Z_i(t)$  is calculated (by applying the priority rule) every time  $t$  when a decision has to be made. Thus, the job with the highest priority value  $Z_i(t)$  is chosen to be performed. In the following, several of well known priority rules [(a)-(m)] to calculate these priority values for a work station  $k$  are presented [79], [82].

*Notations:*

$x_i$	random value assigned to job $i$
$t$	current time
$r_i$	arrival time of job $i$
$p_i$	processing time of job $i$
$d_i$	due date of job $i$
$rt_i$	processing time remaining of job $i$
$N_i(t)$	number of waiting jobs $i$ at time $t$
$Y_i(t)$	total work ( $\sum p_i$ ) of waiting job jobs $i$ at time $t$
$Y_i'(t)$	including that work, that is expected to arrive in a defined period

---

(a)	RANDOM	$Z_i(t) = x_i$
(b)	FIFO (first in first out)	$Z_i(t) = r_i$
(c)	SPT (shortest processing time)	$Z_i(t) = p_i$
(d)	LPT (longest processing time)	$Z_i(t) = -p_i$
(e)	EDD (earliest due date)	$Z_i(t) = d_i$
(f)	ALL (smallest allowance)	$Z_i(t) = d_i - t$
(g)	MST (minimum slack time)	$Z_i(t) = d_i - t - rt_i$
(h)	CR (smallest critical ratio)	$Z_i(t) = (d_i - t) / rt_i$
(i)	SRPT (shortest remaining processing time)	$Z_i(t) = rt_i$
(j)	LRPT (longest remaining processing time)	$Z_i(t) = -rt_i$
(k)	NINQ (least number of jobs in queue of its next operation)	$Z_i(t) = N_{i,k+1}(t)$
(l)	WINQ (least total work in queue of its next operations)	$Z_i(t) = Y_{i,k+1}(t)$
(m)	XWINQ (least total work (WINQ) including soon expected work)	$Z_i(t) = Y_{i,k+1}'(t)$

Other combinations and modifications of these priority rules are also possible; e.g., a combination of the CR and SPT, the modification of the due dates, and so on.

According to Lödging [19] these priority rules have been proven to fulfil different objectives. It is suggested to use FIFO, MST and EDD in order to improve the delivery reliability. For the improvement of the service level, the CR is appropriate. To improve the performance and efficiency, XWINQ and other priority rules that are based on set-up times are beneficial. And finally the SPT, is proper to achieve low inventory levels, low cycle times, low average tardiness of orders and high delivery reliability.

Priority rules can also be much more sophisticated. For instance, priority rules for situational dispatching that are generated by using metaheuristics. Such complex priority rules will be performed in the following, using HeuristicLab. These priority rules can include formula fragments of the actual queue length, machine downtimes, due dates, processing times and other system conditions connected with mathematical and logical operators. A simple example of such a complex priority rule is shown in figure 7.3.

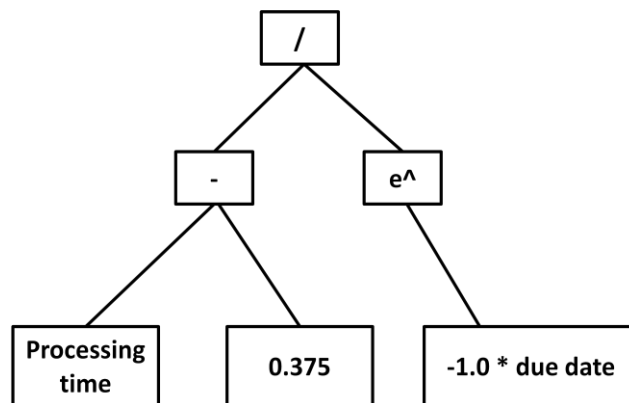


Figure 7.3: Example of a complex priority rule.

As every problem is different (different parameters, attributes, jobs, machines, etc.) there is no best *a priori* priority rule known. However, it has been shown that such sophisticated priority rules outperform classical priority rules (e.g., (a) – (m) ) for most problems [82].

### 7.3 Modelling and implementation

The prototype implementation of the project was done by a cooperation of the project partners. As programming language, C# was elected for the entire implementation. This

programming language was considered beneficial due to several circumstances (e.g., all the tools support this language; high performance and fast results are needed; the communication between the tools, the data base - server connection and others). The process logic is modelled using SiRO.

### **Data model**

In order to obtain the necessary data (as a data model), at the beginning an analysis of the information provided by the company was done. The obtained data model provides the necessary information to build the simulation model. ASMA uses APplus<sup>43</sup> as ERP system. The 502 tables from the ERP system were analyzed intensively and the relevant data was extracted. This information mainly includes data from orders, bills of materials and working plans which may contain several working steps. Furthermore, information about workers and machines, as well as the work schedule, are used for the data model.

### **Data access**

At the beginning of a simulation run a data access is performed. Here real data is provided, which is used to initialize all simulation model objects. This data access is done by the use of *Linq2SQL* queries.

### **Routing**

The class *OrderPart* contains all the orders. These orders can be seen as the entities that flow through the production system. Every one of these orders is associated with a bill of material and a working plan. These working plans define the way the orders move through the system (depending on the working steps of the working plans). Thus, the orders go from one working step – which in general represents a machine or a worker – to the next working step until there are no more working steps left to be processed. Then they leave the system. The logic that implements this is contained in the class *PileAdmin*.

### **Serving the tasks**

SiRO provides a template *TaskMachine*. This template contains tasks that represent jobs which are served on the work stations. Different tasks represent different jobs (e.g., set-up, processing, repair, among others).

---

<sup>43</sup> APpuls: [www.ap-ag.com](http://www.ap-ag.com)



## **The workers**

The workers can have 2 different states and 4 sub-states:

- Unavailable: The worker is not available at the moment.
- Available: The worker is available in one of the following sub-states:
  - Setup: Setup time before starting a task.
  - Working: Serving a task.
  - Pausing: A scheduled pause at convenient times.
  - Bored: The worker is forced to pause due to other circumstances (e.g., machine failures, no more tasks to serve, etc.).

A qualification matrix of the workers defines for which task which worker is qualified. Further, this matrix contains information about the experience and skills of the workers as numerical values. The state changes of the workers are performed using this qualification matrix. `IBrain` and `IWorker` are two interfaces that implement this logic.

## **Simulation output**

In the *Sink*, the simulation collects all the relevant information. This collected data is then provided to the rating model. To represent the simulation results, a Gantt chart is provided. In this Gantt chart the workers are associated to the different work stations and take the jobs according the established priority rule. In the Gantt chart, the states of each worker during time can be seen as well (e.g., pausing, etc.). Further, a basic graphical animation showing the factory layout with the workers moving to the work stations is provided for basic priority rules.

## **Data representation and stochastic**

All the necessary data for a simulation study has to be represented in an adequate and realistic way. Thus, the machine downtimes and failures, arrival of workers, set-up and processing times, personal allowance and others have to be represented in stochastic distributions. For instance arrivals of workers are described by a triangular distribution; set-up times can be represented by a uniform distribution and machines downtimes and failures can be described using an exponential distribution.

A further point that has to be considered is the fact that the casting process is quite error-prone, which leads to output variability. The rejection rate is around 10 percent, depending on the climate (especially the humidity), the experience of the worker, the used production machine and other not feasible factors. This error-prone is represented in the simulation

model through a binominal distribution. Therefore, the binominal distribution returns the probability  $p$  if a produced order is erroneous.

The stochastic influence in the simulation model leads to uncertainty of the simulation output. The certainty of the simulation output can be improved using a sufficient high number  $n$  of performed simulation runs. This  $n$  mainly depends on the variance of the different simulation runs.

In the annex of this work several relevant distributions for simulation are shown, including binominal distribution, triangular distribution and exponential distribution.

## Two project implementations

Two slightly different implementations, namely the user interface (UI) and the console application, are implemented. The UI provides a simple visualisation of the progress of the simulation. Additionally, several predefined priority rules e.g. FIFO, SPT, LPT; EDD as well as a user defined complex priority rule can be chosen. Then, the simulation study is performed using the elected priority rule. On the other hand, the console application is used to perform the before described SBO study. Here a complex priority rule is generated using HeuristicLab. Applying this approach the optimizer (HeuristicLab) is coupled with the simulation (SiRO).

## 7.4 Coupling the tools (WCF-Service)

The communication between simulator SiRO and optimizer HeuristicLab has to be specified. Generally, this communication may not be always an easy job. In this project it was done by using the Windows Communication Foundation (WCF<sup>44</sup>) – service. The used structure of the communication is illustrated in figure 7.4.

For the communication, interfaces in SiRO and HeuristicLab are used. Through a *WCF-client*, the generated complex priority rules are provided to the WCF-service. Then the simulation takes them and performs the simulation run (applying the before generated rule). After that, the rating framework calculates the fitness value. This fitness value is then again provided through the WCF-service to the optimizer. A *rule parser* is needed in order that the simulation can interpret the priority rules as decisions during the simulation run.

---

<sup>44</sup> WCF is an application programming interface in the .NET framework.

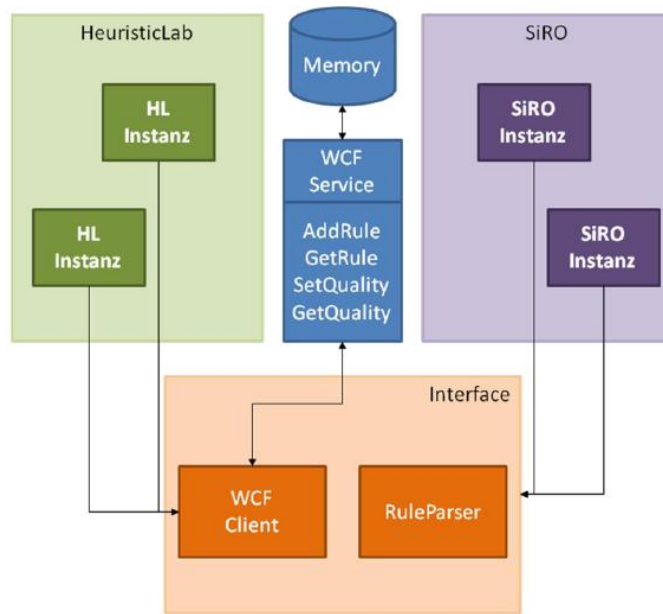


Figure 7.4: Communication between simulator and optimizer.<sup>45</sup>

## 7.5 Problem formulation from HeuristicLab

The HEAL project team has formulated a problem using the HeuristicLab framework. This problem formulation is provided in a so called *workbench* file, namely the *Cached External Evaluation Problem.hl* file.

### Basic Grammar

The problem formulation contains a specification of basic grammar of the problem to be solved. This basic grammar contains the symbolic expressions. The allowed mathematical and logical operations (e.g., +, -, \*, /, average, log, exponential, &, OR, etc.), constants and variables (e.g., qualification, queue length, setup time, due date, processing time, etc.) are defined. The exact formulation of the grammar of the problem from HeuristicLab is shown in figure 7.5.

The basic grammar can be easily changed and extended according the requirements of the process optimization.

<sup>45</sup> Source: Andreas Beham, Procomposite – FoT.

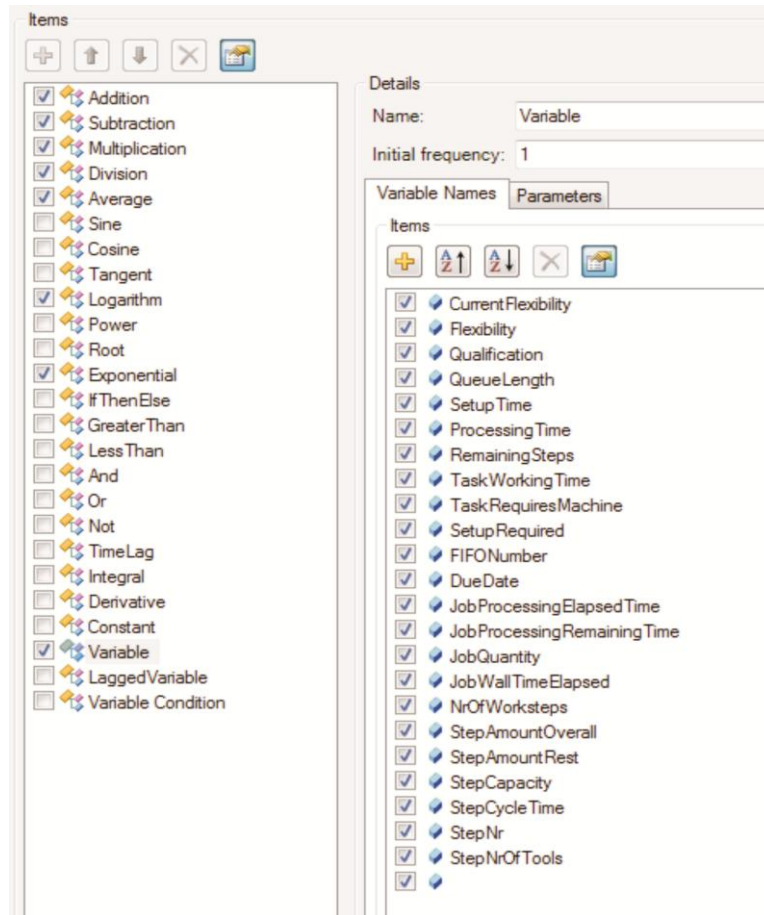


Figure 7.5: Grammar formulation of the priority rule.

On the right column of figure 7.5 the allowed variables of the complex priority rule are specified. Such a variable can be anything that is represented in the simulation model.

### Parameters of the OSGA

The OSGA which is implemented in HeuristicLab framework provides different parameters that can be specified. They include specifications about the mutation, crossover, selection and the OSGA specific success ratio (compare also chapter 6) among others. Moreover, the maximum tree height and tree size can also be specified within the framework. In the annex an exemplary parameter setting for the used OSGA can be found.

## 7.6 The rating process

In this SBO study, the rating of the simulation can be seen as the objective function in classical optimization problems. Using the rating framework, a rating model is constructed. This rating model represents the optimization objectives of the company. In a more technical way, the rating model represents the fitness function and the design of the penalty. Several different, sometimes opposite, indices have to be taken into account (compare also chapter 2 and chapter 5).

The rating model gets the simulation output data as input. Then, the rating model gathers the relevant indices, weights them and calculates a fitness value. Afterwards, this fitness value is provided to the optimizer. The setting of the rating model depends on the companies interests. As it was mentioned before a main objective for this study is to reduce the cycle times by about 25 percent, together with the improvement of the adherence to the delivery dates.

### The rating model

For the rating process, a multi-stage rating model is used. Figure 7.6 illustrates a basic form of a used rating model.

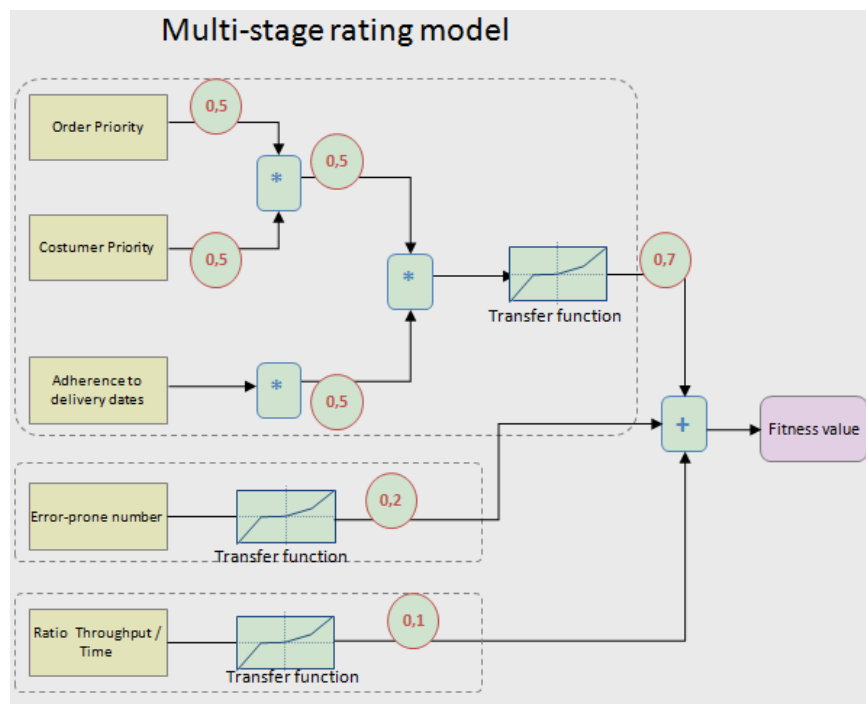


Figure 7.6: Exemplary rating model.

The layout of the rating model is tree based and might consist of several stages. In this case two stages are used. In stage 1 the orders can have different priorities (e.g., depending on the urgency, the contract volume, etc.). The customers also can have different categories (e.g. customer class A, B or C). Thus, the adherence to delivery dates are multiplied and weighted with these priorities, before they are transferred in the respective transfer function. This transfer has also been done for the error-prone number, which represents the defective products, as well as the ratio of the throughput and the needed time. In stage 2 each index or indices group is weighted and then added to the final fitness value. This can be done, since the indices are transferred into comparable values in stage 1.

To obtain the overall fitness value the rating model has to be performed for every order. This rating model can be extended easily including several other criteria (e.g., the criteria of energy and production resource usage, wastage and pollutant emissions, stress load of workers, work monotony, etc.). Also different rating scenarios are possible. For instance, the weightings of the indices or penalty assignment in the transfer functions can be changed.

## 7.7 Test cases

In the following, two cases of performed simulation studies are presented. For that, the existing prototype implementation is used. The first test case is simulation with basic priority rules, which are already predefined in the simulation model. The second test case uses complex priority rules generated with HeuristicLab using OSGA.

### Simulation with basic priority rules

To perform simulation with basic priority rules the UI application of the simulation is executed. First of all, the database connection has to be established. Figure 7.7 illustrates this.

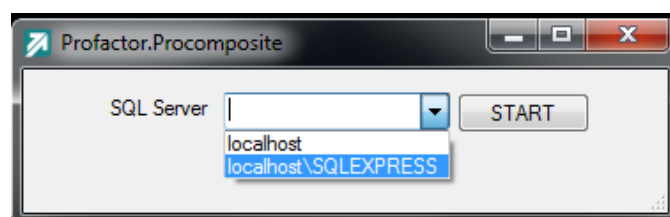


Figure 7.7: Establishing of a database connection.

Having performed the database connection, the UI of the simulation is opened. Then, the desired priority rule has to be chosen and the simulation starts using the selected priority rule (EDD in this case).

The obtained result of the simulation run is provided in a Gantt chart where all the workers are allocated to different work stations. Figure 7.8 illustrates such a Gantt chart for different workers, where different colours represent the different work stations. This illustrates the allocation of the workers to the work stations. Additionally, the view of the Gantt chart can be switched from a worker representation to a machine representation.

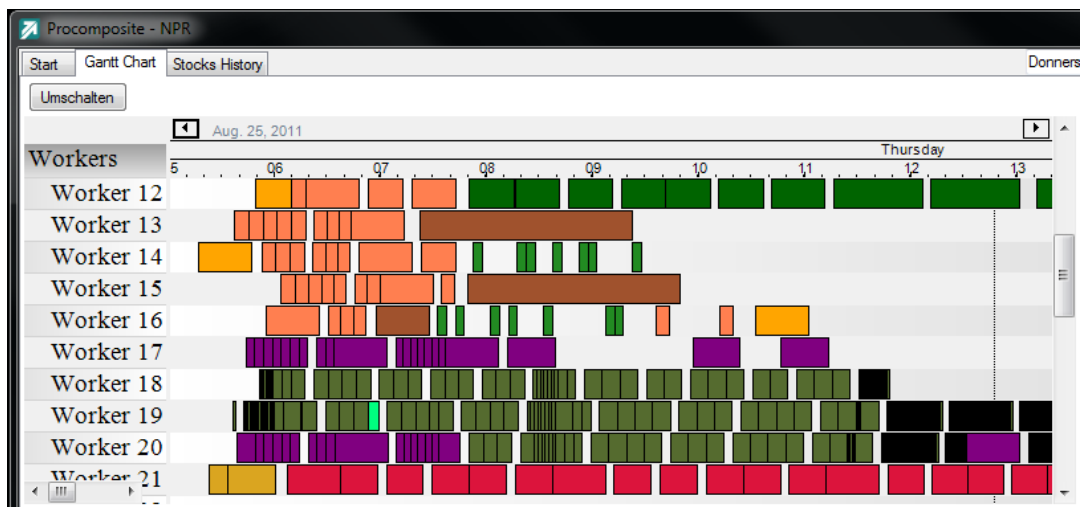


Figure 7.8: Screenshot of the Gantt chart representing the workers.

As always, when priority rules are used, the orders are picked up from the queues according to the priority rules and then processed. For the Gantt chart illustration this means that just the workers and work station are illustrated and not the different jobs. The following figure 7.9 shows the provided simple animation of the simulation run.



Figure 7.9: Animation of a simulation run.

### Simulation using HeuristicLab with an offspring selection genetic algorithm

To perform the atomized SBO study the specified .hl file with the specified problem formulation (compare section 7.5) has to be opened in the HeuristicLab framework. Then, the console application of the simulation has to be executed. Figure 7.9 provides a screenshot of the logging of a simulation run during the study.

```

file:///C:/Profactor_27_05/Profactor.Procomposite/Profactor.Procomposite.Console/bin/Debug/Pro...
19.01.2010 11:52:40 - OrderSink has received Profactor.Procomposite.Data.OrderPa
rt
19.01.2010 12:00:18 - OrderSink has received Profactor.Procomposite.Data.OrderPa
rt
19.01.2010 13:39:20 - OrderSink has received Profactor.Procomposite.Data.OrderPa
rt
19.01.2010 13:47:20 - OrderSink has received Profactor.Procomposite.Data.OrderPa
rt
19.01.2010 13:59:54 - OrderSink has received Profactor.Procomposite.Data.OrderPa
rt
20.01.2010 05:28:09 - OrderSink has received Profactor.Procomposite.Data.OrderPa
rt
20.01.2010 05:44:37 - SetupSink has received D16/5.3x5
20.01.2010 06:01:05 - OrderSink has received Profactor.Procomposite.Data.OrderPa
rt
20.01.2010 06:34:37 - OrderSink has received Profactor.Procomposite.Data.OrderPa
rt
20.01.2010 07:23:13 - OrderSink has received Profactor.Procomposite.Data.OrderPa
rt
Simulation done.
Sink received 25391,00 items.
Timeliness: 92501
Throughput: 25391
AvgLeadTime: 175,586814233634

```

Figure 7.10: Screenshot of the logging during the study run.





In order to be able to analyse a complex priority rule, its allowed complexity was decreased by the Procomposite project team. The obtained priority rule of that is presented in figure 7.12.

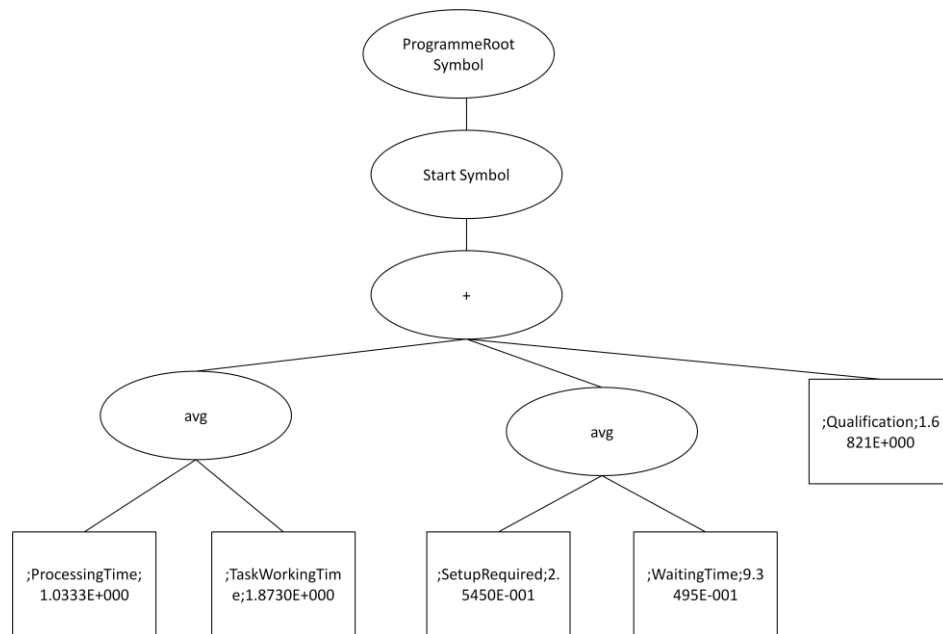


Figure 7.12: Complex priority rule with decreased allowed complexity.<sup>47</sup>

These obtained priority rules contains 5 variables which form the rule together with the mathematical operations of average and addition. These 5 variables that are considered in this priority rule are:

- The processing time of an order
- The task working time, which discriminates the processing times of the different tasks of an order
- If a set-up is required or not
- The accumulated waiting time of an order in the queue
- The qualification of a worker on the work station

Interestingly, for using timeliness as main optimization criteria, the complex priority rule (figure 7.11) and the complex priority rules with decreased allowed complexity (figure

<sup>47</sup> The values were obtained by the Procomposite project team.

7.12) performed quite similar with an average overall fitness value of 797.393,34 and 797.436 respectively.

A final report of the Procomposite project entitled in German: *Nachhaltige Produktionsregelung* can be found on the web page of the FoT programme after its approval by the FFG [77]. In this final report, in which I have contributed, can be also found further information and performed analyses of the generated priority rules by the project team.

*“All truths are easy to understand once they are discovered; the point is to discover them.”*  
**Galileo Galilei**

## 8 Resume and Outlook

---

In this work a SBO study was performed in order to improve the operative scheduling and lot sizing of the production of a company in the plastic industry. Besides the classical indices, a holistic rating was used. The aim of holistic production is to see the production processes as whole and not just as each part separately. This is in contrast to Taylorism<sup>48</sup>, which proposes as basic idea to reduce the complexity of processes by dividing them into sub-processes [83], [84]. This also includes the separation of the process planning and the process execution.

In this study, the objectives as well as the elected operation point were also holistic. This means that, besides the technological and economical indices, other indices may be considered in the rating process. Further examples from the plastic processing industry are material and colour changes. These lead to losses of energy and resources during repeated heating of casting moulds or raw material.

For the development of this study the company provided real data from their ERP system. This data was analyzed, extracted and used to construct the simulation model. Then the optimization was performed coupling the simulator with an optimizer using two different optimization approaches. Both optimization approaches RCPS and priority rule based dispatching were performed separately. Besides this, another possible approach could have been a hybridisation of them. In such hybridisation the calculated schedule with RCPS is adopted by priority rules when stochastic influences appear. A second possible hybridisation could have been that the schedule calculated using RCPS forms part of the priority rule.

---

<sup>48</sup> Frederick W. Taylor (1856 – 1915) published in his work “The Principles of Scientific Management” (1911) the principles that are known today as Scientific Management or Taylorism.

The simulation output was evaluated using a rating framework which formulates a fitness function. A difficulty of this formulation is to represent the company objectives in an accurate and valid way. Often a coupled optimizer tries to get out maximum of the fitness function, by using all possible improvements. This is, of course, the aim of the optimizer. However, this can also lead to a not desired optimization (e.g., when the penalty for work pauses is too high or too low the workers will have too long pause or never; or when the penalty for non utilization of work stations is too high this work station will accumulate inventory). Consequently, the fitness function (represented by the rating model) has to be carefully formulated. The used rating model is online based rating. The advantage of using online rating is the possibility to infer the quality of the evaluated input parameters during the simulation run. Thus, unfeasible, invalid and poor solutions can be identified and discarded during runtime. However, it has been shown in practical use that such online ratings are quite time consuming. This is due to the many calculations that have to be performed during simulation runtime.

On the other hand there exists offline rating. Here, having performed the whole simulation replication the rating gets the data sets and evaluates them; which is faster in most cases. The success of SBO studies depends highly on the quality and the availability of data. Moreover, of major importance is the connection and exchange of data from the data storage, (e.g., an ERP system) to the simulation model and after having performed the SBO study back to the used data platform (Figure 8.1).

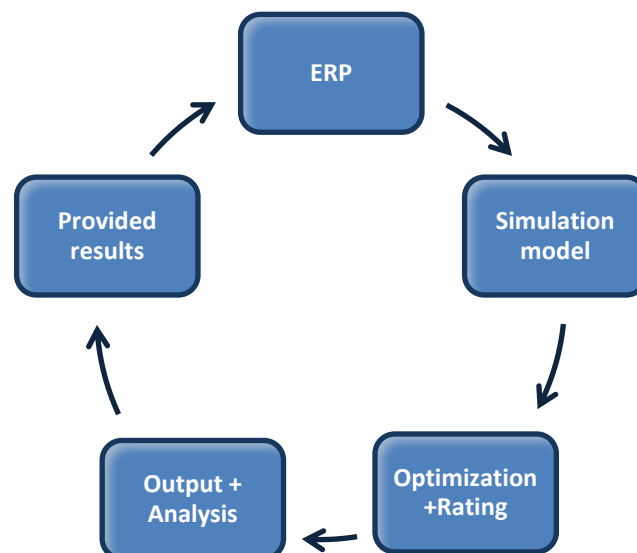


Figure 8.1. General overview.

Further, the obtained working schedule has to be provided in a suitable way (e.g., providing a GUI for all workers or work stations).

In addition, as in most cases SBO studies are quite time consuming, it is also important to obtain an applicable priority rule in an adequate time frame. For instance, the simulation model that was used to perform the presented SBO study takes around 5 seconds to perform a single simulation replication together with its rating process. Depending on the stochastic influence, the simulation has to be replicated  $n$  times for  $m$  optimizing iterations in order to generate a complex priority rule. Another important question is the robustness of generated rules (e.g. a generated rule is good on Mondays but not on Fridays).

There is no general best priority rule applicable for all producing companies. If not, the best rule depends on the main aims of the companies. Clearly, these aims may differ. Furthermore, within one company they may vary for different departments and may also change over time.

In different stages of the study several analyses can be performed. Many times analyses are beneficial to know which product brings the most turnover or at which production stage the most value is added to a product. In practice the factory management and the personal often do not know this. To obtain these information exist tools, like ABC-Analysis, value stream mapping, bottleneck analysis and sensitivity analysis, among others.

One very interesting result, which was found out during the study, is that there is a correlation between queue length and the stress levels of the workers. Thus, the workers that have higher backlogs of work (represented by longer queue length in the simulation model) feel higher level of stress and workloads. The length of queues can be easily observed in the simulation study. The stress levels were previously identified by questionnaires and interviews done by Arbeitsleben Geißler KG.

## **8.1 The use of simulation in the industry – outlook**

SBO methods have been proven as a good way to solve complex industrial problems in the field of PPC. Over the last 20 years the use of DES has been considerable increased. In a questionnaire made in the year 1996 just 5% of asked companies used simulation as a support tool. However, around 26% had the intention to use simulation in the future [85]. Nowadays exist plenty of simulation tools for various kinds of applications. Further, it is expected that the increasingly use of simulation will continue.

However, SBO is not a completely solved problem. A problem to perform successfully SBO studies is that still a lot of specific knowledge is needed and just few instructions in these concerns are available [1].

## **8.2 Outlook of the project**

This study, with its prototype implementation, is an ongoing research study. However, the Procomposite project has already been finished. A final report of the Procomposite project was submitted by the project partners. After the approval of the FFG, the final report will be found on the webpage of Factory of Tomorrow in the section of publications – sustainable development [77].

A follow-up project was approved by the FFG. This follow-up project includes further steps like the extension of the simulation model, the extension of the rating process, the improvement of the visualisation, a GUI implementation, a hybridization of the optimization approaches and further analysis, among others. At the end of the follow-up project, the prototype should be used in the operative production process by the company of study.

## Annex

---



## A. Simulation software survey

In the following an abstract of the simulation software survey from James Swain is given. It includes the most important simulation software for DES. The system requirements which are given in page 1 for each software package are omitted here. Page 2 to 7 from the survey are not given in the annex, but can be found in [37].

Software	Vendor	Typical Applications of the software	Primary Markets for which the software is applied
@RISK	Palisade Corporation	Risk analysis for Microsoft Excel spreadsheets using Monte Carlo simulation	Finance; Insurance; Oil/Gas/Energy; Pharmaceuticals; Manufacturing
@RISK 5.5	Palisade Corporation	@RISK performs risk analysis using Monte Carlo simulation to show you many possible outcomes in Excel	Finance/Banking, Energy & Utilities, Manufacturing, Insurance/Reinsurance, Healthcare Six Sigma , Academic
Analytica 4.2	Lumina Decision Systems, Inc	Investment, risk, decision, portfolio network flow analysis; systems dynamics; resource, R&D planning; organization simulation	Business, financial, public policy, energy, environmental, healthcare, defense, manufacturing, education, telecommunication
AnyLogic	XJ Technologies	Flexible general purpose simulation tool. Discrete Event, Agent Based and System Dynamics modeling	Healthcare, Logistics, Supply Chains, Manufacturing, Defense, IT, Pharmaceuticals, Marketing, Finance, Energy, Education
Arena Simulation Software	Rockwell Automation	A proven and easy to use general purpose DES software tool. The only boundaries are in the ability to describe the process.	Manufacturing, Six-Sigma, Packaging, Supply Chain/Logistics, Healthcare, Military/Defense, Service, Contact Centers, etc.
Blues Simulation System (Bluesss)	Stanislaw Raczynski	General purpose, discrete and continuous models	Education, research
Clinical Trials Simulator	ProModel Corporation	Automated simulation analysis and mathematical optimization of multiple clinical trial site and patient recruitment scenarios	Pharmaceutical Clinical Trials
CSIM for Java	Mesquite Software, Inc.	Models of processes and systems of processes	Education
CSIM20	Mesquite Software, Inc.	Modeling large, complex systems	Computer systems, networks, education
DecisionTools Suite	Palisade Corporation	Integrated risk, decision, and data analysis in Excel combining Monte Carlo simulation, decision trees, optimization, & more.	Finance; insurance; oil/gas/energy; pharmaceuticals; manufacturing; six sigma
Emergency Department Simulator	ProModel Corporation	Emergency Department throughput, staffing, patient flow, and efficiency analysis.	Emergency Department Performance Improvement
Enterprise Portfolio Simulator	ProModel Corporation	Web-based simulation analysis of multiple, simultaneous project/product plans across one or more portfolios of projects	Project and Portfolio Planning, Strategic Resource Capacity Planning; New Product Development, R&D, Scheduling
ExtendSim AT	Imagine That Inc.	Rate-based modeling functionality for manufacturers operating in a mixed-mode environment: batch process and discrete event.	Any industry that operates in a high-speed or high-volume arena: packaging lines, chemical processes, network traffic, etc.
ExtendSim OR	Imagine That Inc.	Message-based discrete event architecture to model processes involving physical or logical objects moving through systems.	Manufacturing and business modeling; communication systems, healthcare, six sigma, transportation, service, education, etc.
ExtendSim Suite	Imagine That Inc.	Professional 3D modeling of continuous, discrete event, and discrete rate processes.	When impressive presentations count. 3D modeling of manufacturing, logistics, business, government, education, engineering.
Flexsim	Flexsim	Process improvement, process optimization,	manufacturing, warehousing, distribution,

	Software Products, Inc.	capital investment justification, lean implementations	supply chain, and mining
Flexsim CT	Flexsim Software Products, Inc.	Container shipping terminal applications, yard stacking segregation strategies, quay crane allocation, dispatching	Container terminals
Flexsim HC	Flexsim Software Products, Inc.	Hospital ER and related healthcare/patient flow applications	Healthcare, emergency rooms, patient flow, patient services, lab, staff balancing
ForeTell-DSS	DecisionPath, Inc.	„Test drive“ decisions e.g., managing risks from terrorist threats, org.change, competitive mktg strategy, M&A transactions	Government (e.g. homeland security, DOD, civil sector policy), Pharma/Life Sciences, Financial Services
GoldSim	GoldSim Technology Group	Engineering risk analysis, including water resource management, mining, waste management, and aerospace mission risk analysis	Mining, Energy, Water Resource Management, Aerospace and Defense, Engineering Consulting, Radioactive Waste Management
Integrated Performance Modelling Environment (IPME)	Alion Science and Technology MA&D Operation	human performance modeling, manning analysis, human workload evaluation, product design	government, defense, manufacturing, staffing
MedModel Optimization Suite	ProModel Corporation	Design, plan, evaluate and improve processes of hospitals, clinics, and other healthcare systems to optimize performance	Hospitals, Clinics, Healthcare Systems, Medical Device Manufacturing and Sales
Micro Saint Sharp	Alion Science and Technology, MA&D Operation	human performance modeling,manufacturing,healthcare,service industry,military,business process reengineering,supply chains	human performance modeling,manufacturing,healthcare,supply chains
Portfolio Simulator	ProModel Corporation	Simulation and optimization analysis of multiple,simultaneous project/product plans across one or more portfolios of projects	Project and Portfolio Planning, Strategic Resource Capacity Planning; New Product Development, R&D, Scheduling
Process Simulator	ProModel Corporation	Lean, SixSigma, value stream mapping, process mapping, flow chart simulation, continuous process improvement	All
Project Simulator	ProModel Corporation	Enables project managers to more accurately predict outcomes of their project plans	Anyone who uses Microsoft Project
ProModel Optimization Suite	ProModel Corporation	Lean, SixSigma, capacity planning, cost analysis, process modeling, cycle time reduction, throughput optimization and more	Manufacturing and logistics, pharmaceutical, defense
Proof 3D	Wolverine Software Corporation	Provides 3D animation capabilities for use in discrete event simulation	Air Traffic control, transportation, manufacturing
Proof 5	Wolverine Software Corporation	Provides 2D animation capabilities for use in discrete event simulation	Air Traffic control, transportation, manufacturing
PSM++	Stanislaw Raczynski	General purpose, discrete and continuous models	Education, Delphi users
QMS	QuantMethods	Pedagogical tool for teaching quantitative methods, including simulation.	Higher education
REACT	MJC2	Real-time scheduling and simulation	All large logistics and transport operations.
Renque	Renque Corporation	General purpose discrete event simulation software	Transport, Communications, Engineering & Design, Business modelling
Risk Solver	Frontline Systems Inc.	Risk analysis and risk budgeting, Monte Carlo simulation	Academic, Oil & Gas, Pharmaceuticals, Insurance
Risk Solver Platform	Frontline Systems Inc.	Monte Carlo simulation, conventional optimization, simulation optimization, stochastic programming, robust optimization	Academic, General Business, Oil & Gas, Pharmaceuticals, Insurance, Manufacturing/Distribution
Risk Solver Premium	Frontline Systems Inc.	Risk analysis, Monte Carlo simulation, simulation optimization	Academic, Oil & Gas, Pharmaceuticals, Insurance, Manufacturing/Distribution
ServiceModel Optimization Suite	ProModel Corporation	Design, plan, evaluate and improve service industry systems such as Financial Services, Logistics, Business Re-Engineering	Financial Services, Logistics, Transportation, Food & Hotel Services, Entertainment, and Other Service Industries

ShowFlow	Webb Systems Limited	manufacturing; logistics; financial services	manufacturing - bottleneck analysis; layout revisions; new equipment evaluation
Simcad Pro-Patented Dynamic Process Simulator	CreateASoft, Inc	Continual Process Improvement, Facility Layout/Design, RFID/RTLS, Process Optimization, Lean, CapEx Justification	Manufacturing Solutions, Healthcare Solutions, Supply Chain Logistics Solutions, Service/Office Simulation Tools
Simio	Simio LLC	manufacturing, healthcare, airports, military, ports, supply chain, warehouse, lean/six sigma	manufacturing, healthcare, airports, military, ports, supply chain, warehouse, lean/six sigma
SIMUL8 Professional	SIMUL8 Corporation	Optimize throughput, maximize resource utilization, identify bottlenecks, reduced risk decisions, business process management	Business processes: call center, manufacturing, supply chain, logistics, healthcare, financial, education
SIMUL8 Standard	SIMUL8 Corporation	Optimize throughput, maximize resource utilization, identify bottlenecks, reduced risk decisions, business process management	Business processes: call center, manufacturing, supply chain, logistics, healthcare, financial, education
SIMUL8 Web	SIMUL8 Corporation	Simulation on the web. Share the benefits and power of simulation with others, no install, no learning curve.	Business processes: call center, manufacturing, supply chain, logistics, healthcare, financial, education
SLIM	MJC2	Strategic modelling, simulation and optimisation.	All large logistics and transport operations.
SLX	Wolverine Software Corporation	High-end, unique applications that require features not built into off-the-shelf simulation software	Air Traffic control, homeland security, telecommunications, logistics
Stat::Fit	Geer Mountain Software Corporation	Statistically fits input data to analytical distributions and exports into specific forms to simulation software.	Simulation and modeling, risk assessment, reliability, quality, engineering and financial management.
Tecnomatix Plant Simulation	Siemens PLM Software	Tecnomatix Plant Simulation software enables the simulation and optimization of production systems and processes.	Airports Aerospace Automotive and Supplier Consulting houses CPG Defense Electronic Logistic MRO Ports Shipyard Whitegood
Vanguard Business Analytics Suite	Vanguard Software	Collaborative modeling, strategic planning, risk analysis, forecasting, portfolio management, optimization, Web-based what-if	Consulting (internal/external), pharmaceuticals, oil & gas, aerospace, manufacturing, marketing, finance, academic
Vanguard Strategic Forecasting Suite	Vanguard Software	Collaborative modeling, strategic planning, risk analysis, forecasting, portfolio management, optimization, Web-based what-if	Consulting (internal/external), pharmaceuticals, oil & gas, aerospace, manufacturing, marketing, finance, academic
Vanguard System	Vanguard Software	Collaborative modeling, strategic planning, risk analysis, forecasting, portfolio management, optimization, Web-based what-if	Consulting (internal/external), pharmaceuticals, oil & gas, aerospace, manufacturing, marketing, finance, academic
WebGPSS	Beliber AB	General purpose discrete events simulation of situations with uncertainty, requiring many runs	Education, esp. students of business, logistics, supply chain systems
XLSim 3.0	AnalyCorp	Small traditional Monte Carlo applications, and interactive simulations based on DISTs.	General market for small Monte Carlo simulation models.

## B. SiRO: Exemplary implementation of the dining philosophers problem in C# from Profactor

To illustrate in more detail the modelling process using SiRO, exemplary implementation of the dining philosophers problem in C# are shown. The uses of event handlers as well as the chopstick class are presented:

```
LeftChopStickPickedUpEvent = new
    UnaryEvent<Philosopher>(this.Name +
        ".LeftChopStickPickedUp");
LeftChopStickPickedUpEvent.AddHandler(
    InternalLeftChopStickPickedUpHandler);
}
```

Figure A.1: A handler is added to an event.

```
private void InternalLeftChopStickPickedUpHandler(
    Philosopher sender) {
    Model.Seize(typeof(Chopstick), this,
        RightChopstickReceived, CheckRightChopstick);
}
```

Figure A.2: The event handler method.

```
using Profactor.SimEngine.Templates;
using System;

namespace DiningPhilosophers.Model
{
    public class Chopstick : ResourceObject
    {
        private int index;

        public int Index {
            get { return index; }
        }

        public Chopstick(int index) : base("Chopstick " +
            index.ToString())
        {
            this.index = index;
        }
    }
}
```

Figure A.3: The Chopstick class.

### C. Queueing theory: Model notation

Queueing theory studies mathematically the waiting (queueing). Therefore queueing models are used to represent the queueing systems.

The queueing models are conventionally labelled as follows [33]:

$$A/B/C$$

being:

$A$  .... Distribution of interarrival times

$B$  .... Distribution of service times

$C$  .... Number of servers

where  $A$  and  $B$  can be the following:

$M$  : exponential distribution (Markovian / stochastic times)

$D$  : degenerate distribution (constant / deterministic times)

$E_k$  : Erlang distribution (shape parameter  $k$  )

$G$  : general distribution (any arbitrary distribution allowed)

and  $C$  :

$s$  : the number of servers;  $s = 1$  .... single-server queueing system

$s > 1$  ....multi-server queueing system

## D. Basic statistics for simulation

The stochastic influences in a system are represented by statistical distributions. Thus, for instance the arrival of people at a bank or a supermarket can be adequately represented by an exponential distribution (compare chapter 3: A simple simulation experiment). The error-prone number of defective goods can be well represented by the binominal distribution. If little information about the stochastic influence is available the triangular distribution may be adequate.

### Exponential distribution

The exponential distribution describes the probability distribution to the next Poisson event (e.g., e.g., people arriving at a bank or a supermarket).

Assuming that in average every 2 minutes a client arrives. This arrival of the clients is i.i.d and can be described using an exponential distribution with parameters  $\lambda$ .

$$\lambda = \frac{1}{2} = 0,5$$

$$E(X) = \frac{1}{\lambda} \quad \Rightarrow \quad E(X) = \frac{1}{0,5}$$

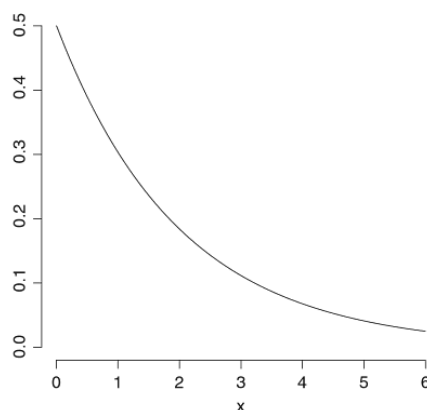


Figure A.4: Exponential distribution  $\lambda = 0,5$

### Triangular distribution

Figure A.5 shows the triangular distribution, which provides 3 parameters:

$a$  : the minimum

$b$  : the maximum

$c$  : the most likely value (peak)

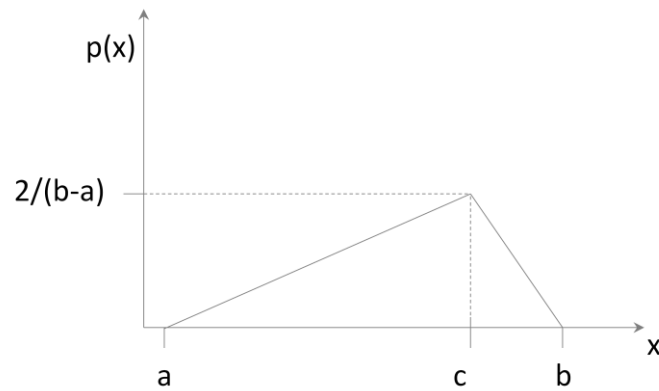


Figure A.5: Triangular distribution.

### Binominal distribution

The Binominal distribution is defined by its probability function:

$$p(x) = \binom{n}{x} p^x (1-p)^{n-x}$$

Figure A.6 illustrates the Binominal distribution for:

$$N = 10$$

$$p = 0.2$$

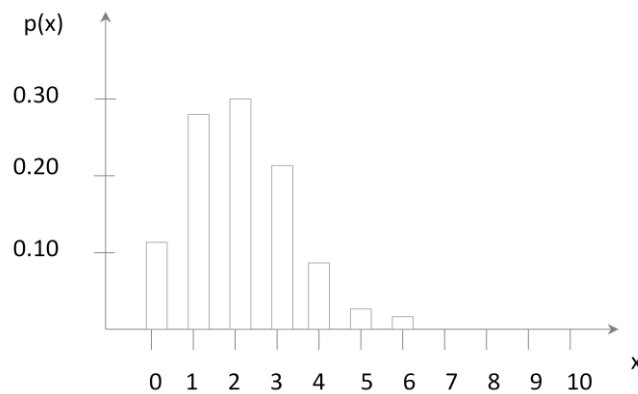


Figure A.6: Binomial distribution.

## E. Parameters of the OSGA

Also the parameters of the OSGA can be changed. Table A.1 shows possible parameter settings of the OSGA using HeuristicLab in version 3.3 .These parameter values have been proven as advantageous for the performed simulation study in chapter 7.

Table A.1: Parameters of the OSGA<sup>49</sup>.

Parameter	Value
ComparisonFactorLowerBound	0
ComparisonFactorModifier	LinearDiscreteDouble ValueModifier
ComparisonFactorUpperBound	1
Crossover	SubtreeCrossover
Elites	0
MaximumSelectionPressure	20
MutationProbability	25%
OffspringSelectionBeforeMutation	False
PopulationSize	20
Selector	RandomSelector
SucessRatio	0.95
MaxTreeHeight	6
MaxTreeSize	40

---

<sup>49</sup> Values obtained from the Procomposite project team.



## F. Submitted ARGESIM comparisons to the SNE Journal: C6 Emergency Department - Follow-up Treatment

### A GPSS Solution to ARGESIM Comparison C6 Emergency Department –Follow-up Treatment using S.N.A.K.E

Manfred Neubacher, Vienna Univ. of Technology, Austria; [manfred\\_neubacher@gmx.at](mailto:manfred_neubacher@gmx.at)  
Thomas Löscher, Vienna Univ. of Technology, Austria; [tloescher@osiris.tuwien.ac.at](mailto:tloescher@osiris.tuwien.ac.at)

**Simulator:** The simulator S.N.A.K.E (Simulation New Advanced Kit Enterprise edition) was developed by Sergio Sirotnsky at the University of Buenos Aires / School of Engineering. This simulator has been used for years by Professor Horacio Rocca to teach his students simulation.

S.N.A.K.E is a free software, which implements the simulation language GPSS (General Purpose Simulation System). With GPSS it is possible to construct mathematical, dynamic, discrete, stochastic and numeric systems. In GPSS active objects, called transactions, go through the system and compete for the use of passive resources. S.N.A.K.E offers a variety of tools to develop, evaluate and simulate a system. Furthermore it provides statistical analysis of the results of the performed system simulation. As input so called blocks are used to model a system. These blocks are text-based, and consist of a label, a verb and operator(s). The structure of GPSS blocks is as follow:

LABEL BLOCKNAME OPERATOR(S)

Using S.N.A.K.E a graphical simulation of the simulation run is not available.

**Model:** To simulate the emergency Department system, the created model consists of a set of about 90 to 100 GPSS blocks.

The resources of the hospital are defined by *Facilities* (for registration and the plaster room) and *Storages* (for the waiting area and the x-ray room).

Distributions and other functions are implemented by *Function* blocks. Two functions are shown below, the triangular distribution for the registration time (continuous) and the type of patient distribution (discrete). This is done randomly with pseudo random numbers (RN), which are provided by S.N.A.K.E.

```
TREG FUNCTION RN11,C3
0,12/.5,30/1,60
TYPE FUNCTION RN2,D4
.35,1/.55,2/.6,3/1,4
```

On creating the patients using the *Generate* block, local variables containing the patient type and the case of injury, are randomly assigned. Case 1 and

2 mean that the patients are assigned to be attended respectively in casualty ward 1 or 2.

```
GENERATE FN$ARRIVAL,,250
ASSIGN TYPE, FN$TYPE
ASSIGN CASE, FN$CASE
```

After that, the patients make their way through the system according their types and cases. When arriving to the x-ray or plaster room other local variables are altered, with the aim to know that they have already passed through this treatment points. So the transactions representing patients know which treatment point they have to process next, until they leave the emergency department.

30 minutes after the emergency department opens four doctor transactions are *Generated*. *User Chains* are used to Model the behaviour of the four doctors working in two casualty wards. First the patients are *Linked* in a *User Chain* and wait to be attended respectively by a doctor in CW1 or CW2. Then the doctors *Unlink* and examine them.

```
LINK P$CASE, FIFO
ODOC2 UNLINK 1,DOC2,1,,NOCW1
```

GPSS provides SNA's (System Numerical Attributes). One of these SNA's is M1, which is a timestamp on creation of each transaction. When leaving the emergency department the treatment time of each transaction is tabulated in a *Table*, using the total time in the system, M1.

```
TABULATE P$TYPE
1 TABLE M1,4800,1200,10
```

Figure 1 illustrates a part of a Simulation report, which is automatic generated by S.N.A.K.E.

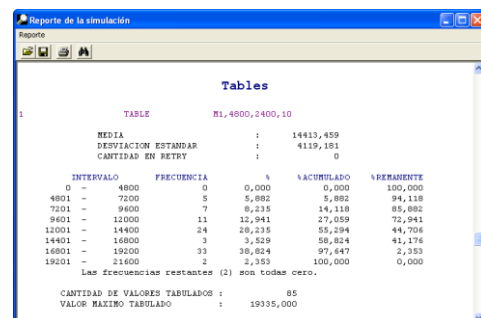


Figure 1: Statistics of patients type 1 (Screenshot)

In task b, a further transaction observes the length of the queue of waiting people to be attended in CW2 and changes the value of the *Savevalues*. Thus the two doctors know that they have to change casualty wards. *Savevalues* in GPSS are similar to the concept of global variables in modern objective orientated programming languages.

In task c, priorities of patients are altered, for instance before entering again into a treatment point. Therefore the linking strategy has also to be changed from FIFO to PR (priority).

LINK

P\$CASE, PR

**Task a: Simulation – Average Treatment Times.** The mean treatment time is between 130,5 and 248,7 minutes. After 387,9 minutes at 1.59 p.m. the last patient leaves and the emergency department closes.

Task a	#	mean [min]	stdev [min]
Patient T. 1	85	240,2	68,7
Patient T. 2	52	134,2	61,4
Patient T. 3	13	248,7	60,2
Patient T. 4	100	130,5	62,4
Overall time	250	174,7	83,4
Close hour	at 1.59 p.m.		

Table 1: Result of task a

**Task b: Doctors Exchange Strategy.** Using this strategy the overall treatment time increases as well as the end up time. At 8.15 a.m., 45 minutes after the emergency department opens, the queue of patients waiting to be attended in CW2, goes up to 20. At 11.38 a.m. it goes down to 5 and do not exceed 20 again until the emergency department closes. So there is just one change of doctors from CW1 to CW2 at 8.15 a.m. and one change back at 11.38 p.m. The emergency department closes at 2.08 p.m. Using this strategy of doctors exchange, the result is as follow:

Task b	#	mean [min]	stdev [min]
Patient T. 1	85	242,2	80,6
Patient T. 2	52	134,8	65,7
Patient T. 3	13	250,4	77,0
Patient T. 4	100	132,0	66,3
Overall time	250	176,4	89,7
Close hour	at 2.08 p.m.		

Table 2: Result of task b

**Task c: Priority Ranking.** In this task different strategies to minimize the standard deviation of the overall treatment time should be tested. A given strategy, which gives patients type 1 and 3 a higher priority when they enter again into a treatment point is suggested. This leads to a decrease of the mean treatment time for patients type 1 and 3. On the other hand the mean treatment time for patients type 2 and 4 is

increased. The mean treatment time of all four patient types is now between 160,3 and 166,3 minutes. The aim of this task, to minimize the standard deviation of overall treatment time, is well achieved. The standard deviation of all patients types decreases from 83,4 minutes in task a to 77,5 minutes.

Task c	#	mean [min]	stdev [min]
Patient T. 1	85	163,6	76,5
Patient T. 2	52	160,3	80,0
Patient T. 3	13	166,3	70,3
Patient T. 4	100	162,2	79,1
Overall time	250	162,5	77,5
Close hour	at 2.02 p.m.		

Table 3: Result of task c

No better priority ranking was found than this suggested solution. However, other strategies were implemented, trying to find potential better solutions than the above mentioned priority ranking. Table 4 shows the mean treatment time and standard deviation according the patients assigned casualty ward from task c.

Task c	#	mean [min]	stdev [min]
Case 1	150,6	169,0	79,4
Case 2	99,4	145,7	62,8

Table 4: Additional analysis of the result of task c (average value of simulation runs)

It can be observed, that the patients who are assigned to be attended in CW1, stay much longer in the emergency department. They have also a higher standard deviation than patients of case 2. Using this data several strategies were studied. Some of them are:

- Patients case 1 get higher priority when entering the system.
- When there are more patients of a case 1 or 2 in the system than expected (60% case 1 and 40% case 2), this case gets higher priority.

These strategies minimize the standard deviation just in some particular cases. For instance when there are more than 63 % of patients assigned to CW1.

S.N.A.K.E. Version 1.2.0 was used to simulate the emergency department. The simulation was performed on a Pentium 4, 2,6GHz Laptop and took about 6 to 7 seconds.

#### Corresponding author:

Manfred Neubacher, Vienna University of Technology,  
Department of Analysis and Scientific Computing  
Wiedner Hauptstrasse 8-10, 1040 Vienna, Austria  
[manfred\\_neubacher@gmx.at](mailto:manfred_neubacher@gmx.at)

## C6 Emergency Department Follow-up Treatment solved using SiRO – a highly sophisticated object-oriented simulation library

Thomas Löscher, Matthias Gruber, Profactor GmbH, Austria; [thomas.loescher@profactor.at](mailto:thomas.loescher@profactor.at)  
Manfred Neubacher, Gerald Auer, Vienna Univ. of Technology, Austria; [manfred\\_neubacher@gmx.at](mailto:manfred_neubacher@gmx.at)

**Simulator:** The SiRO (Simulation, Rating, Optimization) simulation library offers both features of a simulator and an optimizer. Thus, SiRO allows the use of simulation based optimization.

With the use of modern object-orientated programming languages, the SiRO developers try to bring a new approach into the area of discrete simulation. This highly sophisticated simulation library was developed by Thomas Löscher and Matthias Gruber from the Profactor company. The simulator framework is available as Java and C# edition as well. So the models are written entirely in those languages with the advantage that the modellers are not restrained in a predefined structure.

SiRO discriminates between events and event handlers. An event is defined to be named, abstract entity, instances of which can occur repeatedly. An event symbolizes a physical interaction, predetermined by a (possibly stochastic) process. An event handler is basically just a method, executing some logic as a consequence to the event occurrence and dependent on the event data.

All random numbers used by a model are initialized automatically through a central seed. An index can be assigned to this central seed. This brings the big advantage that all model objects have the same random number sequence, in despite of the model scheduling changes. Consequently different simulation runs can be easily compared. Another advantage by using an initialized central seed is that the simulation experiments are repeatable.

Detailed information, further documentation, tutorials and demo versions in C# and Java including n-doc and java-doc are available at the Profactor web page: [www.profactor.at/en/production/produkte/siro.htm](http://www.profactor.at/en/production/produkte/siro.htm)

**Model:** The model of an emergency department was built using the Java environment of SiRO in Version 2.1. The simulation model makes strong use of generics and it could be extended very easily.

The simulated emergency department comprises the following facilities:

- Registration (one person)
- Waiting area
- Casualty wards (CW1 and CW2) each stuffed with two doctors
- X-ray room containing two x-ray units
- Plaster room (one person)

These facilities (the rooms and the people working inside) are represented as simulation objects in the model. Additionally there is a source and a sink object in the model. The waiting area is represented by two queues.

By calling the `addObject()` method all model components and other relevant data are added to the model. Furthermore, the model reference (`getModel()`) can be used to add simulation events, simulation objects, the simulation start time, and so on.

Patients go through the emergency department according to their types and casualties. For that each patient instance gets as a property a map with the corresponding way from the registration until they leave. E.g. for patients type 1 and assigned casualty 2 the corresponding destination map is: "registration", "cw2", "xray", "cw2", "exit".

The architecture of the simulation model consists of three packages:

- A program class where the 3 different tasks are built and executed
- The necessary interfaces
- The model itself containing nine classes representing the emergency department system

The heart of the simulation is the occurrence of events. An instance of an event is scheduled at a (probably stochastic) point in time. When this point (which must be in the future) becomes current all Handlers registered at this Event are executed.

The model contains three different events, which are:

*patientAppearsEvent:*

Each of the 250 patients arrives at the registration of the emergency department. The arriving rate is exponentially distributed with a parameter of 0.3. SiRO provides a variety of mathematical distributions. One of these is the exponential distribution, which is used here.

#### ***startWorkingEvent:***

At 8.00 a.m. an event is scheduled, that the four doctors start working in the two casualty wards.

#### ***treatmentFinishedEvent***

The stochastic finishing time of a treatment in one of the treatment points is scheduled. According the parameters of each treatment point the time of these events is triangular distributed. The corresponding triangular distribution returns the value when the treatment will be finished and the treatmentFinishedEvent is scheduled at this specific point of time. When this scheduled point in time becomes current the logic of the Event Handler is executed and the patient leaves the treatment point. After that the patient goes on processing its destination map.

The patients are queuing to enter the treatment points. In task a and c the queuing strategy is FIFO, while this strategy is changed to a priority ranking in task b. For that a LinkedList is sorted which represents the queuing people.

When the patients have passed all their assigned treatment points they have to process “exit” as the last item in their destination map. Exit is a Java class, which is used for the collection of relevant statistic data. So, for each patient (type 1 to 4) the number of occurrence and the total time in system are stored. As well the standard deviation of all patients is calculated.

After the last of the 250 patients was simulated the statistics is printed out on console. SiRO also allows logging data (e.g. event information), which may be of interest for the modellers as well.

**Task a: Simulation – Average Treatment Times.** The simulated treatment time takes between 124 and 242 minutes depending on the patient type. In this task the standard deviation of all patients is 75 minutes. The emergency department closes at 1.21 p.m.

**Task b: Doctors’ Exchange Strategy.** The doctors’ exchange was intended to be modelled like in real life. As soon as the queue of patients waiting to be attended in CW2 exceeds 20, the first doctor of both casualty wards who has

already finished the current examination changes to the other casualty ward.

To enter into the other CW this doctor has to wait until another doctor from the other CW also finishes its examination.

Using this strategy of doctors’ exchange does not bring any advantage at all. It leads to an increased treatment time for all four patient types as well the standard deviation goes up to 100 minutes. In this task the emergency department closes at 3.32 p.m.

**Task c: Priority Ranking.** A LinkedList is used to represent the patients queuing before each treatment point. These LinkedLists are sorted by the time in system of each patient.

Using this priority ranking strategy the treatment time for patients, who enter more than once into a treatment point (type 1 and 3) decreases significantly. Also the standard deviation of the overall treatment time decreases down to 69 minutes. In this task the emergency department closes at 1.21 p.m.

Mean time	#	Task a	Task b	Task c
Patient 1	71	218	259	143
Patient 2	48	137	146	165
Patient 3	11	242	278	144
Patient 4	120	124	129	150
Std.dev.		75	100	69
Ov.treat.time	250	351	482	351
Close hour (p.m.)		1.21	3.32	1.21

Table 1: Simulation results in minutes

#### **Resumé:**

SiRO provides all features of a modern discrete event simulator with the flexibility of high level programming languages. By discriminating events and their handler SiRO allows a very natural design of real world problems.

A further advantage using SiRO is the excellent performance. The duration of the performed simulation for the run of three different tasks took between 90ms and 310ms. It was done on a Pentium 4, 2.6GHz Laptop.

#### **Classification: Event-oriented, object-oriented**

#### **Corresponding Authors:**

Thomas Löscher, Profactor GmbH,  
[thomas.loescher@profactor.at](mailto:thomas.loescher@profactor.at)  
 Manfred Neubacher, Vienna University of Technology,  
[manfred\\_neubacher@gmx.at](mailto:manfred_neubacher@gmx.at)

## References

- [1] Averill M. Law and W. D. Kelton, *Simulation Modeling and Analysis* vol. Third Edition: McGraw-Hill, 2000.
- [2] J. Banks, *et al.*, *Discrete Event System Simulation*: Pearson Education, 2010.
- [3] W. Kühn, *Digitale Fabrik - Fabriksimulation für Produktionsplaner*. München - Wien: Hanser, 2006.
- [4] H. Jodlbauer, *Produktionsoptimierung - Wertschaffende sowie kundenorientierte Planung und Steuerung*, Second Edition ed.: Springer Wien NewYork, 2008.
- [5] G. Chryssolouris, *Manufacturing Systems: Theory and Practice*, Second Edition ed.: Springer Science+Business Media, Inc., 2006.
- [6] G. Schuh, *Produktionsplanung und -steuerung: Grundlagen, Gestaltung und Konzepte*: Springer Berlin Heidelberg, 2006.
- [7] S. Wenzel, "Modellbildung und Simulation in Produktion und Logistik - Stand und Perspektiven," *ASIM Review*, 2009.
- [8] T. Löscher and M. Gruber, "SiRO Simulation Engine," *White paper, Profactor GmbH*, 2009.
- [9] L. März, "Simulation und Optimierung in Produktion und Logistik," *Springer Berlin Heidelberg*, 2011.
- [10] G. Deng, Ferris, M.C., "Variable-number sample-path optimization," *Mathematical Programming: Series B*, vol. 117, pp. 81-109, 2009.
- [11] N. Holthöfer, "3. Grundlagen klassischer PPS-Ansätze (MRP-II)," *Fraunhofer ALB*, 2000.
- [12] E. Westkämper, *Einführung in die Organisation der Produktion*. Berlin Heidelberg: Springer-Verlag, 2006.
- [13] M. C. Fu, "Optimization for simulation: Theory vs. Practice," *INFORMS JOURNAL ON COMPUTING*, vol. 14, pp. 192-215, January 1, 2002.
- [14] M. C. Fu, Glover, F. W., April, J., "Simulation Optimization: A review, new developments, and applications," *Proceedings of the 2005 Winter Simulation Conference*, 2005.
- [15] J. Banks, *Handbook of Simulation: Principles, Methodolgy, Advances, Applications, and Practice*: John Wiley & Sons, 1998.
- [16] S. Bangsow, "Fertigungssimulationen mit Plant Simulation und SimTalk," *Hanser Verlag München Wien*, 2008.
- [17] M. P. Groover, *Fundamentals of Modern Manufacturing: Materials, Processes, and Systems*, Third Edition ed.: John Wiley & Sons, 2002.
- [18] Wallace J. Hopp and M. L. Spearman, *Factory physics: foundations of manufacturing management*, Second Edition ed. New York: McGraw-Hill Higher Education, 2000.
- [19] H. Lödding, *Verfahren der Fertigungssteuerung*, Second edition ed.: Springer-Verlag Berlin Heidelberg, 2008.
- [20] W. Dangelmaier, *Theorie der Produktionsplanung und-steuerung: Im Sommer keine Kirschpralinen?: Springer*, 2009.
- [21] Nedeß Ch., *et al.*, "Simulation-supported Optimisation for the Improvement of Processes and Structures in Production Facilities," *Advances in Simulation for Production and Logistics Application - Fraunhofer IRB*, 2008.
- [22] Albert D. Baker, "A Survey of Factory Control Algorithms That Can Be Implemented in a Multi-Agent Heterarchy: Dispatching, Scheduling, and Pull," *Journal of Manufacturing Systems*, vol. Vol. 17/Nr. 4, 1998.
- [23] K. N. McKay and V. C. S. Wiers, *Practical Production Control - A Survival Guide for Planners and Schedulers*: J. Ross Publishing, 2004.
- [24] K. R. M. John M. Gross, *Kanban Made Simple - Demystifying and Applying Toyota's Legendary Manufactruing Process*: AMACOM - American Management Association, 2003.

- [25] E. M. Goldratt and J. Cox, *The goal: excellence in manufacturing*: North River Press, 1984.
- [26] W. A. Levinson, *Beyond the Theory of Constraints - How to Eliminate Variation and Maximize Capacity*. New York: Productivity Press, 2007.
- [27] B. Sproull, *The Ultimate Improvement Cycle: Maximizing Profits through the Integration of Lean, Six Sigma, and the Theory of Constraints*. Boca Raton - London - New York: CRC Press Taylor & Francis Group, 2009.
- [28] Levinson Productivity Systems, P.C.; *Matchsticks-and-Dice Simulaiton from The Goal*; [http://www.ct-yankee.com/lean/toc\\_dice.html](http://www.ct-yankee.com/lean/toc_dice.html) last visit 22.6.2011.
- [29] W. Marquardt, et al., "Overview on OntoCAPE," *OntoCAPE*, 2010.
- [30] H. Stachowiak, *Allgemeine Modelltheorie*: Springer-Verlag, 1973.
- [31] "The Association of German Engineers," *VDI Guidelines*, <http://www.vdi.eu/>.
- [32] R. E. Shannon, *Systems Simulation the art and science*: Prentice-Hall, 1975.
- [33] Frederick S. Hillier and G. J. Lieberman, *Introduction to Operations Research* vol. Seventh Edition, 2001.
- [34] J. Banks, "Discrete Event Simulation," *Proceedings of the 1999 Winter Simulation Conference*, pp. 7-13, 1999.
- [35] M. Pidd and A. Carvalho, "Simulation software: not the same yesterday, today or forever," *J of Sim*, vol. 1, pp. 7-20, 2006.
- [36] D. N. Goldsman, Richard, E.; Wilson, James, R., "A Brief History of Simulation," *Proceedings of the 2009 Winter Simulation Conference*, 2009.
- [37] J. J. Swain, "The 2009 Simulation Software Survey," *OR/MS Today*, vol. 36, October 2009.
- [38] EUROSIS, "[www.eurosis.org/cms/?q=node/61](http://www.eurosis.org/cms/?q=node/61)," *Simulation Software*, , last visit: 7. March 2011.
- [39] P. L'Ecuyer, "Chapter 3 Uniform Random Number Generation," *Handbooks in Operations Research and Management Science*, vol. Volume 13, pp. 55-81, 2006.
- [40] W. D. Kelton, "Some modest proposals for simulation software: Design and analysis of experiments," *Proceedings of the 34th Annual Simulation Symposium*, pp. 237-242, 2001.
- [41] W. D. Kelton, "Chapter 7 Implementing Representations of Uncertainty," *Handbooks in Operations Research and Management Science*, vol. Volume 13, pp. 181-191, 2006.
- [42] G. Gordon, "The Development of the General Purpose Simulation Systems (GPSS)," *ACM SIGPLAN Notices*, Vol. 13, No. 8,, 1978.
- [43] I. Ståhl, "GPSS - 40 Years of Development," *Proceedings of the 2001 Winter Simulation Conference*, 2001.
- [44] ARGESIM, "List of ARGESIM Benchmarks," [www.argesim.org/index.php?id=68](http://www.argesim.org/index.php?id=68) last visit: 23.02.2011.
- [45] C. D. Pegden, "How Simio Objects Differ from Other Object-Oriented Modeling Tools," *Simio White Paper* <http://www.simio.com/resources/white-papers/How-Simio-Objects-Differ-From-Others/How-Simio-v2-Objects-Differ-from-Other-Object-Oriented-Modeling-Tools-WSC-2009.pdf>, last visit 15.6.2011, 2009.
- [46] B. Eidenmüller, *Die Produktion als Wettbewerbsfaktor* vol. Second Edition: Verl. Industrielle Organisation, 1991.
- [47] "Winter Simulation Conference [www.wintersim.org](http://www.wintersim.org)," last visit: 06.06.2011.
- [48] Gartner, "Identifies the Top 10 Strategic Technologies for 2010," <http://www.gartner.com/it/page.jsp?id=1210613> , last visit 10.06.2011."
- [49] J. L. Hong, "A Brief Introduction to Optimization via Simulation," *Proceedings of the 2009 Winter Simulation Conference*, 2009.
- [50] S. Ólafsson, Kim, J., "Simulation optimization," *In: Proceedings of the 2002 Winter Simulation Conference*, vol. IEEE Press, Piscataway, NJ, pp. 79-84, 2002.
- [51] S. Henderson, et al., "Simulation," *Handbooks in Operations Research and Management Science*, vol. Volume 13, Sep 2006.

- [52] A. Gosavi, *Simulation-Based Optimization: Parametric Optimization Techniques and Reinforcement Learning*: Kluwer, Norwell, 2003.
- [53] B. B. Keskin, et al., "A simulation-optimization approach for integrated sourcing and inventory decisions," *Computers & Operations Research*, vol. 37, pp. 1648-1661, 2010.
- [54] S. Andradóttir, "Chapter 20 An Overview of Simulation Optimization via Random Search," *Handbooks in Operations Research and Management Science*, vol. Volume 13, pp. 617-631, 2006.
- [55] Myers R.H., et al., *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, Third Edition ed.: John Wiley & Sons, 2009.
- [56] R. R. Barton and M. Meckesheimer, "Chapter 18 Metamodel-Based Simulation Optimization," *Handbooks in Operations Research and Management Science*, vol. Volume 13, pp. 535-574, 2006.
- [57] H. Zhang, "Simulation-based optimization of dispatching rules for semiconductor wafer fabrication system scheduling by the response surface methodology," *Int. J. Adv. Manuf. Technol.*, vol. 41:110-121, 2009.
- [58] S. Andradóttir, "A review of simulation optimization techniques," *Proceedings of the 1998 Winter Simulation Conference*, 1998.
- [59] M. C. Fu, "Chapter 19 Gradient Estimation," *Handbooks in Operations Research and Management Science*, vol. Volume 13, pp. 575-616, 2006.
- [60] S.-H. Kim and B. L. Nelson, "Chapter 17 Selecting the Best System," *Handbooks in Operations Research and Management Science*, vol. Volume 13, pp. 501-534, 2006.
- [61] Rubinstein R. and A. Shapiro, *Discrete Event Systems: Sensitivity Analysis and Stochastic Optimization by the Score Function Method*. New York: John Wiley & Sons, 1993.
- [62] S. Robinson, "Analysis of sample-path optimization," *Mathematics of Operations Research*, vol. 21, pp. 513-528, 1996.
- [63] Z. Michalewicz and D. Fogel, *How to Solve It: Modern Heuristics*, Second Edition ed.: Springer, 2004.
- [64] S. Ólafsson, "Chapter 21 Metaheuristics," *Handbooks in Operations Research and Management Science*, vol. Volume 13, pp. 633-654, 2006.
- [65] W. K. B. Box G. E. P., "On the Experimental Attainment of Optimum Conditions," *Journal of the Royal Statistical Society. Series B (Methodological)*, Vol. 13, No. 1 pp 1-45, 1951.
- [66] S. Voss, Martello, S., Osman, I., Roucairol, C., *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*. Boston: Kluwer Academic Publishers, 1999.
- [67] F. Glover, "Tabu Search - Part I," *ORSA Journal on Computing*, vol. 1, pp. 190-206, 1989.
- [68] F. Glover, "Tabu Search - Part II," *ORSA Journal on Computing*, vol. 2, pp. 4-32, 1990.
- [69] F. Glover and M. Laguna, *Tabu Search*: Kluwer Academic Publishers, 1997.
- [70] S. Kirkpatrick, Gelatt, C., Vecchi M., "Optimization by Simulated Annealing," *Science*, vol. 220, pp. 671-680, 1983.
- [71] J. H. Holland, "Adaption in Natural and Artificial Systems," *University of Michigan Press*, 1975.
- [72] M. Affenzeller, Winkler, S., Wagner S., Beham, A., *Genetic Algorithms and Genetic Programming - Modern Concepts and Practical Applications*: CRC Press, 2009.
- [73] M. Affenzeller, Wagner, S., "Offspring Selection: A New Self-Adaptive Selection Scheme for Genetic Algorithms," *Adaptive and Natural Computing Algorithms, Coimbra, Portugal*, pp. pp. 218-221, 2005.
- [74] S. Wagner, "Heuristic Optimization Software Systems - Modeling of Heuristic Optimization Algorithms in the HeuristicLab Software Environment, PhD Thesis, Institute for Formal Models and Verification, Johannes Kepler University Linz, Austria," 2009.
- [75] S. Terzi, Cavalieri, S., "Simulation in the supply chain context: a survey," *Computers in Industry*, vol. 53, pp. 3-16, 2004.

- [76] P. Kochel, Nielander U., "Simulation-based optimization of multi-echelon inventory systems," *Int. J. of Production Economics*, vol. 93-94, pp. 505-13, 2005.
- [77] "Factory of Tomorrow, Publications: <http://www.fabrikderzukunft.at/publikationen/index.en.htm>, last visit 16.6.2011."
- [78] ASMA GmbH. <http://www.asma.at/index-e.html> last visit 17.08.2011.
- [79] R. Haupt, "A Survey of Priority Rule-Based Scheduling," *OR Spektrum*, vol. 11, pp. 3-16, 1989.
- [80] W. Hacker and P. Richter, *Psychische Fehlbeanspruchung. Psychische Ermüdung, Monotonie, Sättigung und Stress*. Berlin: Verlag der Wissenschaften, 1980.
- [81] P. Brucker, Drexl, A., Möhring, R., Neumann, K., Pesch, E., "Resource-constrained project scheduling: Notation, classification, models, and methods," *European Journal of Operational Research*, vol. 112, pp. 3-41, 1999.
- [82] M. Kapanoglu and M. Alikalfa, "Learning IF-THEN priority rules for dynamic job shop using genetic algorithms," *Robotic and Computer-Integrated Manufactuuring*, vol. 27, pp. 47-55, 2011.
- [83] S. Graebe and M. Schleicher, "Holistische Produktion," *e & i Elektrotechnik und Informationstechnik*, vol. 119, pp. 274-281, 2002.
- [84] G. Zeichen and K. Fürst, *Automatisierte Industrieprozesse*. Vienna - New York: Springer, 2000.
- [85] Fraunhofer IML, "Situationsanalyse "Simulation in Produktion und Logistik in Deutschland"," *IML*, 1996.