

# Entwicklung eines validierten Prozessreferenzmodells zur effizienten Modellierung und Simulation klinischer Prozesse

Dissertation zur Erlangung des akademischen Grades  
Doktoringenieur (Dr.-Ing.)

Eingereicht an der  
Fakultät für Informatik und Automatisierung  
der Technischen Universität Ilmenau

Vorgelegt von Dipl.-Kfm. Frank Richter

1. Gutachter: Prof. Dr. Horst Salzwedel
2. Gutachter: Prof. Dr. Jan Torzewski
3. Gutachter: Prof. Dr. Wolfgang Fengler

Tag der Einreichung: 21.05.2008

Tag der wissenschaftlichen Aussprache: 19.01.2009

*Für meinen Vater.*

# Inhaltsverzeichnis

Zusammenfassung / Summary .....	V
Abkürzungsverzeichnis .....	IX
Abbildungsverzeichnis .....	XII
Tabellenverzeichnis .....	XXI
1. Einleitung .....	1
1.1 Problemstellung .....	1
1.2 Zielsetzung und Vorgehensweise der vorliegenden Arbeit.....	10
2. Systemsimulation.....	12
2.1 Kybernetische System- und Modelltheorie .....	12
2.1.1 Grundlagen der kybernetischen System- und Modelltheorie .....	12
2.1.2 Systemerfassung und -modellbildung.....	13
2.1.2.1 Systemzustände und -strukturen.....	13
2.1.2.2 Systemabbildung.....	17
2.1.3 Steuerung und Regelung.....	20
2.2 Simulation und deren Anwendungsgebiete .....	24
3. Prozessmodellierung .....	31
3.1 Hauptmerkmale des Modellbegriffs .....	31
3.2 Prozesse .....	34
3.2.1 Begriffsdefinitionen .....	34
3.2.2 Standardisierte Beschreibung von Prozessen .....	35
3.2.3 Methoden zur Beurteilung und Verbesserung der Prozessqualität .....	42
3.2.3.1 Capability Maturity Model Integration .....	43
3.2.3.2 EFQM.....	50
3.2.3.3 Six Sigma .....	56
3.3 Prozessmodelle.....	61
3.3.1 Petri-Netze.....	61
3.3.2 Ereignisgesteuerte Prozessketten .....	64
3.3.3 Unified Modeling Language .....	66

4.	Grundlegende Komponenten von Simulationssystemen - Ein Überblick .	77
4.1	Basiskomponenten zur Modellierung von Unternehmensprozessen .....	78
4.1.1	Automaten .....	78
4.1.2	Input-Output-Verknüpfung von Automaten .....	84
4.1.3	Buffer .....	86
4.1.4	Ressourcen .....	88
4.1.4.1	Personal .....	88
4.1.4.2	Maschinen, Werkzeuge und Hilfsmittel .....	92
4.1.4.3	Material .....	93
4.1.4	Transaktionen.....	94
4.2	Basiskomponenten zur Darstellung und Analyse entscheidungs- relevanter Informationen .....	98
4.2.1	Ressourcenbezogene Statistiken .....	100
4.2.1.1	Personal .....	100
4.2.1.2	Maschinen, Werkzeuge und Hilfsmittel .....	106
4.2.1.3	Material .....	107
4.2.1.4	Ressourcenübergreifende Statistiken.....	109
4.2.2	Automatenbezogene Statistiken .....	111
4.2.3	Bufferbezogene Statistiken.....	115
4.2.4	Transaktionsbezogene Statistiken.....	117
5.	Standardisierte Beschreibung klinischer Prozesse.....	119
5.1	Klinische Behandlungspfade .....	119
5.2	Das akute Koronarsyndrom.....	121
5.2.1	Diagnose und Risikostratifizierung .....	122
5.2.2	Therapie des akuten Koronarsyndroms.....	125
5.2.2.1	Therapie des akuten Koronarsyndroms ohne ST- Hebung.....	126
5.2.2.2	Therapie des akuten Koronarsyndroms mit ST- Hebung.....	128
5.3	Entwicklung der Prozesskomponenten für den klinischen Behandlungspfad des akuten Koronarsyndroms.....	129

6.	Entwicklung eines validierten Prozessreferenzmodells mittels standardisierter und wieder verwendbarer Modellkomponenten .....	137
6.1	Modellkomponenten zur Darstellung des Prozesses „Patientenaufnahme“ .....	138
6.2	Modellkomponenten zur Darstellung des Prozesses „Untersuchung“ .....	142
6.3	Modellkomponenten zur Darstellung des Prozesses „Therapie“ ...	152
6.4	Modellkomponenten zur Darstellung des Prozesses „Patientenentlassung“ .....	156
7	Entwicklung des Simulationsmodells und Durchführen der Simulation..	158
7.1	Ausgangssituation und Zielsetzung des Modells.....	158
7.2	Aufbau des Simulationsmodells .....	163
7.3	Modellbestimmung und Validierung des Modells .....	176
7.4	Lösungen zur Optimierung der Prozesse .....	178
7.4.1	Performance-Indikatoren zur Analyse von Schwachstellen.....	180
7.4.2	Eliminierung redundanter Arbeitsschritte .....	183
7.4.3	Überprüfen der Möglichkeit zur Wegstrecken-Optimierung .....	184
7.4.4	Reduktion von Wartezeiten.....	186
7.4.4.1	Analyse der Verfügbarkeit und Auslastung eingesetzter Ressourcen.....	187
7.4.4.2	Optimierung der Einbestellungsreihenfolge der Patienten mittels eines genetischen Algorithmus .....	189
7.4.4.3	Übertragung der bisherigen Ergebnisse in das Soll-Modell und Durchführung weiterer Prozessoptimierungen.....	200
8	Fazit und Ausblick.....	208
	Literaturverzeichnis .....	211
	Anhang.....	229

## Zusammenfassung

Um einem weiteren Anstieg der Gesundheitsausgaben entgegenzuwirken, wurden in den letzten Jahren von der Bundesregierung die gesetzlichen Rahmenbedingungen sukzessive verändert. Eine der wesentlichsten Neuerungen ist das pauschalierte Entgeltsystem für voll- und teilstationäre Krankenhausleistungen auf Basis der Diagnosis Related Groups (DRGs). Seit der Einführung der DRGs im Jahre 2004 werden die Leistungen von Krankenhäusern nicht mehr tageweise, d.h. in Form einer tagesgleichen Pflegesatzvergütung abgerechnet, sondern pauschal je nach Krankheitsbild. Die DRGs führen gegenüber der tagesgleichen Pflegesatzvergütung zu völlig anderen ökonomischen Anreizen für Krankenhäuser. Krankenhäuser müssen künftig versuchen, Ressourcen effizienter einzusetzen, um über die Verkürzung von Behandlungs- und Wartezeiten die Verweildauer der Patienten zu senken sowie die entstehenden Kosten je Behandlungsfall zu minimieren. Entscheidend für das Erreichen dieser Ziele ist letztendlich die Qualität der Prozesse, also die Effizienz und Effektivität der Arbeitsschritte, die während der Behandlung eines Patienten durchlaufen werden. Um über Prozessoptimierung sowohl die Verweildauern von Patienten zu reduzieren als auch Kosten zu senken, benötigen Kliniken geeignete Tools und Methoden. Das dynamische Verhalten von komplexen klinischen Prozessen kann im Rahmen einer Prozessoptimierung mit geeigneten leistungsfähigen Simulationssystemen abgebildet, analysiert und optimiert werden. Für den Einsatz im klinischen Umfeld sollten diese Simulationssysteme u.a. ein validiertes Prozessreferenzmodell zur Modellierung und Simulation klinischer Prozesse bereitstellen, welches aus wieder verwendbaren, ausführbaren Modellbausteinen besteht. Dies ist zum einen Grundlage für die Vergleichbarkeit von Ergebnissen, die z.B. im Rahmen von Benchmark-Studien erzielt werden. Zum anderen wird durch die Verwendung von standardisierten und wieder verwendbaren Modellkomponenten die Entwicklungszeit von Simulationsmodellen erheblich verkürzt.

In der vorliegenden Arbeit wurde mit dem Simulationssystem MLDesigner ein validiertes Prozessreferenzmodell, welches aus standardisierten und wieder

verwendbaren Modellkomponenten zur Modellierung und Simulation klinischer Prozesse besteht, entwickelt. Die Modellkomponenten sind modular aufgebaut, in definierten Bibliotheken des MLDesigners strukturiert hinterlegt und mit den entsprechenden klinischen Begriffen aus der Krankenhauspraxis bezeichnet. Als Basis hierfür dienen ausgewählte klinische Behandlungspfade.

Die entwickelten Modellkomponenten wurden anschließend für die Modellierung, Simulation und Optimierung der Prozesse einer onkologischen Tagesklinik erfolgreich eingesetzt. Im Rahmen der durchgeführten Prozessoptimierung konnte eine signifikante Reduktion der Patientenwartezeiten in der Tagesklinik erzielt werden.

Das in dieser Arbeit entwickelte Prozessreferenzmodell stellt den konkreten Ansatz für die Standardisierung der Modellierung und Simulation von Krankenhausprozessen dar. Aufgrund seines modularen Aufbaus kann das validierte Prozessreferenzmodell jederzeit ergänzt bzw. erweitert werden. Die bereits entwickelten Modellkomponenten können zur Durchführung weiterer Simulationsstudien, z.B. als Basis für den Systementwurf oder zur Optimierung der transsektoralen Integration<sup>1</sup>, eingesetzt werden.

---

<sup>1</sup> Zusammenarbeit von niedergelassenen Ärzten, Krankenhäusern und Reha-Einrichtungen.

## Summary

To stop further increase of health care cost, the German government successively changed the regulatory framework including the reimbursement policies of hospitals. One of the major innovations is the lump-sum compensation of in-patient treatments based on Diagnosis Related Groups (DRGs). Since the implementation of the DRG-system in 2004, clinics are no longer being paid per day of performed services for each treated patient, but based on the syndrome/diagnosis of the respective patient. Compared to the former hospital and nursing charges, that were identical each day of performed services, DRGs now lead to different economical incentives for hospitals. From now on, hospitals need to utilize their resources much more efficiently in order to reduce waiting and treatment time of patients as well as to minimize cost per case. Mandatory for the accomplishment of those objectives is the quality of processes, meaning the efficiency and effectiveness of each process step that needs to be performed during patient treatment. In order to reduce the length of stay of patients as well as to reduce the cost for treatment, hospitals need appropriate tools and methods. The dynamic behavior of complex clinical processes in process optimization studies can only be modeled, analyzed and optimized with adequate, well performing simulation systems. Among others, those simulation systems shall provide a validated process framework for modeling and simulating clinical processes, comprising re-usable and executable building blocks. This is basis for comparisons of accomplished results, for instance in benchmark studies. On the other hand, the length of time for the development of simulation models can be drastically reduced by using standardized and re-usable building blocks.

In this thesis a validated process framework was developed with the simulation system MLDesigner. This process framework comprises standardized building blocks for rapid modeling and simulation of hospital processes. Those building blocks are modularized, structured in pre-defined libraries of MLDesigner and named after the respective clinical process step. Selected clinical care paths were the basis for the development of the process framework.



The developed building blocks were utilized for modeling, simulating and optimizing the processes of a cancer treatment center. As part of the process optimization study a significant reduction of patients' waiting times was achieved.

The process framework represents the concrete approach for standardization of modeling and simulation of clinical processes. Due to its modular set-up the process framework can be added and extended by additional processes at any time. The existing building blocks can be utilized for further simulation studies, e.g. as the basis for system design or the optimization of the integration of different sectors of the healthcare environment, i.e. integration of processes between doctors in private practices, hospitals and rehabilitation centers.

## Abkürzungsverzeichnis

Abb. ....	Abbildung
ACS.....	Acute Coronary Syndrom
akt .....	aktuell
Angio .....	Angiographie
AMTOS .....	Automaton based Modelling and Task Operating System
ARIS.....	Architektur integrierter Informationssysteme
ASS.....	Acetylsalicylsäure
avg. ....	average
BpflV .....	Bundespflegegesetzverordnung
BIP .....	Bruttoinlandsprodukt
BWS.....	Brustwirbelsäule
bzw. ....	beziehungsweise
CMM® .....	Capability Maturity Model
CMMI® .....	Capability Maturity Model Integration
CT .....	Computertomographie
CTQ .....	Critical to Quality
DE .....	discrete event
DIN.....	Deutsche Industrienorm
d.h. ....	das heißt
DMP .....	Disease-Management-Programm
DRG .....	Diagnosis Related Group
dt. ....	deutsch
E.....	Einheiten
EFQM .....	European Foundation for Quality Management
EKG .....	Elektrokardiographie
Engl. ....	Englisch
EPK.....	Ereignisgesteuerte Prozesskette(n)
etc. ....	et cetera
evtl. ....	eventuell
f .....	folgende (Seite)

ff .....	fortfolgende (Seiten)
FB .....	Fachbereiche
FIFO.....	first in, first out
GMG .....	Gesetz zur Modernisierung der gesetzlichen Krankenversicherung
GKV .....	Gesetzliche Krankenversicherung
GwB .....	Geschäftswertbeitrag
ID .....	Identifikationsnummer
i.d.R.....	in der Regel
inkl. ....	inklusive
Int.....	Integer
IT.....	Informationstechnologie
i.v. ....	intravenös
kg .....	Kilogramm
KHG .....	Krankenhausfinanzierungsgesetz
L .....	Liter
LIFO .....	last in, first out
max. ....	maximal
min. ....	minimal
Min. ....	Minuten
mg .....	Milligramm
mmHg .....	Millimeter Quecksilbersäule
MOSAIK-M .....	Modellierung, Simulation und Animation von Informations- und Kommunikationssystemen in der Medizin
Mrd. ....	Milliarden
MRT .....	Magnetresonanztomographie
Nr. ....	Nummer
NSTEMI .....	Nicht-ST-Streckenhebungsinfarkt
o.g.....	oben genannt
OP.....	Operation
o.V. ....	ohne Verfasser
PCI.....	Perkutane Koronarintervention

PTCA .....	Perkutane transluminale Koronarangioplastie
Reha .....	Rehabilitation
S. ....	Seite
s.c. ....	subcutan
SEI .....	Software Engineering Institute
SGB V .....	Sozialgesetzbuch - Fünftes Buch
SIPOC.....	Supplier Input Process Output Customer
s.l. ....	sublingual
sog. ....	so genannt
Sono.....	Sonographie
Std. ....	Stunden
STEMI .....	ST-Streckenhebungsinfarkt
TIMI .....	Thrombolysis in Myocardial Infarction
U .....	Unit(s)
UML .....	Unified Modeling Language
u.a. ....	unter anderem
vgl. ....	vergleiche
VoC .....	Voice of Customer
z.B.....	zum Beispiel

## Abbildungsverzeichnis

Abbildung 1:	Entwicklung der Gesundheitsausgaben in Deutschland im Zeitraum 1997 bis 2005 .....	1
Abbildung 2:	Ergebnisse der Umfrage zum Thema Prozessqualität in Kliniken .....	4
Abbildung 3:	Nebenordnungen eines Systems .....	14
Abbildung 4:	Hierarchische Ordnung mit Nebenordnungen und Abgrenzung zur Umwelt .....	15
Abbildung 5:	Beispiel einer Input-Output-Matrix zur Erfassung von Wirkzusammenhängen.....	16
Abbildung 6:	Klassifizierung von Modellen nach dem Abbildungsmedium.....	18
Abbildung 7:	Klassifizierung von Modellen nach den Zustandvariablen.....	19
Abbildung 8:	Klassifizierung von Modellen nach ihrem Verwendungszweck .....	20
Abbildung 9:	Schematische Darstellung einer verändernden Steuerung ...	21
Abbildung 10:	Schematische Darstellung einer stabilisierenden Steuerung	21
Abbildung 11:	Reaktion einer Strecke mit Verzögerung .....	22
Abbildung 12:	Reaktion einer Strecke mit Totzeit T .....	23
Abbildung 13:	Reaktion einer integralen Strecke mit Sprunginput .....	23
Abbildung 14:	Zustandsvariable eines stetigen Systems .....	27
Abbildung 15:	Zustandsvariable eines diskreten Systems .....	27
Abbildung 16:	Schematische Darstellung des Modellbildungsprozesses.....	33
Abbildung 17:	Aufbau der Toolbox (Prinzipdarstellung) .....	36
Abbildung 18:	Kategorisierung von Unternehmensprozessen auf Hauptprozessebene .....	37
Abbildung 19:	Hauptprozesse und zugehörige Subprozesse des nächsten Prozesslevels.....	38
Abbildung 20:	Prozesse bis Prozesslevel 3 am Beispiel der Patientenaufnahme .....	39
Abbildung 21:	Prozessebene 3 und 4 am Beispiel der Notaufnahme .....	39

Abbildung 22:	Aufbau der Toolbox mit Prozesskomponenten unterschiedlichen Abstraktionsgrades (Prinzipdarstellung) ...	41
Abbildung 23:	Template zur inhaltlichen Beschreibung von Prozessen .....	42
Abbildung 24:	Prozessgebiete des CMMI .....	44
Abbildung 25:	Spezifische und generische Ziele und Praktiken am Beispiel des Prozessgebietes „Organizational Innovation and Deployment“ .....	45
Abbildung 26:	Reifegrade und zugehörige Prozessgebiete des CMMI staged .....	46
Abbildung 27:	Stufenmodell des CMMI staged .....	47
Abbildung 28:	Fähigkeitsgrade des CMMI continuous .....	48
Abbildung 29:	Kategorien und zugehörige Prozessgebiete des CMMI continuous .....	49
Abbildung 30:	Unterschiede zwischen kontinuierlicher und stufenförmiger Darstellung .....	50
Abbildung 31:	EFQM-Modell für Excellence .....	51
Abbildung 32:	Bewertungsmatrix der Befähiger-Kriterien .....	54
Abbildung 33:	Bewertungsmatrix der Ergebnis-Kriterien .....	55
Abbildung 34:	Faktorwerte der Befähiger- und Ergebniskriterien gem. EFQM-Modell .....	56
Abbildung 35:	Leistungs- und Verbesserungsdreieck für Prozesse .....	57
Abbildung 36:	Six Sigma als unterstützende Methode für CMMI .....	60
Abbildung 37:	Produktionssystem als Netz aus Stellen und Transitionen vor dem ersten Schaltvorgang .....	63
Abbildung 38:	Zustand des Produktionssystems nach dem ersten Schaltvorgang .....	63
Abbildung 39:	Symbole zur graphischen Beschreibung ereignisgesteuerter Prozessketten .....	64
Abbildung 40:	Verknüpfungsoperatoren und deren Aussageform .....	65
Abbildung 41:	Anwendungsfalldiagramm .....	71
Abbildung 42:	Aktivitätsdiagramm am Beispiel der Patientenaufnahme .....	72
Abbildung 43:	Sequenzdiagramm am Beispiel der Patientenaufnahme .....	73

Abbildung 44:	Paketdiagramm am Beispiel der Organisationseinheit „Zentrale Patientenaufnahme“ .....	74
Abbildung 45:	Vereinfachtes Klassendiagramm am Beispiel der Patientenaufnahme .....	75
Abbildung 46:	Zustandsdiagramm am Beispiel eines Kernspintomographen .....	76
Abbildung 47:	Arten von Automaten – Beispielhafte Darstellung .....	78
Abbildung 48:	Beispiel für benutzerindividuelle Bezeichnung von Automaten .....	79
Abbildung 49:	Bildschirmmaske zur Parametrisierung von Automaten .....	82
Abbildung 50:	Bildschirmmaske zur Beobachtung des Laufzeitverhaltens von Automaten .....	83
Abbildung 51:	Aufnahmekapazität der impliziten Buffer sowie Input-Output- Beziehungen zu den Automaten .....	84
Abbildung 52:	Beispielhafte Darstellung von Pfeilgewichten im Rahmen von Input-Output-Beziehungen zwischen Automaten.....	85
Abbildung 53:	Beispielhafte Darstellung von Input-Output-Beziehungen zwischen Automaten und expliziten Buffern .....	87
Abbildung 54:	Aufnahmekapazität der expliziten Buffer sowie Input-Output-Beziehungen zu den Automaten .....	87
Abbildung 55:	Personalmatrix .....	89
Abbildung 56:	Eingabemaske für Personengruppe und Arbeitsschicht .....	90
Abbildung 57:	Personenbezogener Kalender .....	90
Abbildung 58:	Erweiterte Personalmatrix .....	92
Abbildung 59:	Equipment Matrix – Zuordnung des Equipments zu den einzelnen Automaten.....	92
Abbildung 60:	Materialmatrix zur Steuerung des Verbrauchs von Material bzw. zur Herstellung eines Gutes.....	93
Abbildung 61:	Materialmatrix mit Angaben zum Verbrauchszeitpunkt, Materialpreis, Bestandsmenge und Bestellzeitpunkte .....	94
Abbildung 62:	Transaktionsmatrix mit zugehörigen Parametern .....	96
Abbildung 63:	Automatentypen Generate, Standard, Split und Assembly ...	97

Abbildung 64:	Maske zum Festlegen zeitbezogener, User-individueller Parameter.....	97
Abbildung 65:	Schematische Darstellung der adressatenabhängigen Verdichtung von Daten.....	99
Abbildung 66:	Aktivitätsverteilungsdiagramm für das Personal.....	101
Abbildung 67:	Gegenüberstellung des kumulierten, prozentualen Anteils des Personals und der zugehörigen prozentualen Auslastung.....	102
Abbildung 68:	Personenbezogene Auswertung bezüglich aktiver Zeiten, Engpässe und Kosten.....	103
Abbildung 69:	Prozessunterbrechungen durch mangelnde Verfügbarkeit von Personal einer bestimmten Personengruppe.....	104
Abbildung 70:	Beschäftigungsverhalten einzelner Personengruppen.....	104
Abbildung 71:	Matrix zur Feinanalyse während des Simulationslaufs.....	105
Abbildung 72:	Matrix zur Feinanalyse des Einsatzverhaltens einer bestimmten Person.....	105
Abbildung 73:	Aktivitätsverteilungsdiagramm für das Equipment.....	106
Abbildung 74:	Equipmentbezogene Auswertung bezüglich aktiver Zeiten, Ressourcenengpässe und Kosten.....	106
Abbildung 75:	Prozessunterbrechungen durch mangelnde Verfügbarkeit von Equipment.....	107
Abbildung 76:	Materialverbrauch je Prozessabschnitt.....	108
Abbildung 77:	Materialbezogene Auswertung bezüglich aktiver Zeiten, Ressourcenengpässe und Kosten.....	108
Abbildung 78:	Prozessunterbrechungen durch mangelnde Verfügbarkeit von Material.....	109
Abbildung 79:	Prozentual aufgetretene Engpässe und dafür verantwortliche Ursachen.....	110
Abbildung 80:	Auslöser für Engpässe je Prozessschritt.....	111
Abbildung 81:	Aktivitätsverteilungsdiagramm der Automaten.....	112
Abbildung 82:	Aktivitätsprofil je Automat.....	113
Abbildung 83:	Automatenbezogene Informationen.....	114
Abbildung 84:	Laufzeitverhalten von Automaten im Gantt-Diagramm.....	115



Abbildung 85:	Kapazitäts- und kostenbezogene Informationen für Buffer (1) .....	116
Abbildung 86:	Kapazitäts- und kostenbezogene Informationen für Buffer (2) .....	116
Abbildung 87:	Detailinformationen je Buffer .....	117
Abbildung 88:	Auswertung zeitbezogener User-Parameter .....	118
Abbildung 89:	Terminologie zum akuten Koronarsyndrom.....	122
Abbildung 90:	Röntgendarstellung einer Gefäßverengung mit Kontrastmittel .....	125
Abbildung 91:	Klinischer Behandlungspfad für das akute Koronarsyndrom.....	130
Abbildung 92:	Gesamtübersicht mit Prozesskategorien .....	131
Abbildung 93:	Darstellung des Teilprozesses „Patientenuntersuchung“ mit zugehörigen Subprozessen .....	132
Abbildung 94:	Darstellung des Teilprozesses „klinische Untersuchung“ am Beispiel der Elektrokardiographie bis Prozesslevel 4 ....	133
Abbildung 95:	Darstellung des Teilprozesses „bildgebende Verfahren“ am Beispiel der perkutanen Koronarintervention mittels Angiographie bis Prozesslevel 4 .....	134
Abbildung 96:	Darstellung des Teilprozesses „Patient behandeln“ mit zugehörigen Subprozessen bis Prozesslevel 3 .....	135
Abbildung 97:	Darstellung des Teilprozesses „Nicht-invasive Behandlung“ am Beispiel der Medikation bis Prozesslevel 4.....	135
Abbildung 98:	Darstellung des Teilprozesses „invasive Behandlung“ am Beispiel der Koronarangioplastie (PTCA) bis Prozesslevel 4 .....	136
Abbildung 99:	Modul „Empfang_Hauptmodul“ .....	139
Abbildung 100:	Aufbau des Moduls „Empfangsbereich“ .....	139
Abbildung 101:	Modul „Patientenaufnahme“ .....	140
Abbildung 102:	Substruktur des Moduls „Ambulante_Aufnahme“ .....	140
Abbildung 103:	Substruktur des Moduls „Erfassung_der_Patientendaten“ ..	141
Abbildung 104:	Baumstruktur der Bibliothek „Empfang“ .....	142
Abbildung 105:	Baumstruktur der Bibliothek „Untersuchung“ .....	143

Abbildung 106:	Modul „Blutabnahme_Hauptmodul“ .....	145
Abbildung 107:	Modul „Blutabnahme_vorbereiten“ .....	145
Abbildung 108:	Modul „Blutabnahme_durchfuehren“ .....	145
Abbildung 109:	Modul „Bluttransport“ .....	146
Abbildung 110:	Modul „Untersuchung_aerztlich“ .....	146
Abbildung 111:	Baumstruktur der Bibliothek „Fachbereiche“ .....	148
Abbildung 112:	Modul „Labor_Hauptmodul“ .....	149
Abbildung 113:	Modul „Untersuchung_vorbereiten“ .....	149
Abbildung 114:	Modul „Computertomographie_Hauptmodul“ .....	150
Abbildung 115:	Baumstruktur der Bibliothek „Bausteine_FB“ .....	150
Abbildung 116:	Modul „Untersuchung_veranlassen“ .....	151
Abbildung 117:	Baumstruktur der Bibliothek „Therapie“ .....	152
Abbildung 118:	Modul „Therapie_Hauptmodul“ .....	154
Abbildung 119:	Modul „Behandlung_vorbereiten“ .....	154
Abbildung 120:	Modul „Behandlung_durchfuehren“ .....	154
Abbildung 121:	Modul „Ergebnisse_dokumentieren“ .....	155
Abbildung 122:	Modul „Patientenakte_aktualisieren“ .....	155
Abbildung 123:	Durch das Modul „Therapieplatz“ erzeugte Grafik.....	156
Abbildung 124:	Komponenten des Moduls „Entlassung_Hauptmodul“ .....	157
Abbildung 125:	Repräsentative Belegungsstatistik der Patientenliegen im Behandlungsraum .....	160
Abbildung 126:	Verteilung der Wartezeiten (in Std.) am Beispiel ausgewählter Behandlungstage .....	161
Abbildung 127:	Verteilung der Patientenanzahl am Beispiel ausgewählter Behandlungstage .....	162
Abbildung 128:	Grundmodell (Layout) der onkologischen Tagesklinik.....	163
Abbildung 129:	Modell der onkologischen Tagesklinik und angrenzender Bereiche (Abstraktionsebene 1) .....	165
Abbildung 130:	Modell der Tagesklinik (Abstraktionsebene 2).....	166
Abbildung 131:	Belegungsstatistik am Beispiel der Ressource „Aerzte“ .....	167
Abbildung 132:	Modellkomponente „Patientenaufnahme“ .....	168
Abbildung 133:	Ambulante Patientenaufnahme als Subprozess der Patientenaufnahme .....	168

Abbildung 134:	Struktur des Warteraums .....	169
Abbildung 135:	Aufbau der Modellkomponente „Untersuchung“ .....	170
Abbildung 136:	Aufbau der Modellkomponente „Apotheke“ .....	170
Abbildung 137:	Aufbau des Moduls „Medikation versenden“ .....	171
Abbildung 138:	Baumstruktur des Modells der Tagesklinik im MLDesigner .	172
Abbildung 139:	Datenstruktur der Text-Datei und Vektoren.....	173
Abbildung 140:	Property Editor am Beispiel der Ressource „Aerzte“ .....	174
Abbildung 141:	Parametrisierung des Prozessschrittes „Empfang“ .....	174
Abbildung 142:	Übersicht der Durchlaufzeiten der einzelnen Modellkomponenten .....	175
Abbildung 143:	Minimale, maximal und durchschnittliche Dauer der Patientenaufnahme .....	177
Abbildung 144:	Prinzipdarstellung von Patienten-, Material- und Informationsflüssen sowie Personlaressourcen .....	179
Abbildung 145:	Soll-Modell der Tagesklinik.....	184
Abbildung 146:	Distanzmatrix.....	185
Abbildung 147:	Belegungsstatistik der Ärzte .....	187
Abbildung 148:	Belegungsstatistik der Pflegekräfte .....	187
Abbildung 149:	Belegungsstatistik der Behandlungsplätze (Therapieplätze) .....	188
Abbildung 150:	Prinzipdarstellung des Ablaufs genetischer Algorithmen.....	191
Abbildung 151:	Grundlegender Aufbau des Individuums .....	193
Abbildung 152:	Darstellung der Vorgehensweise bei Crossover.....	195
Abbildung 153:	Darstellung der Vorgehensweise bei Mutation .....	195
Abbildung 154:	Screenshot des mit dem ML Designer entwickelten Modells .....	197
Abbildung 155:	Einbestellungsreihenfolge für drei exemplarische Behandlungstage bei Anwendung des genetischen Algorithmus .....	199
Abbildung 156:	Veränderung der Liegenbelegung nach Optimierung der Patienteneinbestellung .....	201
Abbildung 157:	Optimiertes Modell zur Abbildung von bereits am Vortag erstellten Laborwerten.....	203

Abbildung 158: Optimierter Soll-Prozess für die Einbestellung von Patienten .....	204
Abbildung 159: Vergleich der Verbesserung der Liegenbelegung nach Implementierung aller Optimierungsschritte mit der Liegenbelegung nach Implementierung des ersten Optimierungsschrittes.....	205
Abbildung 160: Fragebogen zum Prozessreifegrad .....	229
Abbildung 161: Prozessbeschreibung der Echokardiographie .....	233
Abbildung 162: Prozessbeschreibung des Belastungs-EKGs .....	234
Abbildung 163: Graphical User Interface (GUI) des MLDesigners .....	236
Abbildung 164: Prozessbeschreibung der Besuchsvorbereitung (Ist-Prozess).....	237
Abbildung 165: Prozessbeschreibung der Patientenaufnahme (Ist-Prozess).....	238
Abbildung 166: Prozessbeschreibung Patientenaufnahme und Blutentnahme (Ist-Prozess).....	238
Abbildung 167: Prozessbeschreibung der Blutentnahme und Laboruntersuchung sowie ärztliche Untersuchung (Ist-Prozess).....	239
Abbildung 168: Prozessbeschreibung der Therapievorbereitung (Ist-Prozess).....	239
Abbildung 169: Prozessbeschreibung Therapiebeginn und Therapieüberwachung (Ist-Prozess).....	240
Abbildung 170: Prozessbeschreibung Therapie-Ende und Entlassung (Ist-Prozess).....	240
Abbildung 171: Prozessbeschreibung Nachbearbeitung (Ist-Prozess) .....	241
Abbildung 172: Prozessbeschreibung der optimierten Patienteneinbestellung (Soll-Prozess) .....	242
Abbildung 173a: Prozessbeschreibung der Patientenaufnahme (Soll-Prozess).....	243
Abbildung 173b: Prozessbeschreibung der Patientenaufnahme (Soll-Prozess).....	243
Abbildung 174: Prozessbeschreibung der Blutentnahme und Laboruntersuchung (Soll-Prozess).....	244

Abbildung 175: Prozessbeschreibung Therapievorbereitung (Soll-Prozess).....	244
Abbildung 176a: Prozessbeschreibung Therapiedurchführung und -überwachung (Soll-Prozess) .....	245
Abbildung 176b: Prozessbeschreibung Therapiedurchführung und -überwachung (Soll-Prozess) .....	245
Abbildung 177: Ende-Zeitpunkt der jeweiligen Transaktion (in Minuten) und zugehörige Patienten-ID .....	246

## Tabellenverzeichnis

Tabelle 1: Auswahl an wichtigen Novellierungen im Gesundheitswesen .....	235
Tabelle 2: Änderungen im Sozialgesetzbuch – Fünftes Buch (SGB V) ..	235
Tabelle 3a: Zusammenfassung der Funktionsübersicht ausgewählter Simulationssysteme .....	8
Tabelle 3b: Zusammenfassung der Funktionsübersicht ausgewählter Simulationssysteme .....	9
Tabelle 4a: Elemente der UML .....	68
Tabelle 4b: Elemente der UML .....	69
Tabelle 4c: Elemente der UML .....	70
Tabelle 5: Differentialdiagnosen zum akuten Koronarsyndrom .....	123
Tabelle 6: Ergebnisse ausgewählter Simulationsläufe .....	198
Tabelle 7: Tagesgenauer Vergleich der Werte im optimierten Soll-Modell mit den Werten des Ist-Modells am Beispiel von 5 aufeinander folgenden Arbeitstagen .....	206
Tabelle 8: Musterdatenblatt zur Datenerhebung .....	236
Tabelle 9: Tagesgenauer Vergleich der Ergebnisse der Simulation mit den Werten des Realsystems (Werte in Minuten).....	237
Tabelle 10: Vergleich der Ist-Werte mit den Werten des Simulationsmodells (Beispiel eines ausgewählten Arbeitstages).....	241

## 1. Einleitung

### 1.1 Problemstellung

In den letzten Jahren sind die Gesundheitsausgaben in Deutschland kontinuierlich gestiegen (vgl. Abb. 1). Mit einem Anteil von 10,7% am Bruttoinlandsprodukt im Jahr 2005 hat Deutschland im Vergleich eines der teuersten Gesundheitssysteme in Europa [StaB08; GeBu08; OECD08].

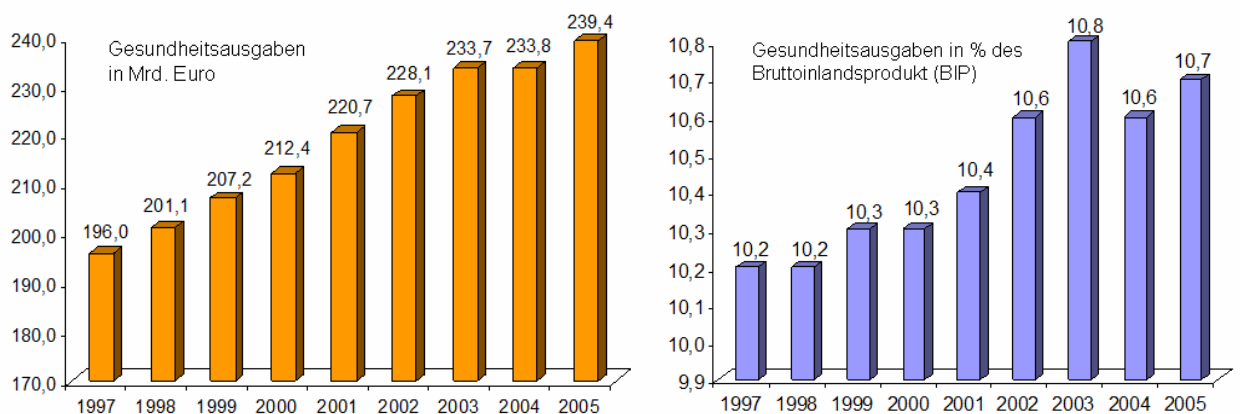


Abbildung 1: Entwicklung der Gesundheitsausgaben in Deutschland im Zeitraum 1997 bis 2005

Um einem weiteren Anstieg der Gesundheitsausgaben entgegenzuwirken, wurden in den letzten Jahren von der Bundesregierung die gesetzlichen Rahmenbedingungen<sup>2</sup> sukzessive verändert. Eine der wesentlichsten Neuerungen wurde mit dem Gesundheitsreformgesetz des Jahres 2000 beschlossen: Ein pauschaliertes Entgeltsystem für voll- und teilstationäre Krankenhausleistungen auf Basis der Diagnosis Related Groups (DRGs) löst seit dem Jahr 2004 die bisherige tagesgleiche Pflegesatzvergütung ab. Von 1972 bis 2003 gab das so genannte Selbstkostendeckungsprinzip das Prinzip der Krankenhausfinanzierung vor. Im Rahmen des Selbstkostendeckungsprinzips erhielten Krankenhäuser für die stationäre Betreuung von Patienten je Behandlungstag eine tagesgleiche Pauschalvergütung – unabhängig vom jeweiligen Behandlungsaufwand. Krankenhäuser hatten zu dieser Zeit folglich wenig Anreiz, Patienten möglichst effizient zu behandeln, da mit der Dauer des

<sup>2</sup> Siehe Anhang 1, Tabelle 1 und Tabelle 2.

stationären Aufenthalts auch der Erlös für das betreffende Krankenhaus stieg. Dies führte zu unnötig hohen Verweildauern der Patienten im Krankenhaus und letztendlich zu Zusatzkosten für das deutsche Gesundheitssystem. Seit der verbindlichen Einführung der DRGs<sup>3</sup> im Jahre 2004, werden die Leistungen von Krankenhäusern daher nicht mehr tageweise abgerechnet, sondern pauschal je nach Krankheitsbild.<sup>4</sup> Die Höhe der Fallpauschale richtet sich nach Krankheitsart und Schwere der Erkrankung.<sup>5</sup> Die DRGs führen gegenüber der tagesgleichen Pflegesatzvergütung deshalb zu völlig anderen ökonomischen Anreizen für Krankenhäuser. Grundsätzlich gilt im DRG-System: Je länger die Behandlungsdauer<sup>6</sup> eines Patienten, desto kleiner die Gewinnmarge, die das Krankenhaus für den jeweiligen Fall erwirtschaftet. Nicht zuletzt aufgrund der Einführung der DRGs kam es bereits in den letzten Jahren zur Schließung von defizitären Kliniken im gesamten Bundesgebiet [Klin06; Aertz06; Spie06; Domi06]. Krankenhäuser müssen deshalb versuchen, Ressourcen künftig effizienter einzusetzen, um über die Verkürzung von Behandlungs- und Wartezeiten die Verweildauer der Patienten zu senken sowie die entstehenden Kosten je Behandlungsfall zu minimieren. Entscheidend für das Erreichen dieser Ziele ist letztendlich die Qualität der Prozesse, also die Effizienz und Effektivität der Arbeitsschritte, die während der Behandlung eines Patienten durchlaufen werden [Röh03 S.5; MoEi03 S.10; Joc03 S.9; Grei04 S.21; Ame04 S.281; Kien02 S.20f; Kuch05 S.1ff]. Im Bereich der Prozessqualität haben Kliniken deutliches Verbesserungspotential. Dies wird auch durch eine im Rahmen dieser Arbeit durchgeführte Umfrage<sup>7</sup> an Kliniken belegt. Befragt wurden ärztliches Personal, Pflegefachkräfte und Mitarbeiter der Verwaltung von insgesamt 15 Kliniken. Der hierfür entwickelte Fragebogen (vgl. Anhang 1, Abb. 160) besteht aus 11 Fragen zu Themenkomplexen, die einen Einfluss auf die Prozessqualität haben.

---

<sup>3</sup> Die DRG wird auch als fallbezogene Vergütungsform bezeichnet.

<sup>4</sup> Die bisherigen Ausnahmen bilden psychiatrische und psychosomatische Erkrankungen.

<sup>5</sup> Im Rahmen des DRG-Systems müssen die behandelnden Ärzte jedem Behandlungsfall sog. Haupt- und Nebendiagnosen zuordnen und diese ausführlich dokumentieren. Dabei findet ein unterschiedlicher Schweregrad der Erkrankung in der Vergütung der Krankenhäuser Berücksichtigung.

<sup>6</sup> Bezogen auf den jeweiligen Fall.

<sup>7</sup> Ausgewertet wurden 92 beantwortete Fragebögen von 15 Kliniken in Deutschland.



Die gestellten Fragen sind nachfolgend aufgelistet:

- (1) Wie beurteilen Sie die qualitative Prozessbeschreibung / -dokumentation in Ihrer Klinik?
- (2) Ist ein Prozessverantwortlicher benannt?
- (3) Sind Prozesskennzahlen zur Messung der Prozessperformance definiert worden?
- (4) Werden definierte Prozesskennzahlen kontinuierlich gemessen?
- (5) Werden ausgehend von den Prozesskennzahlen Verbesserungsmaßnahmen abgeleitet?
- (6) Wie beurteilen Sie die Prozessmanagement-Erfahrung in Ihrer Klinik?
- (7) Sind Ihre vorhandenen Prozesse ausreichend durch IT-Verfahren unterstützt?
- (8) Werden aus Ihrer Sicht bestehende Prozesse eingehalten und genutzt?
- (9) Wie schätzen Sie die abteilungsübergreifende Prozessintegration ein?

Um die grundsätzliche Einschätzung der Befragten zur Höhe des möglichen Verbesserungspotentials durch Prozessoptimierung sowie zur Notwendigkeit der Prozessoptimierung unter Wettbewerbsaspekten zu erhalten, wurden zwei ergänzende Fragen zu diesen Themen gestellt:

- (10) Wie hoch schätzen Sie das Verbesserungspotential der vorhandenen Prozesse?
- (11) Glauben Sie, dass das Thema Prozessoptimierung ein ausschlaggebender Faktor für die langfristige Wettbewerbspositionierung Ihrer Klinik ist?

Bei den beiden ergänzenden Fragen 10 und 11 gaben alle befragten Personen an, dass Prozessoptimierung eines der wichtigen, künftigen Kernthemen ist, um die Wettbewerbsfähigkeit langfristig zu sichern. Ebenfalls eindeutig war die Aussage zur Einschätzung des Verbesserungspotentials der vorhandenen Prozesse: 86% der Befragten gaben an, dass sie insgesamt ein hohes Verbesserungspotential bei den vorhandenen Prozessen sehen.

Aus den Fragen 1 bis 9 wurde die nachstehende Grafik (vgl. Abb. 2) entwickelt. Diese spiegelt die Zusammenfassung der ausgewerteten Antworten wider und zeigt in allen neun Dimensionen Verbesserungspotential. Die Ergebnisse der Umfrage zeigen zudem, dass die öffentlichen Kliniken in Bezug auf die Prozeßqualität i.d.R. hinter den privaten Kliniken zurückliegen.

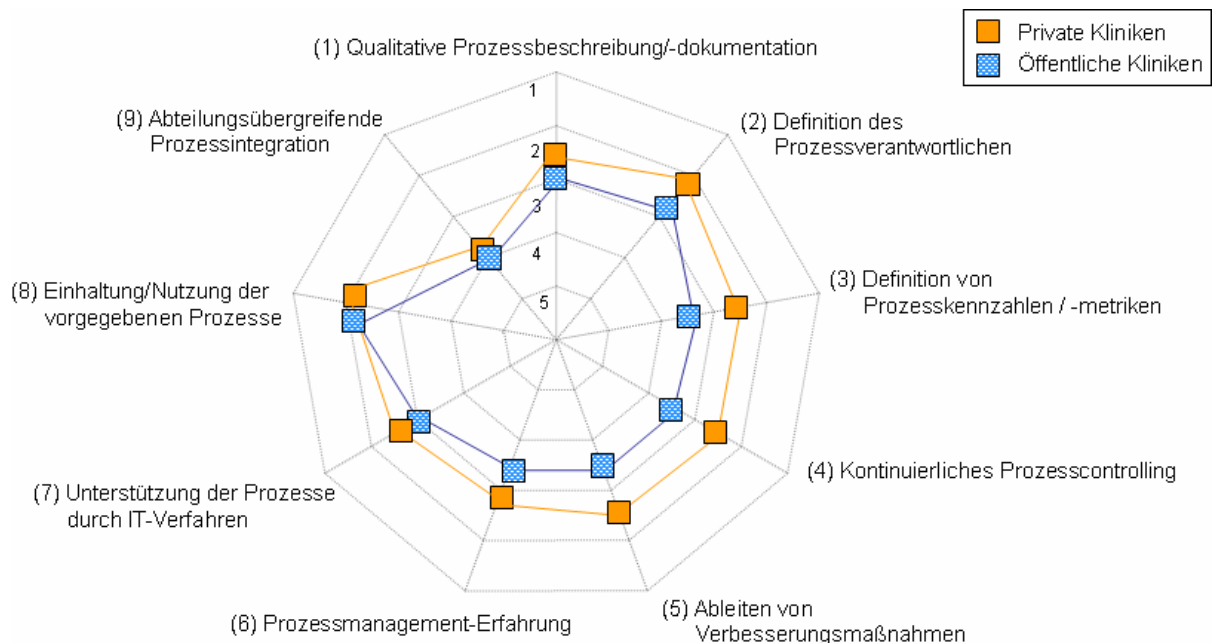


Abbildung 2: Ergebnisse der Umfrage zum Thema Prozessqualität in Kliniken<sup>8</sup>

Um über Prozessoptimierung sowohl die Verweildauern von Patienten zu reduzieren als auch Kosten zu senken, benötigen Kliniken geeignete Tools und Methoden. Prozesse erreichen aufgrund der Anzahl an Input- und Output-Parametern, der Anzahl an Schnittstellen zwischen Kern- und Querschnittsprozessen sowie der Anzahl an Subprozessen unterschiedlicher hierarchischer Ebenen einen hohen Grad an Komplexität. Herkömmliche Methoden, wie z.B. Gap-Analysen, Benchmarking und mathematische Modelle, können aufgrund ihres statischen Charakters sowie der Vielzahl zu berücksichtigender Prozessparameter nur bedingt zur Optimierung von Prozessen beitragen. Das dynamische Verhalten von Prozessen kann nur adäquat abgebildet werden, wenn die zugrunde liegenden Modelle ebenfalls dynamisch sind. Dynamische Modelle können mit Simulationssystemen

<sup>8</sup> Skala: 1 = sehr gut bis 5 = mangelhaft.

entwickelt und analysiert werden. Der Einsatz von Simulationssystemen zur Durchführung von Simulationsstudien hat zudem den Vorteil, dass die Auswirkungen veränderter Prozesse, wie auch das Design neuer Prozesse, vor deren Implementierung im Realsystem in einem ablauffähigen Modell „gefahrlos“ getestet werden können. Gerade im Gesundheitswesen ist dies ein wichtiger Faktor, der für den Einsatz von Simulationssystemen spricht. Werden neue oder veränderte Prozesse implementiert, bevor deren mögliche Auswirkungen auf das Realsystem ausreichend in einem Modell getestet wurden, können ungewollte Nebeneffekte im „worst-case“ zu gesundheitlichen Schäden bei Patienten führen. Dass Simulationsstudien im Gesundheitswesen erfolgreich zur Optimierung von Prozessen eingesetzt werden, zeigen diverse durchgeführte Studien, wie z.B. in [STB99; GCY02; BaSe01; WeHo99; Kobl06; Wuer06; UniB06; PMC06] beschrieben. Um diese komplexen Modelle zu entwickeln, werden geeignete, leistungsfähige Simulationssysteme benötigt. Für den Einsatz im klinischen Umfeld sollten diese insbesondere folgenden Ansprüchen genügen:

- (1) Patienten, die in einer Klinik eintreffen, lösen Ereignisse zu bestimmten Zeitpunkten aus. Simulationssysteme müssen deshalb in der Lage sein, ereignisorientierte Modelle abzubilden und zu simulieren.
- (2) Zur Darstellung stochastischer Variablen im Zeitablauf müssen Simulationssysteme in der Lage sein, Pseudo-Zufallszahlenreihen über entsprechende Wahrscheinlichkeitsverteilungen zu erzeugen.<sup>9</sup> Auf diese Art und Weise können z.B. Patientenströme erzeugt werden.
- (3) Wartezeiten beeinflussen die Verweildauer von Patienten. Gerade diese müssen Krankenhäuser - wie bereits weiter oben ausgeführt - verkürzen. Simulationssysteme sollten deshalb in der Lage sein, Warteschlangen<sup>10</sup> im Modell darzustellen und diese auszuwerten.
- (4) Ressourcenengpässe führen bei der Behandlung von Patienten zu längeren Wartezeiten und damit insgesamt auch zu einer längeren Durchlaufzeit. Für modellierte Prozesse sowie deren Subprozesse müssen deshalb Ressourcen, wie z.B. Personal und medizinische Geräte, zugeordnet werden können. Die Ressourcenbeanspruchung muss für Analysezwecke

---

<sup>9</sup> Monte Carlo Simulation.

<sup>10</sup> Sog. Queues.

im Zeitablauf dargestellt werden können. Ressourcenengpässe müssen ersichtlich sein.

- (5) Simulationssysteme müssen geeignete Tools zur Fehlersuche<sup>11</sup> und Analyse von modellierten Systemen bereitstellen. Dies sind wichtige Hilfsmittel, um zum einen zu überprüfen, ob ein Modell das gewünschte Modellverhalten aufweist. Zum anderen dienen sie dazu, Fehler im Modell<sup>12</sup> aufzufinden und zu beseitigen.
- (6) Simulationssysteme müssen in der Lage sein, hierarchische Prozesse<sup>13</sup> abzubilden. Hierarchisierung ist notwendig, um Prozesse mit den zugehörigen Subprozessen ausreichend detailliert abbilden zu können.
- (7) Simulationssysteme sollten ein validiertes Prozessreferenzmodell zur Modellierung und Simulation klinischer Prozesse bereitstellen. Dieses sollte aus wieder verwendbaren, ausführbaren Modellbausteinen bestehen, die in klar strukturierten Bibliotheken hinterlegt sind. Ein Prozessreferenzmodell ist zum einen Grundlage für die Vergleichbarkeit von Ergebnissen, die z.B. im Rahmen von Benchmark-Studien erzielt werden. Zum anderen wird durch die Verwendung von standardisierten und wieder verwendbaren Modellkomponenten die Entwicklungszeit von Simulationsmodellen erheblich verkürzt.

Im Rahmen dieser Arbeit wurde eine Analyse durchgeführt, um festzustellen, ob und wenn ja welche am Markt erhältlichen Simulationssysteme die oben genannten Anforderungen erfüllen. Dabei wurden folgende Simulationssysteme betrachtet: (1) MedModel<sup>14</sup> [Pro06a], (2) Process Simulator<sup>15</sup> [Pro06b] (3) IPA-business-toolkit<sup>16</sup> [IDW06], (4) MOSAIK-M<sup>17</sup> [UniB06], (5) Aris Simulation<sup>18</sup> [IDS06a], (6) Aris Business Optimizer<sup>19</sup> [IDS06b] und (7) MLDesigner<sup>20</sup> [MLD04].

---

<sup>11</sup> Sog. Debugger.

<sup>12</sup> Z.B. fehlende Verknüpfungen zwischen Modellkomponenten.

<sup>13</sup> D.h. Prozesse unterschiedlicher Prozessebenen.

<sup>14</sup> MedModel ist ein Produkt der Firma ProModel Corporation, Utah, USA.

<sup>15</sup> Process Simulator ist ein Produkt der Firma ProModel Corporation, Utah, USA.

<sup>16</sup> IPA-business-toolkit ist ein Produkt des Fraunhofer-Instituts für Produktionstechnik und Automatisierung.

<sup>17</sup> MOSAIK-M wurde am Institut für medizinische Informatik der Universität Braunschweig entwickelt.

<sup>18</sup> Aris Simulation ist ein Produkt der Firma IDS Scheer AG, Saarbrücken.

<sup>19</sup> Aris Business Optimizer ist ein Produkt der Firma IDS Scheer AG, Saarbrücken.

<sup>20</sup> MLDesigner ist ein Produkt der Firma MLDesign Technologies, Inc., USA.

Das Ergebnis der Analyse ist in den nachfolgenden Tabellen 3a und 3b dargestellt. Die Analyse zeigt, dass keines der untersuchten Simulationssysteme die o.g. Anforderungen vollständig erfüllt. Insbesondere gilt, dass keines der genannten Simulationssysteme auf Basis eines Prozessreferenzmodells standardisierte und wieder verwendbare Modellkomponenten zur Abbildung klinischer Prozesse zur Verfügung stellt. An dieser Stelle setzt die vorliegende wissenschaftliche Arbeit an.

Simulationstool	MedModel	ProcessSimulator	IPA-business-toolkit	Mosaik-M
<b>Kriterium</b>				
<b>Ereignisorientierte Simulation</b>	Ja	Ja	Ja	Ja
<b>Monte-Carlo-Simulation</b>	Ja	Ja	Ja	Ja
<b>Darstellung und Auswertung von Warteschlangen</b>	Ja	Ja	Ja	Ja
<b>Verwaltung von Ressourcen</b>	Ja	Ja	Ja	Ja
<b>Abbildung hierarchischer Prozesse</b>	Nein	Nein	Ja	Ja
<b>Debugger und Analysewerkzeuge</b>	Debugger vorhanden; Analysemöglichkeiten sind durch standardisierte Auswertungen (grafisch und tabellarisch) von Warteschlangen, Ressourceneinsatz/-belegung sowie Darstellung von Prozesskosten möglich.	Kein Debugger vorhanden; Analysemöglichkeiten sind durch standardisierte Auswertungen (grafisch und tabellarisch) von Warteschlangen, Ressourceneinsatz/-belegung sowie Darstellung von Prozesskosten möglich.	Debugger vorhanden; teils besteht die Möglichkeit standardisierter Auswertungen (z.B. Ressourcenauslastung).	Debugger vorhanden. Grafische Auswertungen innerhalb Mosaik/M sind bedingt möglich. Üblich ist der Export von Daten nach Excel.
<b>Wieder verwendbare Modellkomponenten zur Modellierung und Simulation klinischer Prozesse</b>	Nicht vorhanden	Nicht vorhanden	Nicht vorhanden	Einzelne Modellkomponenten zur Abbildung klinischer Prozesse existieren. Diese besitzen einen hohen Abstraktionsgrad. Ein zugrunde liegendes klinisches Prozessreferenzmodell existiert nicht. Es existiert keine strukturierte Modelldatenbank.
<b>Anmerkungen</b>	MedModel wurde zur Layout-Planung von Kliniken entwickelt und wird bis heute vom Hersteller primär dafür eingesetzt.			Ursprünglicher Fokus von Mosaik/M: Softwareentwicklung/-design; Informationssysteme. Mittlerweile wurden auch vereinzelt Projekte im Bereich der Prozessoptimierung durchgeführt.

Tabelle 3a: Zusammenfassung der Funktionsübersicht ausgewählter Simulationssysteme

Simulationstool	Aris Simulation	Business Optimizer	MLDesigner
<b>Kriterium</b>			
<b>Ereignisorientierte Simulation</b>	Nein	Nein	Ja
<b>Monte-Carlo-Simulation</b>	Nein	Nein	Ja
<b>Darstellung und Auswertung von Warteschlangen</b>	Nein	Nein	Ja
<b>Verwaltung von Ressourcen</b>	Nur eingeschränkt möglich (Ressourcen können einzelnen Prozessschritten zum Zweck der Prozesskostenrechnung starr zugeordnet werden)	Nur eingeschränkt möglich (Ressourcen können einzelnen Prozessschritten zum Zweck der Prozesskostenrechnung starr zugeordnet werden)	Ja
<b>Abbildung hierarchischer Prozesse</b>	Nein	Ja	Ja
<b>Debugger und Analysewerkzeuge</b>	Kein Debugger vorhanden; Analysemöglichkeiten sind nur bedingt durch die vorhandenen tabellarischen Auswertungen möglich; Auswertungen bzgl. Warteschlangen sowie Ressourcenbelegungen sind nicht möglich.	Debugger vorhanden; Auswertungen von Warteschlangen und Ressourcenbelegung im Zeitablauf sind nicht möglich. Lediglich Prozesskosten können ausgewertet werden.	Debugger und Analysewerkzeuge u. a. in Form von sog. Probes, Breakpoints sowie diversen Grafiken z.B. zur Analyse von Warteschlangen und Ressourceneinsatz/-belegung vorhanden.
<b>Wieder verwendbare Modellkomponenten zur Modellierung und Simulation klinischer Prozesse</b>	Modelle sind nicht ausführbar. Sie können deshalb auch nicht validiert werden.	Modelle sind nicht ausführbar. Sie können deshalb auch nicht validiert werden.	Es sind umfangreiche Bibliotheken vorhanden, jedoch keine spezifischen, wieder verwendbaren Modellkomponenten zur Simulation klinischer Prozesse.
<b>Anmerkungen</b>	Aris Simulation kommt laut IDS Scheer AG aufgrund seiner eingeschränkten Funktionalität nur noch selten zum Einsatz.	Der Business Analyzer wurde für die Prozesskostenrechnung entwickelt. Im Kern ist der Business Analyzer daher ein reines Rechen- und zur Prozesssimulation nicht geeignet.	Der MLDesigner ist in der Lage, Berechnungen automatisch auf verschiedene Rechner zu verteilen. So können auch komplexe Prozesse, wie dies bei der Simulation klinischer Systeme i.d.R. der Fall ist, in adäquater Zeit simuliert und ausgewertet werden.

Tabelle 3b: Zusammenfassung der Funktionsübersicht ausgewählter Simulationssysteme

## 1.2 Zielsetzung und Vorgehensweise der vorliegenden Arbeit

Ziel der vorliegenden Arbeit ist die Entwicklung eines validierten Prozessreferenzmodells, welches aus standardisierten und wieder verwendbaren Modellkomponenten besteht. Die Modellkomponenten werden mit dem für die vorliegende Aufgabenstellung geeigneten Simulationssystem MLDesigner entwickelt und in definierten Bibliotheken des MLDesigner strukturiert hinterlegt. Die Modellkomponenten werden modular aufgebaut und mit den entsprechenden klinischen Begriffen aus der Krankenhauspraxis bezeichnet. Letzteres erleichtert die Akzeptanz von Simulationsmodellen im klinischen Umfeld, da sich klinisches Personal aufgrund der ihnen dann vertrauten Begrifflichkeiten besser mit entwickelten Modellen identifizieren kann.

Die Arbeit ist wie folgt aufgebaut: In Kapitel 2 werden die für die vorliegende Arbeit relevanten Begriffe zum Thema Systemsimulation sowie mögliche Einsatzgebiete von Simulationsmodellen erläutert. Kapitel 3 beschäftigt sich mit dem Thema der Prozessmodellierung. Die Begriffe „Modell“ und „Prozess“ werden definiert. Am Beispiel des klinischen Kernprozesses wird die Hierarchisierung von Prozessen auf unterschiedlichen Prozessebenen erläutert. Anschließend werden Methoden zur Messung und Verbesserung der Prozessqualität sowie Techniken zur Modellierung von Prozessen vorgestellt. In Kapitel 4 werden grundlegende Komponenten von Simulationssystemen, die zur Modellierung, Simulation und Analyse von Prozessen notwendig sind, dargestellt und anhand von Beispielen aus dem klinischen Umfeld erläutert. In Kapitel 5 findet ausgehend von einem ausgewählten klinischen Behandlungspfad die standardisierte Beschreibung klinischer Prozesse statt. Die hier beschriebenen Prozesse werden zudem in eine Ablaufreihenfolge gebracht. Darauf aufbauend wird in Kapitel 6 ein validiertes Prozessreferenzmodell entwickelt. Dieses besteht aus wieder verwendbaren Modellkomponenten, die mit dem Simulationssystem MLDesigner entwickelt und in einer geeigneten Bibliotheksstruktur hinterlegt werden. Die Modellkomponenten werden anschließend für die Simulation und



Optimierung der Prozesse am Beispiel einer medizinischen Tagesklinik eingesetzt. Letzteres ist in Kapitel 7 dargestellt.

## **2. Systemsimulation**

Der Begriff der Systemsimulation setzt sich aus den Begriffen „System“ und „Simulation“ zusammen. Kapitel 2.1 erläutert anhand der kybernetischen System- und Modelltheorie den Systembegriff. Kapitel 2.2 erläutert den Begriff der Simulation und geht auf die möglichen Anwendungsgebiete von Simulationen ein.

### **2.1 Kybernetische System- und Modelltheorie**

#### **2.1.1 Grundlagen der kybernetischen System- und Modelltheorie**

Das fundamentale Theorem der kybernetischen<sup>21</sup> Systemtheorie erklärt das Universum als ein Wirkgefüge, bestehend aus materiellen Komponenten. Diese Komponenten befinden sich in Wechselwirkung unterschiedlicher Intensität, durch die sie im Zeitablauf verändert werden.

Ein kybernetisches System ist eine Menge von materiellen Komponenten. Sie sind durch Interaktions- und Kombinationsbeziehungen miteinander verbunden. Komponenten repräsentieren entweder Teilsysteme des betrachteten Systems oder aber dessen Elemente. Für Teilsysteme gilt analog die obige Definition. Elemente im Sinne des kybernetischen Systembegriffs lassen sich nicht weiter unterteilen und sind somit Elementarteilchen, die Grundbausteine der Materie. Von Komponenten gehen Wirkungen aus, die andere Komponenten beeinflussen. Umgekehrt können sie auch Wirkungen von anderen Komponenten empfangen, die dann die eigenen Wirkpotentiale<sup>22</sup> verändern. Interaktionsbeziehungen beschreiben die Einflussnahme von Komponenten auf andere Komponenten – was eine ständige Veränderung des Systems im Zeitablauf impliziert. Kombinationsbeziehungen beschreiben die momentane Zusammensetzung von Komponenten eines Systems. Der kybernetische Systembegriff ist deshalb zugleich ein dynamischer wie auch ein hierarchischer Systembegriff. Zuletzt ist der kybernetische Systembegriff auch ein materieller Systembegriff. Nur Materie kann Energie speichern und damit auf umgebende

---

<sup>21</sup> Das griechische Wort „kybernetes“ bedeutet Steuermann.

<sup>22</sup> Wirkpotentiale sind die in der Materie enthaltene Energie.

Materie einwirken. Zustand<sup>23</sup>, Struktur<sup>24</sup> und Dynamik<sup>25</sup> sind Eigenschaften jeglichen materiellen Systems [Niem77 S.1ff; BeDe94 S.24; Stei91 S.8; Patz82 S.19; SSK90 S.13ff; Gord72 S.11; Salz07 S.2 u. S.7].

Materielle Systeme können in offene und geschlossene Systeme eingeteilt werden. Offene Systeme haben mindestens eine Interaktionsbeziehung zu einem umgebenden System, während geschlossene Systeme keinerlei Interaktionsbeziehungen zu anderen Systemen haben [Kral96 S.8ff; Stei91 S.16].

## **2.1.2 Systemerfassung und -modellbildung**

### **2.1.2.1 Systemzustände und -strukturen**

Die Erfassung von Systemen beinhaltet die Erfassung des Systemzustandes, der Systemstruktur und die Abgrenzung gegen andere Systeme.

Zur Erfassung des Systemzustandes können sog. Zustandsvariablen herangezogen werden. Zustandsvariablen sind nichts anderes als Attributklassen des betrachteten Systems, wie z.B. Kosten, Erträge, Alter oder Gewicht. Attribute sind demnach die Ausprägungen der Zustandsvariablen. Sie können in unterschiedliche Kategorien eingeteilt werden. Attribute werden als quantitativ bezeichnet, wenn deren Merkmalsausprägung eine Zahl darstellt. Demgegenüber sind die Merkmalsausprägungen bei qualitativen Attributen i.d.R. durch einen verbalen Ausdruck beschrieben (z.B. sehr gut, gut, mittel, schlecht; hübsch, hässlich). Durch Quantifizierung, also dem Ersetzen des verbalen Ausdrucks durch eine Zahl, kann ein qualitatives Merkmal formal in ein quantitatives Merkmal übergeführt werden.<sup>26</sup> Quantifizierte Attribute lassen sich dann in Nominalskalen einordnen. Wenn die Merkmalsausprägungen unterschieden und zugleich in eine Rangordnung gebracht werden können, lassen sich quantitative Attribute in eine Ordinalskala einordnen.<sup>27</sup> Die Rangordnung spiegelt sich in den

---

<sup>23</sup> Bezeichnet das Wirkpotential.

<sup>24</sup> Bezeichnet das Wirkgefüge.

<sup>25</sup> Bezeichnet den Zeitablauf des Systems.

<sup>26</sup> Z.B. sehr gut = 1; gut = 2, mittel = 3; hässlich = 1, hübsch = 2.

<sup>27</sup> Z.B. Messung der Intelligenz.

zugeordneten Zahlen wider. Quantitative Attribute können in eine Kardinalskala<sup>28</sup> eingeordnet werden, wenn die Merkmalsausprägungen in eine Rangordnung gebracht werden können und zugleich das Ausmaß bestimmt werden kann, in dem sich je zwei Ausprägungen unterscheiden. Dies gilt z.B. für monetäre Größen [Niem77 S.40ff; BaBa89 S.6f; Sali93 S.24].

Laut Niemeyer ist „die Struktur eines Systems [...] durch die Menge der zwischen den Systemkomponenten bestehenden Interaktionsbeziehungen und die Menge der zwischen den Komponenten und dem System bestehenden Kombinationsbeziehungen definiert“ [Niem77 S.46]. Zur Erfassung der Systemstruktur schlägt Niemeyer vier Teilschritte vor:

- (1) Erfassung der Ordnung des Systems durch eine qualitative Analyse der Kombinationsbeziehungen
- (2) Erfassung des Wirkzusammenhangs des Systems durch eine qualitative Analyse der Interaktionsbeziehungen
- (3) Erfassung der einzelnen Kombinationsbeziehungen durch eine quantitative Analyse
- (4) Erfassung der einzelnen Interaktionsbeziehungen durch eine quantitative Analyse

Für die vorliegende Arbeit sind Nebenordnungen (z.B. „parallel“ angeordnete klinische Fachbereiche wie Radiologie, Echokardiographie, Elektrokardiographie) und hierarchische Ordnungen (Hierarchisierung klinischer Prozesse) von Bedeutung. Nebenordnungen bilden Systeme nur einstufig ab (vgl. Abb. 3).

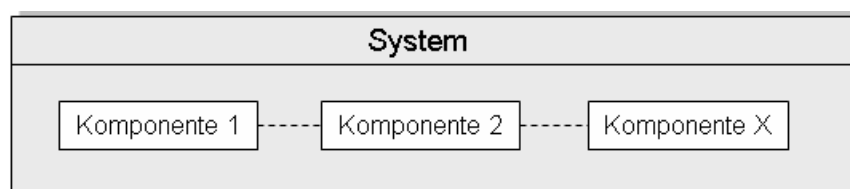


Abbildung 3: Nebenordnungen eines Systems

---

<sup>28</sup> Auch metrische Skala genannt.

Hierarchische Ordnungen hingegen bilden ein System mittels einer n-stufigen Auflösung ab. Abbildung 4 zeigt eine hierarchische Ordnung mit Nebenordnungen. Beginnend mit den unmittelbaren Hauptkomponenten des Systems, setzt sich die Darstellung mit den Subkomponenten fort. Jede Subkomponente kann wiederum weitere Subkomponenten haben. Eine Gruppe von Subkomponenten, welche derselben Komponente unmittelbar untergeordnet ist, bildet eine Nebenordnung [Niem77 S.46ff; Schü98 S.37; KFG99 S.23]. Ordnungskonzepte bilden die Grundlage für die Erfassung einzelner Kombinationsbeziehungen sowie für die Erfassung des Wirkgefüges.

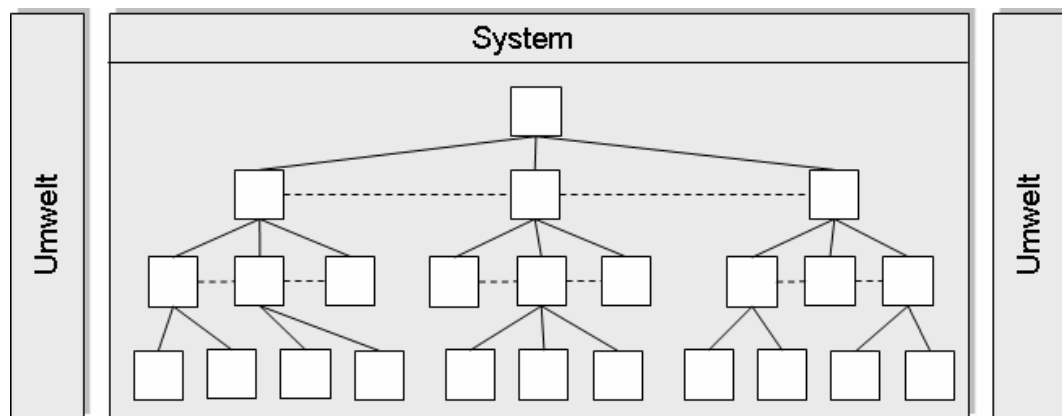


Abbildung 4: Hierarchische Ordnung mit Nebenordnungen<sup>29</sup> und Abgrenzung zur Umwelt<sup>30</sup>

Interaktionsbeziehungen werden laut der allgemeinen Systemtheorie durch Energieströme<sup>31</sup> realisiert. Energieströme fließen in die Komponenten hinein bzw. aus ihnen heraus. Der Wirkzusammenhang in einem System kann in einer Input-Output-Matrix durch Eintragen der entsprechenden Wirkzusammenhänge erfasst werden (vgl. Abb. 5).

<sup>29</sup> Nebenordnung = gestrichelte Linien.

<sup>30</sup> Die Umwelt eines Systems entspricht der Menge der von der Betrachtung ausgeschlossenen (Sub-/Neben-)Systemen.

<sup>31</sup> Diese beinhalten auch Informationsströme.

		Komponenten (Empfänger)			
		1	2	3	n
Komponenten (Sender)	1				
	2				
	3				
	n				

Abbildung 5: Beispiel einer Input-Output-Matrix zur Erfassung von Wirkzusammenhängen

Nachdem eine Interaktionsbeziehung die Abhängigkeit einer Zustandsvariablen von einer oder mehreren Zustandsvariablen des Systems beschreibt<sup>32</sup>, kann sie mittels der Gleichung

$$x_i = f_i(t, x_1, x_2, \dots, x_n) \quad \text{mit } i = (1, \dots, n); \quad t = \text{Zeitvariable};$$

$$x = \text{Zustandsvariable eines Systems}$$

dargestellt werden. Um die Funktion  $f_i$  zu ermitteln, sind Kenntnisse der Umwandlungsprozesse, die in dem betrachteten System stattfinden, notwendig. Teils sind Umwandlungsprozesse bekannt (z.B. chemische Reaktionsgleichungen, Energie-, Beschleunigungs- und Schwingungsgesetze, Kirchhoff'sche Schaltungsregeln) bzw. ergeben sich unmittelbar aufgrund der Systemkonstruktion (z.B. Fortschreibung von Beständen, logische Schaltfunktionen in mechanischen oder elektronischen Schaltwerken). Sind die Umwandlungsprozesse nicht bekannt, müssen Zusammenhänge experimentell ermittelt werden, indem unabhängige Variablen variiert und die Auswirkungen auf die abhängigen Variablen ermittelt werden. Hier kann die Regressionsrechnung – auch Dependenzanalyse genannt - entsprechend angewandt werden [Niem77 S.50ff, BaBa89 S.42ff].

<sup>32</sup> Hier liegt ein kausales Modell zugrunde.

### 2.1.2.2 Systemabbildung

Aufgrund der Komplexität von Systemen sind Interaktions- und Kombinationsbeziehungen nicht immer sofort und eindeutig zu erkennen. Deshalb sind bei der Analyse, Umgestaltung oder Neukonstruktion von Systemen i.d.R. umfangreiche experimentelle Studien erforderlich, um die Komplexität ausreichend zu verstehen und zu erfassen. Notwendige experimentelle Eingriffe können nur in den seltensten Fällen direkt am zu untersuchenden Realsystem durchgeführt werden, da durch derartige Eingriffe das System gestört werden würde und u.U. ungewollte Nebenwirkungen auftreten können. Bei Neukonstruktionen sind Studien am System aufgrund des Nichtexistierens unmöglich und können nur an Systemsurrogaten durchgeführt werden. Das zu untersuchende System muss also in ein experimentell manipulierbares Modell überführt werden.

Modelle sind materielle oder immaterielle Abbildungen natürlicher oder künstlicher Originale. Sie stellen Systeme in einer Art und Weise dar, dass experimentelle Studien, Analysen und Manipulationen der abgebildeten Systemstrukturen und Zustände möglich sind. Modelle sind ferner Simplifizierungen der Realsysteme, die einem bestimmten Zweck dienen.<sup>33</sup> Simplifizierungen entstehen zum einen durch Abgrenzungen des betrachteten Systems zur Umwelt. Dabei werden offene Systeme im Modell durch geschlossene Systeme ersetzt. Andererseits werden bei der Erstellung von Modellen aufgrund der Komplexität nicht alle Attribute erfasst. Vereinfachte Annahmen über die Anzahl der Variablen in Interaktions- und Kombinationsbeziehungen sowie Umwandlungsvorschriften stellen einen weiteren Abstraktionsgrad dar [Stac73 S.131ff; Stei91 S.13ff; Schü98 S.45; Niem77 S.57f].

Modelle können nach ihrem Abbildungsmedium (vgl. Abb. 6), nach den abgebildeten Zustandsvariablen und nach ihrem Verwendungszweck klassifiziert werden.

---

<sup>33</sup> Vgl. dazu auch Kapitel 3.1.

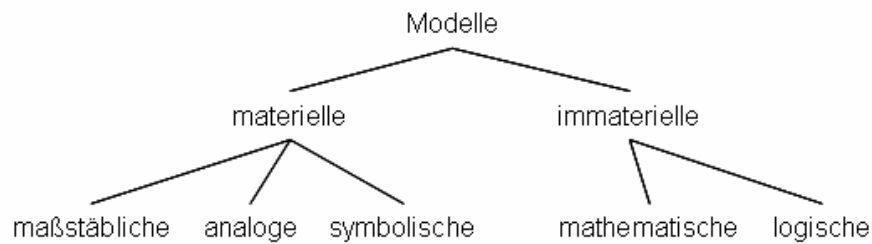


Abbildung 6: Klassifizierung von Modellen nach dem Abbildungsmedium<sup>34</sup>

Bei materiellen Modellen sind die Elemente durch materielle Zustände und Strukturen verkörpert. Materielle Modelle können in maßstäbliche, analoge oder symbolische Modelle unterteilt werden. Systeme, die in verkleinertem oder vergrößertem Maßstab abgebildet werden, nennt man Maßstabsmodelle. Entspricht die Modellstruktur der Systemstruktur und wird das betrachtete System im Modell mit anderen Stoffen abgebildet als das Original, spricht man von einem Analogmodell.<sup>35</sup> Eine Sonderform des materiellen Modells stellen symbolische Modelle dar. Deren materielle Strukturen und Zustände haben mit dem abgebildeten System keine physikalische Ähnlichkeit. Ein Beispiel hierfür ist das Schachspiel, welches zwei konkurrierende, sich „bekämpfende“ Parteien abbildet.

Zur Gruppe der immateriellen Modelle<sup>36</sup> gehören mathematische sowie logische Modelle. Mathematische Modelle verwenden zur Abbildung von Systemen Funktionen, Gleichungen, Variablen und Konstanten.<sup>37</sup> Logische Modelle stellen das wechselseitige Wirken der Attribute des betrachteten Systems durch logische Operatoren und Ablaufregeln<sup>38</sup> dar. Ein Beispiel für logische Modelle sind Flussdiagramme.

<sup>34</sup> Abbildung nach [Niem77 S.58]; vgl. auch [Peer98 S.61 u. Gord S.17ff].

<sup>35</sup> Z.B. Abbildung des menschlichen Kreislaufs mittels mechanisch-hydraulischer Systeme.

<sup>36</sup> Auch formale Modelle genannt.

<sup>37</sup> Z.B. Abbildung von Schwingungsverhalten mittel Differentialgleichungen.

<sup>38</sup> Sog. Algorithmen.



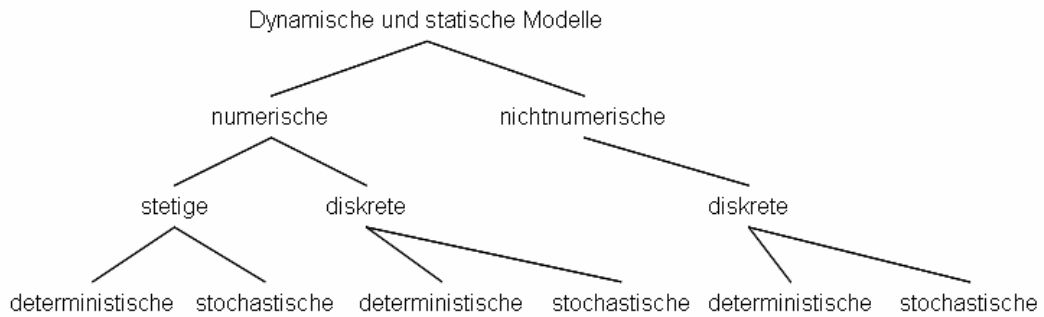


Abbildung 7: Klassifizierung von Modellen nach den Zustandvariablen

Klassifiziert man Modelle nach der Art der Zustandsvariablen, spricht man von dynamischen bzw. statischen Modellen (vgl. Abb. 7). Beide Arten lassen sich in numerische und nicht-numerische Modelle unterteilen. Numerische Modelle gliedern sich in stetige und diskrete Modelle, nicht-numerische sind stets diskrete Modelle. Ferner lassen sich Modelle in deterministische bzw. stochastische Modelle unterteilen. Dynamische Zustandsvariablen werden von der unabhängigen Variablen „Zeit“ bestimmt und bilden demnach Attributinteraktionen ab. Statische Variablen hingegen bilden Attributkombinationen ab und sind nicht abhängig von der Variablen „Zeit“. In einem diskreten Modell ändern sich die Zustandsvariablen unstetig<sup>39</sup> durch das Eintreten bestimmter Ereignisse. In einem stetigen Modell ändern sich die Zustandsvariablen kontinuierlich<sup>40</sup> im Zeitablauf. Gibt es Variablen, die mit anderen Variablen funktional verknüpft sind, heißen sie deterministisch. Stochastische Variablen werden hingegen durch einen Zufallsprozess bestimmt. [Niem77 S.58ff; Stei91 S.16; Kral96 S.8ff, KFG99 S.24f; Zwic81 S.18ff; Miel02 S.14; BCNN01 S.12; Gord72 S.14; Salz07 S.15]

Abbildung 8 zeigt die Einteilung von Modellen nach ihrem Verwendungszweck.

<sup>39</sup> Z.B. Anzahl von Patienten, die im Empfangsbereich warten.

<sup>40</sup> Z.B. Temperaturanstieg.

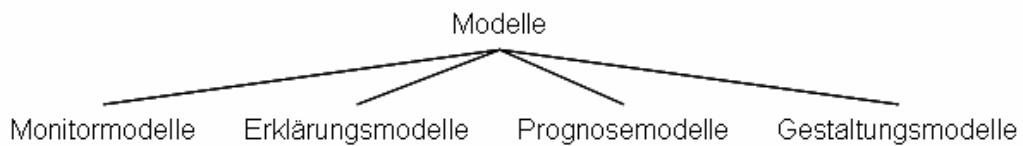


Abbildung 8: Klassifizierung von Modellen nach ihrem Verwendungszweck<sup>41</sup>

Monitormodelle erfassen fortwährend Systemzustände und bilden diese ab. Erklärungsmodelle erfassen Systemstrukturen und bilden diese ab. Zur Vorhersage zukünftiger Systemzustände bei sich verändernden unabhängigen Variablen dienen Prognosemodelle. Gestaltungsmodelle dienen der Ermittlung geeigneter Systemstrukturen. In der Praxis werden meist mehrere dieser Modellarten kombiniert. So schließen Prognosemodelle i.d.R. Erklärungsmodelle ein. Modelle, die im realen Zeitablauf eingesetzt werden, schließen immer Monitormodelle ein [Niem77 S.61].

### 2.1.3 Steuerung und Regelung

Zentrale Themen der kybernetischen Systemtheorie sind Steuerung und Regelung. Unter Steuerung versteht man allgemein die Zustandsänderung eines Systems durch ein anderes System. Im engeren Sinn liegt Steuerung vor, wenn durch intelligente Beeinflussung eines Systems dessen Zustand zielgerichtet verändert wird oder das System nach einer unerwünschten Störung stabilisiert werden soll. Verändernde Steuerung besteht darin, den Wert einer Zustandsvariablen in eine vorbestimmte Richtung zu beeinflussen. Abbildung 9 verdeutlicht das Prinzip. System 1 greift zum Zeitpunkt  $t_0$  die Zustandsvariable  $x_i(t_0)$  des Systems 2 ab. Aus der Abweichung zwischen dem gewünschten Zustand  $w$  (Führungsgröße) und dem aktuellen Zustand  $x$  wird die Stellgröße  $y_i(t)$  ermittelt und in das zu steuernde System eingeführt [Niem89 S.27].

---

<sup>41</sup> Abbildung nach [Niem77 S.61].

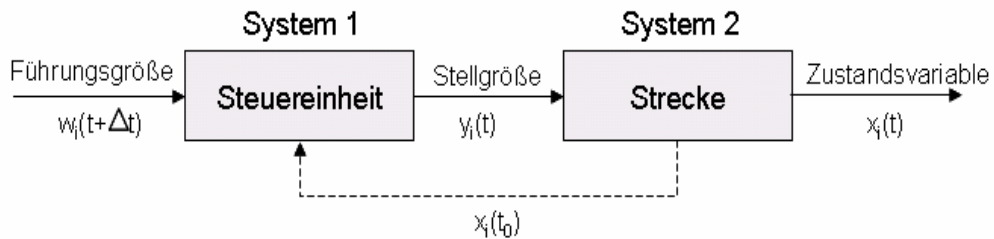


Abbildung 9: Schematische Darstellung einer verändernden Steuerung

Eine stabilisierende Steuerung erfasst Störungen, die auf ein System einwirken, und versucht diese durch entsprechende Maßnahmen zu kompensieren (vgl. Abb. 10).

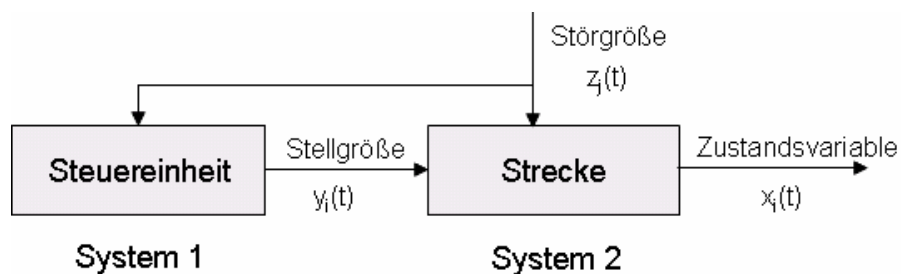


Abbildung 10: Schematische Darstellung einer stabilisierenden Steuerung

Die Störgröße  $z_j$  wird im Zeitpunkt  $t$  erfasst und an die Steuereinheit weitergegeben. Die Steuereinheit berechnet auf Basis der zu erwartenden Veränderung am System 2 eine Stellgröße  $y_i$  zur Kompensation der Störung und führt diese in das System 2 ein [Niem89 S.28]. Die Kenntnis über die Wirkung der Stellvariablen auf die zu steuernde Zustandsvariable ist für beide Arten der Steuerung von entscheidender Bedeutung.

Regelung ist eine sich wiederholende Steuerung mit permanenter Überwachung des Zustandes des betreffenden Systems. Hier kommt es also zu einer laufenden Überprüfung des Erfolges der Regelung und gegebenenfalls zu weiteren Eingriffen in das zu regelnde System. Wie bei der Steuerung wird auch hier zwischen stabilisierender und verändernder Regelung unterscheiden.

Hinsichtlich des Zeitverhaltens einer Strecke werden Strecken mit Verzögerung, Strecken mit Totzeit sowie integrale Strecken unterschieden [Niem89 S.33ff].

Führt der Input einer Stellgröße einer Steuereinheit A, die auf eine zu beeinflussende Zustandsvariable  $x$  trifft, nur Schritt für Schritt - also in Fraktionen - zu einer Zustandsänderung, so spricht man von einer Strecke mit Verzögerung (vgl. Abb. 11). Eine Strecke kann bis zu  $n=\infty$  Verzögerungsstufen haben. Bei  $n=0$  würde eine sofortige, vollständig durchschlagende Wirkung beim Auftreffen der Stellgröße auf die Zustandsvariable eintreten. Formal lässt sich diese Form der Strecke mit folgender Differentialgleichung darstellen:

$$\frac{dx}{dt} = a \cdot z_t - b \cdot x_t$$

mit

$a$  = Konstante zur Darstellung der Wirkung des Inputs  $z$  je infinitesimaler Zeiteinheit  $dt$

$b$  = Konstante zur Darstellung der Gegenwirkung je infinitesimaler Zeiteinheit  $dt$

$x_t$  = Zustand  $x$  zum Zeitpunkt  $t$

$z_t$  = Stellgröße (Input)  $z$  zum Zeitpunkt  $t$

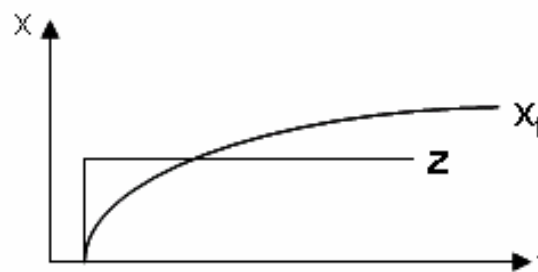


Abbildung 11: Reaktion einer Strecke mit Verzögerung

Zur Verdeutlichung zwei Beispiele: (1) Ein Patient bekommt eine Infusion mit Kochsalzlösung. Nach Öffnen des Beuterverschlusses tritt Flüssigkeit „schlagartig“ aus. Die Flüssigkeit muss jedoch eine Trichteröffnung passieren und tritt damit nur dosiert in die Blutbahn des Patienten ein. (2) Nach Spritzen von Kontrastmittel vor einer Kernspintomographie verteilt sich das Kontrastmittel nach und nach im Körper des Patienten.

Bei Strecken mit Totzeit wirkt ein Input  $z$  erst nach einer Zeitdauer  $T$  auf die betreffende Zustandsvariable  $x$  (vgl. Abb. 12). Es gilt:  $b \cdot x_{(t+T)} = a \cdot z_t$ .

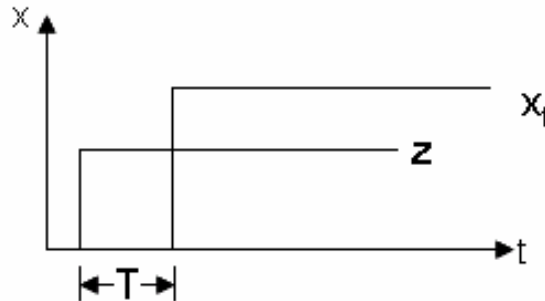


Abbildung 12: Reaktion einer Strecke mit Totzeit T

Auch hierzu zwei Beispiele: (1) Nach abgeschlossener Behandlung des Patienten und erfolgter Rechnungsstellung führt erst der Zahlungseingang auf dem Bankkonto – i.d.R. nach Verstreichen des Zahlungsziels und eventuellen Transaktionslaufzeiten zwischen den Bankinstituten – zu einer Liquiditätserhöhung. (2) Die Ergebnisse einer radiologischen Untersuchung erreichen erst nach Ende der Postlaufzeit den Hausarzt des Patienten.

Bei integralen Strecken gibt es keine Gegenwirkungen bzw. Rückkopplungen (vgl. Abb. 13). Die Wirkung des Inputs  $z$  bleibt also unvermindert. Dies hat zur Folge, dass sich bei integralen Strecken die Wirkungen des Inputs kumulieren. Es gilt:

$$\frac{dx}{dt} = a \cdot z_t$$

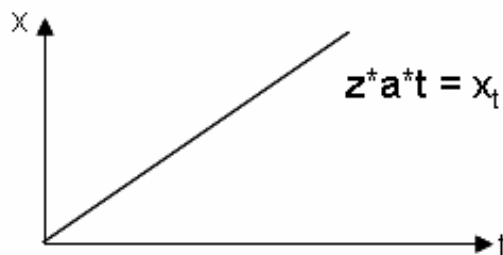


Abbildung 13: Reaktion einer integralen Strecke mit Sprunginput

Die beiden folgenden Beispiele verdeutlichen die Wirkungsweise: (1)  $z$  Dialysegeräte reinigen in einer Zeiteinheit  $t$  jeweils die Menge  $a$  an Blut. Nach  $t$  Zeiteinheiten ist insgesamt die Menge  $z \cdot a \cdot t$  an Blut gereinigt worden.

In der Radiologie eines Krankenhauses werden zum Zeitpunkt  $t=0$   $z$  Magnetresonanztomographen gleicher Baureihe<sup>42</sup> in Betrieb genommen. Diese erstellen jeweils  $a$  Bilder je Zeiteinheit  $t$ . Nach  $t$  Zeiteinheiten ist die Gesamtmenge an erstellten Bildern  $z \cdot a \cdot t$ .

## 2.2 Simulation und deren Anwendungsgebiete

Simulation kann als Nachbildung des Verhaltens von Prozessen oder Systemen über die Zeit definiert werden [BCNN01 S.3; Gord72 S.26; Gülk04 S.27]. Mit Hilfe der Simulation können durch Modellexperimente Informationen über das dynamische Verhalten von Systemen gewonnen werden [LaKe04 S.1ff]. Dabei liefern die simulierten Systemzustände des vorangegangenen Zeitpunktes die Werte für die Bestimmung des gegenwärtigen Systemzustandes. Die Werte des gegenwärtigen Systemzustandes determinieren wiederum die zukünftigen Systemzustände [Niem73 S.37].

Simulation eignet sich insbesondere in den folgenden Fällen [BCNN01 S.3f; BLR04 S.1; Gord72 S.39]:

- (1) Experimente am Realsystem können nicht durchgeführt werden, da das System erst entwickelt wird.
- (2) Mathematische Modelle kommen aufgrund der Komplexität des betrachteten Systems nicht in Frage.
- (3) Experimente am Realsystem sind zu teuer und zeitintensiv<sup>43</sup>
- (4) Experimente am Realsystem bergen ein zu hohes Risiko. Dies gilt u.a. für Krankenhäuser, Pflege- und Reha-Einrichtungen, bei denen durch Experimente am Realsystem die Gesundheit des Menschen (=Patient) gefährdet werden kann.

---

<sup>42</sup> Und damit gleicher Geschwindigkeit.

<sup>43</sup> Z.B. in der Raumfahrt.

Simulation kann in vier Anwendungsgebiete unterteilt werden. Diese sind: Systemanalyse, Systementwurf, Systemoptimierung und Systemschulung [Niem73 S. 45ff; BCNN01 S. 4f; Gord72 S.27f; BLR04 S.14].

- (1) Eines der Ziele der **Systemanalyse** ist es, das Verhalten einzelner Komponenten des Systems sowie deren Interaktionsbeziehungen zu verstehen. Ferner gilt es, durch Systemanalyse die Wirkungsweise des betrachteten Systems unter sich ändernden Ablaufbedingungen zu verstehen. Im Rahmen der Systemanalyse wird zuerst ein Modell entwickelt, welches das zu untersuchende Realsystem repräsentiert. Die korrekte Abbildung sowie die korrekte Wirkungsweise des Modells werden anschließend im Rahmen der Modellverifikation und -validierung untersucht. Unter Berücksichtigung der im jeweiligen Modell bestehenden Modellgenauigkeit können die gewünschten Analysen durchgeführt werden. Ein Beispiel für Systemanalyse ist die Simulation des Verhaltens einer klinischen Abteilung bei unterschiedlichen Patientenströmen.<sup>44</sup>
- (2) Der **Systementwurf** dient der Entwicklung eines funktionsfähigen Systems, welches bestimmte Voraussetzungen und Ziele erfüllen soll. Das zu entwickelnde System wird in einem Modell abgebildet. Im Modell wird das Verhalten der einzelnen Komponenten und deren Wechselwirkungen unter verschiedenen Ablaufbedingungen getestet, bis die geforderten Ergebnisse erreicht werden. Anschließend kann das mit dem Simulationsmodell dargestellte System entwickelt bzw. gebaut werden. Die Simulation dient in diesem Fall beispielsweise dem Bau einer neuen Klinik bzw. eines klinischen Teilbereiches oder kann zur Wegeoptimierung und zur Kapazitätsplanung eingesetzt werden.
- (3) Systeme können mit Hilfe der Simulation optimiert werden. **Systemoptimierung** bezieht sich hier auf die Aufbauorganisation bzw. die im System ablaufenden Prozesse. Bei der Systemoptimierung wird der Zielwert bzw. Zielkorridor für die Zustandsvariablen bzw. für Kombinationen aus Zustandsvariablen festgelegt. Um die Optimierung der Zielgrößen bzw. Zielfunktionen zu erreichen, wird das Modell entsprechend variiert.

---

<sup>44</sup> Z.B. Unterschiedliche Patientenzahlen oder/und unterschiedliche Ankunftszeiten der Patienten.

Variationen können bezüglich der Anordnung der Systemkomponenten und deren Eigenschaften, wie auch bezüglich der Wirkzusammenhänge erfolgen. Bei der Optimierung der Durchlaufzeiten von Patienten können unter Berücksichtigung von Kosten-Nutzen-Aspekten z.B. der Einsatz verbesserter Technologien, die Variation der Anzahl eingesetzter Mitarbeiter sowie deren Auslastung, und die Variation der zeitlichen Abfolge der einzelnen Behandlungsschritte simuliert werden.

- (4) Im Rahmen einer **Systemschulung** können Personen mit Hilfe der Simulation geschult und trainiert werden. Der klare Vorteil der Schulung an Modellen liegt darin, dass Personen nicht am Realsystem üben müssen. Dadurch werden Störungen am Realsystem ferngehalten. Beispiele sind Schulungen, die mit Hilfe von Fahrzeug- und Flugzeugsimulatoren durchgeführt werden, sowie Unternehmensplanspiele.

Die oben beschriebenen Anwendungsgebiete lassen sich in der Praxis nicht immer strikt voneinander trennen. In Kapitel 7 der vorliegenden Arbeit werden die Prozesse einer Tagesklinik mit einer bestimmten Zielsetzung optimiert (=Systemoptimierung). Für die Optimierung ist es jedoch notwendig, über eine fundierte Systemanalyse zuerst das Verhalten der Tagesklinik unter sich ändernden Ablaufbedingungen kennen zu lernen (=Systemanalyse).

Simulationen werden nach stetiger und diskreter Systemsimulation unterschieden [Gord72 S.14 u. S.39ff u. S.128ff; BCNN01 S.12; Niem73 S.234ff]. Man spricht von stetiger Simulation, wenn sich die Zustandsvariablen kontinuierlich im Zeitablauf ändern (vgl. Abb. 14). Im Falle einer diskreten Simulation ändern sich die Zustandsvariablen des Systems unstetig durch das Eintreten bestimmter Ereignisse (vgl. Abb. 15). Sind Systeme nicht ausschließlich diskret oder stetig, erfolgt die Einteilung nach der Art der Änderungen der Zustandsvariablen, die im System überwiegen.



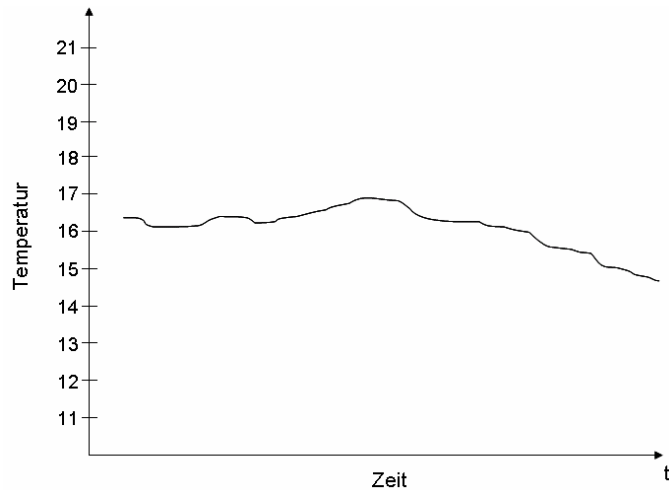


Abbildung 14: Zustandsvariable eines stetigen Systems

Die Zustandsvariable „Temperatur“ in Abbildung 14 ändert sich stetig im Zeitablauf.

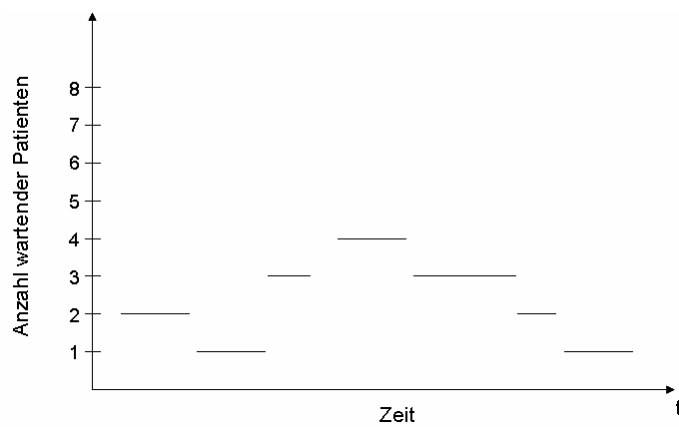


Abbildung 15: Zustandsvariable eines diskreten Systems

Die Zustandsvariable „Anzahl wartender Patienten“ in Abbildung 15 ändert sich, wenn ein neuer Patient eintrifft (=Ereignis) oder ein wartender Patient seine Behandlung beginnt (=Ereignis).

In der vorliegenden Arbeit wird in Kapitel 7 ein diskretes System - am Beispiel einer Tagesklinik - in einem Simulationsmodell abgebildet. Deshalb wird an

dieser Stelle auf das Grundkonzept der diskreten Systemsimulation näher eingegangen.

Ereignisse bewirken zum Zeitpunkt  $t$  Änderungen im System. Dabei können Ereignisse Änderungen der Werte von Attributen der Systemelemente bewirken, sie können Elemente entstehen lassen oder „vernichten“ und sie können eine Aktivität beginnen oder beenden [BCNN01 S.64; Gord72 S.8]. Um Änderungen durch Eintritt von Ereignissen über die Zeit mitverfolgen zu können, verfügen Simulationssysteme über eine „interne Uhr“, die die Simulationszeit fortschreibt. Die simulierte Zeit, und die Zeit für die Durchführung der Simulation, sind dabei nicht unbedingt identisch. Ein entscheidender Faktor für die Dauer der Durchführung der Simulation - also der benötigten Rechnerzeit - ist neben der Komplexität und Detaillierung des Modells die Anzahl der auftretenden Ereignisse. Je mehr Ereignisse in einem bestimmten Zeitintervall auftreten, desto mehr Rechnerzeit wird benötigt. Die Rechenzeit wird zudem durch die Rechnerleistung beeinflusst.

Um zu bestimmen, zu welchen Simulationszeitpunkten Ereignisse auftreten, können grundsätzlich zwei Methoden angewandt werden [Gord72 S.129; BCNN01 S.65ff]: Bei der ereignisorientierten Methode wird die Zeit auf den Ereigniszeitpunkt vorgerückt, zu dem das nächste Ereignis eintritt. Bei der intervallorientierten Methode wird die Zeit um einheitliche, kleine Zeitintervalle weitergestellt und in jedem Zeitpunkt geprüft, ob ein Ereignis auftritt. Die Herausforderung bei der intervallorientierten Methode ist es, die Intervalle so zu wählen, dass alle möglichen, eintretenden Ereignisse berücksichtigt werden können. Die benötigte Rechenzeit kann bei dieser Methode zu einer langen Simulationsdauer führen. Die diskrete Systemsimulation wird daher i.d.R. mit der ereignisorientierten Methode durchgeführt. Dabei werden in einer Ereignisliste alle terminierten zukünftigen Ereignisse fortgeschrieben. Der Eintrittszeitpunkt des jeweils nächsten Ereignisses bestimmt dabei die fortgeschriebene Zeit. Bei jedem Fortschreiten der Zeit werden dabei der Systemzustand sowie die Ereignisliste entsprechend aktualisiert. Wenn in der vorliegenden Arbeit im Weiteren von diskreter Simulation gesprochen wird, bezieht sich dies ausschließlich auf die event-basierte Simulation.

Systeme enthalten oftmals Aktivitäten, deren Verhalten zufällig ist bzw. Aktivitäten, über deren Verhalten keine vollständigen Kenntnisse vorhanden sind. In diesen Fällen ist die genaue Reihenfolge der eintretenden Ereignisse unbekannt bzw. liegt nicht vollständig vor. Variablen, die die Ausgangsgrößen solcher Aktivitäten beschreiben, werden stochastische Variablen bzw. Zufallsvariablen genannt [Niem73 S.55; Gord72 S.92; BCNN01 S.165]. Für diese Variablen gilt es, den Wertebereich zu bestimmen, innerhalb dessen sie variieren können, sowie die Wahrscheinlichkeit, mit der sie diese Werte annehmen. Beispiele für derartige Variablen sind z.B. die Behandlungsdauer für ein bestimmtes Krankheitsbild, die Anzahl und Ankunftszeiten von Patienten in der Zentralaufnahme eines Krankenhauses sowie krankheitsbedingte Ausfälle des Personals oder Ausfälle medizinischer Geräten.

Zur Berechnung möglicher Ausprägungen von Zufallsvariablen können je nach Aufgabenstellung bestimmte Wahrscheinlichkeitsverteilungen eingesetzt werden. Zu den diskreten Verteilungen gehören hier z.B. die Binomialverteilung und die Poissonverteilung [BaBa89 S.99f u. S. 103f]. Zu den stetigen Verteilungen, die in Simulationen zur Anwendung kommen, gehören z.B. die Gleichverteilung, die Exponentialverteilung und die Normalverteilung [BaBa89 S.106ff].

Bisher wurde über die Anwendung von Wahrscheinlichkeitsverteilungen zur Beschreibung von stochastischen Variablen gesprochen. Variablen ändern im Zeitablauf i.d.R. aber auch die Werte, die sie annehmen können. Deshalb müssen im Rahmen der Durchführung einer Simulation Zahlenreihen generiert werden, bei denen die aufeinanderfolgenden Werte zufällig sind und die Verteilung berücksichtigen, welche die jeweils betrachtete Zufallsvariable hat. Man spricht hier auch von Pseudo-Zufallszahlen.

Ein einfaches Beispiel zur Generierung solcher Zahlenreihen ist eine diskrete Wahrscheinlichkeitsverteilung mit  $n$  verschiedenen Zahlen, von denen jede mit gleicher Wahrscheinlichkeit auftreten kann (Gleichverteilung). Ein Roulette oder ein Würfel könnte eine derartige Folge von Zufallszahlen generieren. Für die Analyse von Systemen mit derartigen Variablen werden Monte Carlo Simulationen eingesetzt. Der Begriff der Monte Carlo Simulation entstand aufgrund der Verbindung zwischen Monte Carlo und den dort ansässigen Spielcasinos, in denen u.a. Würfelspiele und Roulette angeboten wird, und diese

Glücksspiele ebenfalls eine Art von Zufallszahlenreihe erzeugen [Gord72 S.97]. Beim Würfeln weiß man, dass entweder die Zahl 1, 2, 3, 4, 5 oder 6 kommt. Man weiß aber nicht, bei welchem Wurf welche Zahl auftritt. Ähnlich verhält es sich mit stochastischen Variablen. Man weiß zwar i.d.R., in welchem Wertebereich sich Zufallsvariablen bewegen können, man weiß aber nicht, zu welchem Zeitpunkt bzw. bei welchem Ereignis sie welchen konkreten Wert annehmen werden.

### **3. Prozessmodellierung**

In Kapitel 3 wird der Modell- und Prozessbegriff definiert. Methoden zur Beschreibung von Prozessen werden dargestellt. Die entscheidenden Grundlagen für die Modellbildung werden aus der kybernetischen System- und Modelltheorie abgeleitet.

#### **3.1 Hauptmerkmale des Modellbegriffs**

Stachowiak ordnet dem Modellbegriff drei Hauptmerkmale zu: Das Abbildungsmerkmal, das Verkürzungsmerkmal und das pragmatische Merkmal [Stac73 S.131ff].

##### (1) Abbildungsmerkmal

„Modelle sind stets Modelle von etwas, nämlich Abbildungen, Repräsentationen natürlicher oder künstlicher Originale, die selbst wieder Modelle sein können“ [Stac73 S.131].

##### (2) Verkürzungsmerkmal

In der Regel werden und können in Modellen nie alle Attribute des Originalsystems erfasst werden. Lediglich solche finden Berücksichtigung, die den Entwicklern des entsprechenden Modells und/oder den Benutzern des Modells relevant erscheinen [Stac73 S.132].

##### (3) Pragmatisches Merkmal

Modelle erfüllen zum einen ihre Funktion für bestimmte Personen, die die Modelle nutzen. Zum anderen erfüllen Modelle ihre Ersetzungsfunktion für bestimmte Zwecke und Zielsetzungen [Stac73 S.132f].

Die Entscheidung darüber, welche Attribute relevant sind, hängt letztendlich von der konkreten Zielsetzung des jeweiligen Modells ab. In der Definition des

Verkürzungsmerkmal ist die implizite Annahme enthalten, dass der Modellentwickler bzw. der Nutzer des Modells weiß, welche Attribute wichtig und welche im Modell vernachlässigbar sind. Dies ist im Vorfeld jedoch nicht möglich und wird erst während der Modellentwicklung klar.

Aufgrund der Grenzen menschlichen Wissens und Denkvermögens muss der Mensch das betrachtete Realsystem von Neben- und Subsystemen abgrenzen. Das Modell stellt eine Simplifizierung des Realsystems dar. Art und Umfang der Abgrenzung werden dabei durch die konkrete Zielsetzung des Modells, die subjektive Wahrnehmung des Systembetrachters, seiner Perspektive, seines Hintergrundwissens, seiner Wertvorstellungen, seiner Interessenslage sowie seiner kognitiven Fähigkeiten bestimmt [Stein91 S.10; Niem73, S.3; Stac73 S.131 und S.208; GBB02 S.20ff; Salz06 S.13]. Es entsteht - bewusst oder unbewusst - ein sog. mentales Modell. Das mentale Modell ist das Bindeglied zwischen dem Realsystem (=Original) und dem am Ende des Modellbildungsprozesses entwickelten formalen bzw. physikalischen Modell (vgl. Abb. 16)<sup>45</sup>.

Das mentale Modell stellt ein implizites System dar, welches anschließend in ein explizites Modell<sup>46</sup> überführt wird. Bei diesem Abbildungsvorgang kann eine weitere Reduktion der Komplexität durch den Modellentwickler erfolgen. Ausschlaggebend bei der Entwicklung von Modellen ist immer der jeweilige Modellzweck. Der Wert des Modells ist in der konkreten Anwendung zu sehen.

---

<sup>45</sup> Dem expliziten Modell.

<sup>46</sup> Formales oder physikalisches Modell.

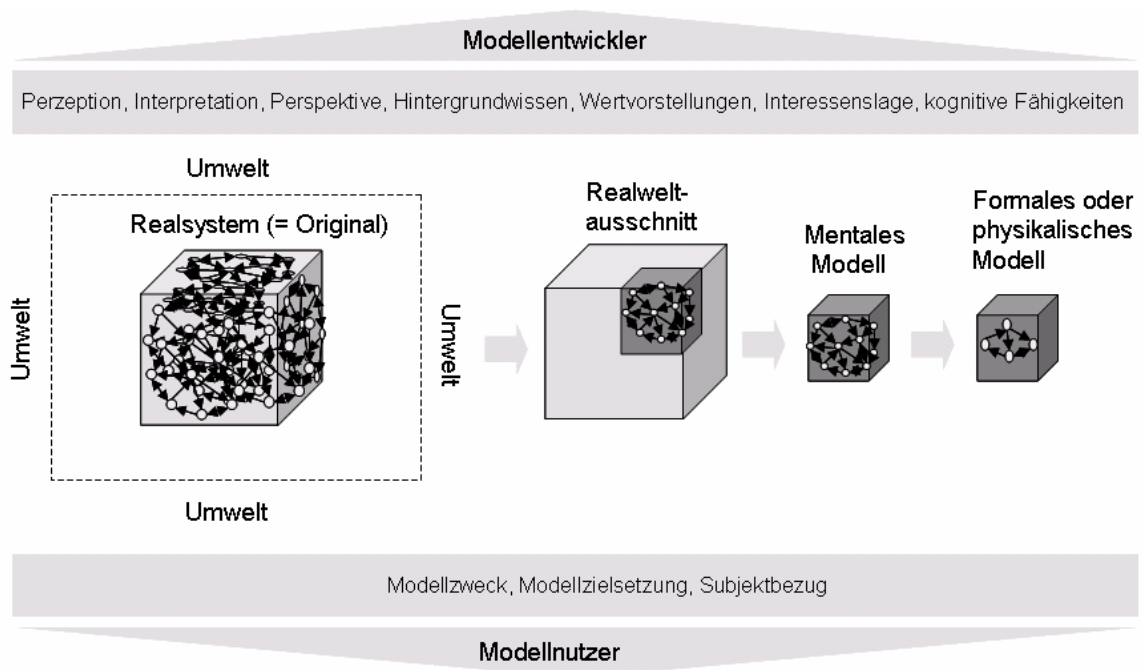


Abbildung 16: Schematische Darstellung des Modellbildungsprozesses

Nachdem die Mechanismen und Einflüsse, die bei der Modellentwicklung auftreten, beschrieben wurden, wird nun der Frage nachgegangen, ob und wie es gelingt, die Entwicklung eines Modells soweit als möglich transparent und objektiv zu machen. Das bedeutet letztendlich die (teilweise) Reduktion der Subjektivität bei der Modellbildung. Grundsätzlich gilt, dass vorab festgelegte Abbildungsregeln und Rahmenbedingungen für die Erstellung eines bestimmten Modells nicht nur den Prozess der Modellentwicklung transparenter machen. Auch das entwickelte Modell selbst wird für die Modellnutzer verständlicher und letztendlich auch objektiver. Folgende Grundsätze sollten bei der Modellentwicklung beachtet werden:

- (1) Ein Modell kann nur für einen bestimmten Zweck validiert werden. Deshalb muss das Modell der konkreten Zielsetzung des Modellnutzers gerecht werden. Nicht relevante Sachverhalte können eliminiert werden, sofern sie für den Modellzweck und der Modellzielsetzung keinen Beitrag leisten. Dies trägt zur Reduktion der Modellkomplexität bei. Struktur- und Verhaltenstreue des Modells gegenüber dem Original, die semantische Konsistenz und die Aktualität des Modells müssen gegeben sein.

- (2) Modelle sollen sowohl für den Modellentwickler als auch den Modellnutzer klar und verständlich sein. Dies wird erreicht durch Strukturiertheit, einheitliche Konventionen, gute Lesbarkeit und Übersichtlichkeit der Modelle. Dabei spielen grafische Darstellungen von Modellen eine bedeutende Rolle.
- (3) Modelle sollen vergleichbar sein. Vergleichbarkeit ist bei arbeitsteiliger Modellierung wie auch bei der Durchführung von Benchmarking-Studien entscheidend. Modellkonformität wird durch Einhaltung von Konventionen bezüglich verwendeter Modellkonstrukten und Bezeichnern erreicht.

## **3.2 Prozesse**

### **3.2.1 Begriffsdefinitionen**

Ein Prozess ist die Bearbeitung bzw. das Abarbeiten von Aufgaben. Aufgaben definieren Ziele mit den zur „Erreichung des Ziels notwendigen Angaben über Daten bzw. Objekte, Mittel, Lösungsvorschriften und weitere Randbedingungen“ [DIN96 S.15; vgl. auch Schei87 S.39 und Peer98 S.9f]. Prozesse bestehen aus einer zusammengehörenden Abfolge von Aktivitäten. Aktivitäten sind Teilschritte eines Prozesses. Sie stehen zueinander in einer Leistungsbeziehung sowie in einer zeitlichen Beziehung. Aktivitäten können ihrerseits in untergeordnete Aktivitäten zerlegt werden [Gai95 S.70f, Sche98 S.3, Hess96 S.20; RLH96 S.17; Peer98 S.10f]. Prozesse verbrauchen Zeit und benötigen zu ihrer Durchführung Ressourcen wie z.B. Personen, Kapital und Betriebsmittel, ferner immaterielle Elemente wie Daten und Informationen [Lang97 S.10].

Prozesse haben einen definierbaren Input und einen messbaren Output [Dave93 S.5; NFSH99]. Der Input eines Prozesses ist eine Teilmenge der am Prozess beteiligten Ressourcen. Der Output des betrachteten Prozesses entsteht durch Be- und Verarbeitung des Inputs. Aus unternehmerischer Sicht stellt der Output eine materielle (Sachleistung) oder immaterielle Leistung (Dienstleistung) dar [Sche98 S.13].



### 3.2.2 Standardisierte Beschreibung von Prozessen

Prozesse können einen hohen Grad an Komplexität erreichen. Vor dem Hintergrund einer effektiven und effizienten Modellbildung erscheint es deshalb sinnvoll, Teilprozessschritte soweit als möglich in eine Art Toolbox<sup>47</sup> zu integrieren und diese quasi als Standard zur Verfügung zu stellen. Dies bietet bei der Modellbildung im Rahmen der Referenzierung diverse Vorteile. Durch den Zugriff auf die Toolbox wird sichergestellt, dass wiederkehrende, mehrfach verwendete Prozessschritte in der gleichen Art und Weise in Modellen dargestellt werden und damit die Vergleichbarkeit von Modellen unterstützt wird.<sup>48</sup> Ferner wird sichergestellt, dass durch standardisierte Prozesskomponenten die zu einem übergeordneten Prozessschritt gehörenden Subprozesse vollständig im Modell abgebildet werden. Durch Referenzmodelle wird der Zeitaufwand bei der Modellierung von Prozessen erheblich reduziert. Gleichzeitig steigt die Qualität des Modells an.

Abbildung 17 zeigt den schematischen Aufbau der Toolbox. In einem Top-Down-Ansatz wird - ausgehend von einem Mastermodell, welches einen noch hohen Abstraktionsgrad aufweist und damit auch eine hohe Allgemeingültigkeit besitzt - ein Referenzmodell entwickelt, welches bereits einen deutlich geringeren Abstraktionsgrad besitzt.

---

<sup>47</sup> Als Toolbox wird ein „Werkzeugkasten“ bezeichnet, in dem alle benötigten Prozessbausteine (= „Werkzeuge“) hinterlegt sind.

<sup>48</sup> Z.B. durch Verwendung einheitlicher Bezeichner.

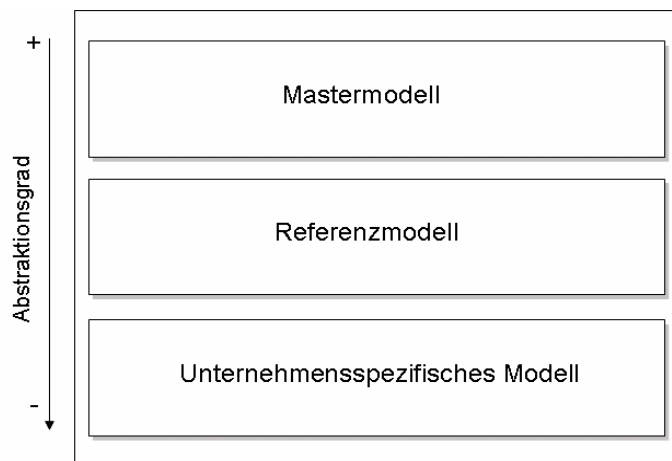


Abbildung 17: Aufbau der Toolbox (Prinzipdarstellung)

Der Übergang vom Master- zum Referenzmodell ist dabei fließend. Das Referenzmodell dient als Grundlage für die Entwicklung der unternehmensspezifischen Modelle [Schü98 S.313ff]. Letztere bilden konkrete reale Prozesse des jeweils betrachteten Unternehmens im Modell ab und haben somit auch den geringsten Abstraktionsgrad.

In der vorliegenden Arbeit werden Unternehmensprozesse in Kliniken in fünf Prozesskategorien<sup>49</sup> unterteilt. Dazu gehören übergeordnete Unternehmensprozesse<sup>50</sup>, das Customer & Partner Relationship Management, die klinischen Kernprozesse<sup>51</sup>, das Product and Service Lifecycle Management sowie diverse Supportprozesse (vgl. Abb. 18). Diese Hauptprozesse werden nachfolgend mit der Prozessebene 1 gleichgesetzt. Sie können anschließend weiter granularisiert werden. Abbildung 19 zeigt beispielhaft die zu den Hauptprozessen (=Prozessebene 1) gehörigen Subprozesse (=Prozessebene 2). Es ist unschwer zu erkennen, dass sowohl Prozessebene 1 als auch Prozessebene 2 generischer Natur sind und damit für eine Vielzahl von Unternehmen<sup>52</sup> gleichermaßen Gültigkeit besitzen. Diese beiden Prozessebenen sind folglich Teil des Mastermodells (vgl. Abb. 17).

Ab Prozessebene 3 wird der Fokus von nun an auf die klinischen Kernprozesse - auch als Patientenprozesse bezeichnet - gelegt, da diese für die vorliegende

<sup>49</sup> Im Weiteren auch als Hauptprozesse bezeichnet.

<sup>50</sup> Im Weiteren auch Managementprozesse genannt.

<sup>51</sup> Im Weiteren auch als Patientenprozesse bezeichnet.

<sup>52</sup> Der Begriff „Unternehmen“ kann hier gleichgesetzt werden mit „Krankenhaus“.

Arbeit relevant sind. Abbildung 20 zeigt am Beispiel der Patientenaufnahme, wie der zugehörige übergeordnete Prozess weiter verfeinert wird. Die Aufnahme des Patienten in der Klinik ist Teil des klinischen Kernprozesses. Sie gliedert sich in die Kategorien Notaufnahme, stationäre Aufnahme und ambulante Aufnahme. Zu jeder Kategorie sind die zugehörigen Subprozesse auf Level 3 erfasst.

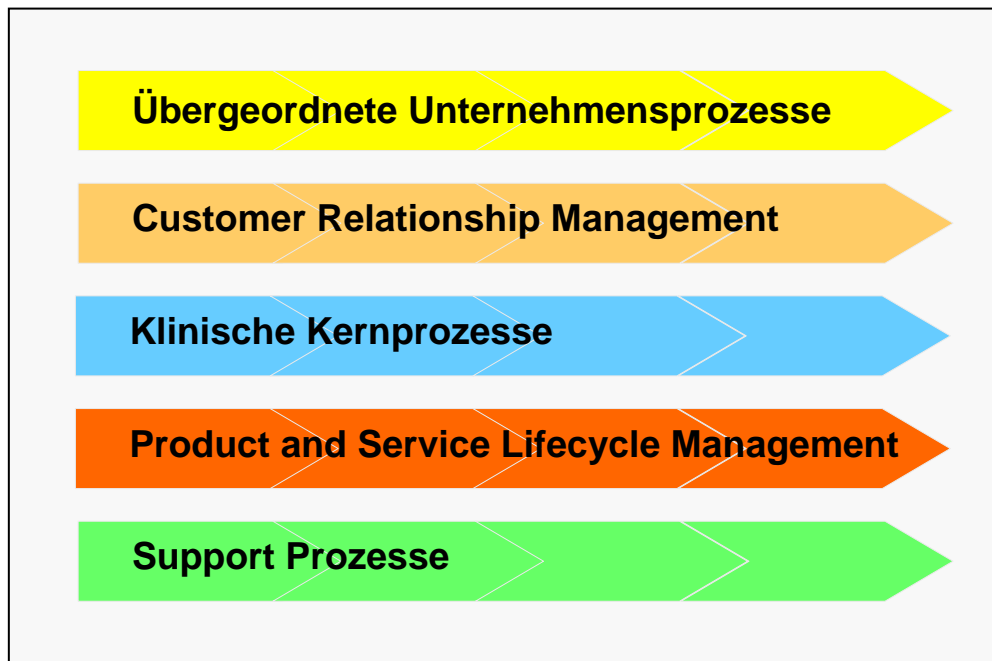


Abbildung 18: Kategorisierung von Unternehmensprozessen auf Hauptprozessebene<sup>53</sup>

Abbildung 21 zeigt am Beispiel der Notaufnahme von Patienten die Prozessebene 4, ferner die übergeordneten Prozessebenen 2 (Patientenaufnahme) und 3 (Notaufnahme). Hier zeigt sich, dass die Detaillierung der Beschreibung einzelner Tätigkeiten gegenüber den hierarchisch übergeordneten Prozessschritten deutlich zugenommen hat. Theoretisch könnte nun noch Prozessebene 5 beschrieben werden. Die praktische Erfahrung jedoch zeigt, dass dies nur in sehr eingeschränkten Fällen sinnvoll ist. Für die spätere Modellierung der Prozesse in Kapitel 5 wird dies keine zusätzlichen Erkenntnisse bringen, da bereits alle Facetten, die bei der Prozessmodellierung in Betracht gezogen werden müssen, in den Prozessebenen 1 bis 4 gezeigt werden. Für die Entwicklung von Modellbausteinen ist eine weitergehende Granularität ebenfalls nicht zielführend, da Subprozesse auf Ebene 5 derart spezifisch werden, dass

<sup>53</sup> Vgl. hierzu auch [FeSe04 A.18f u. S.26f].

allgemein gültige Modellkomponenten praktisch nicht mehr zu entwickeln sind.<sup>54</sup> Auf die Darstellung weiterer Prozessebenen wird deshalb in der vorliegenden Arbeit verzichtet.

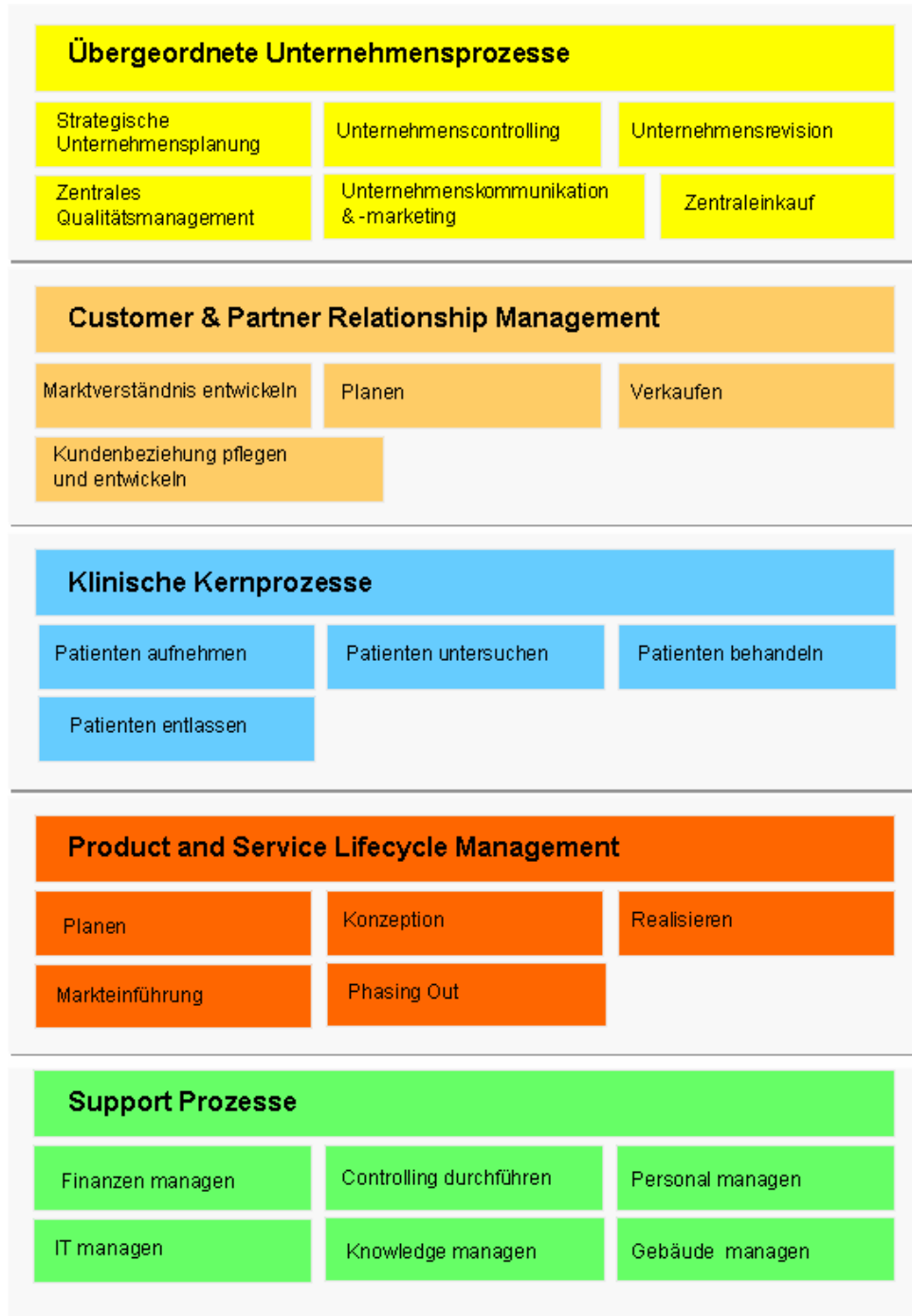


Abbildung 19: Hauptprozesse und zugehörige Subprozesse des nächsten Prozesslevels<sup>55</sup>

<sup>54</sup> Unternehmen verzichten bei der Optimierung von Kernprozessen i.d.R. ebenfalls auf einen zu hohen Detaillierungsgrad, da dem dafür notwendigen Mitteleinsatz i.d.R. kein ausreichender Zusatznutzen gegenübersteht.

<sup>55</sup> Vgl. hierzu auch [FeSe04 S.21].

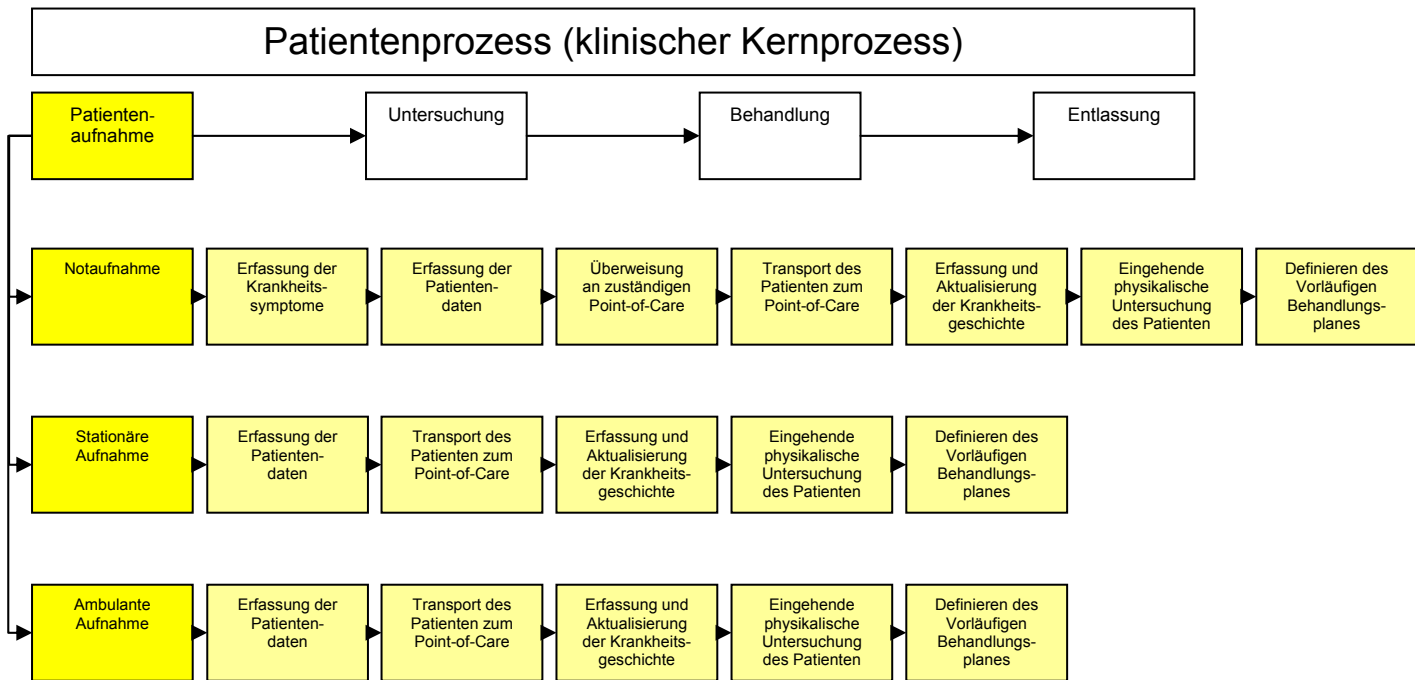


Abbildung 20: Prozesse bis Prozesslevel 3 am Beispiel der Patientenaufnahme

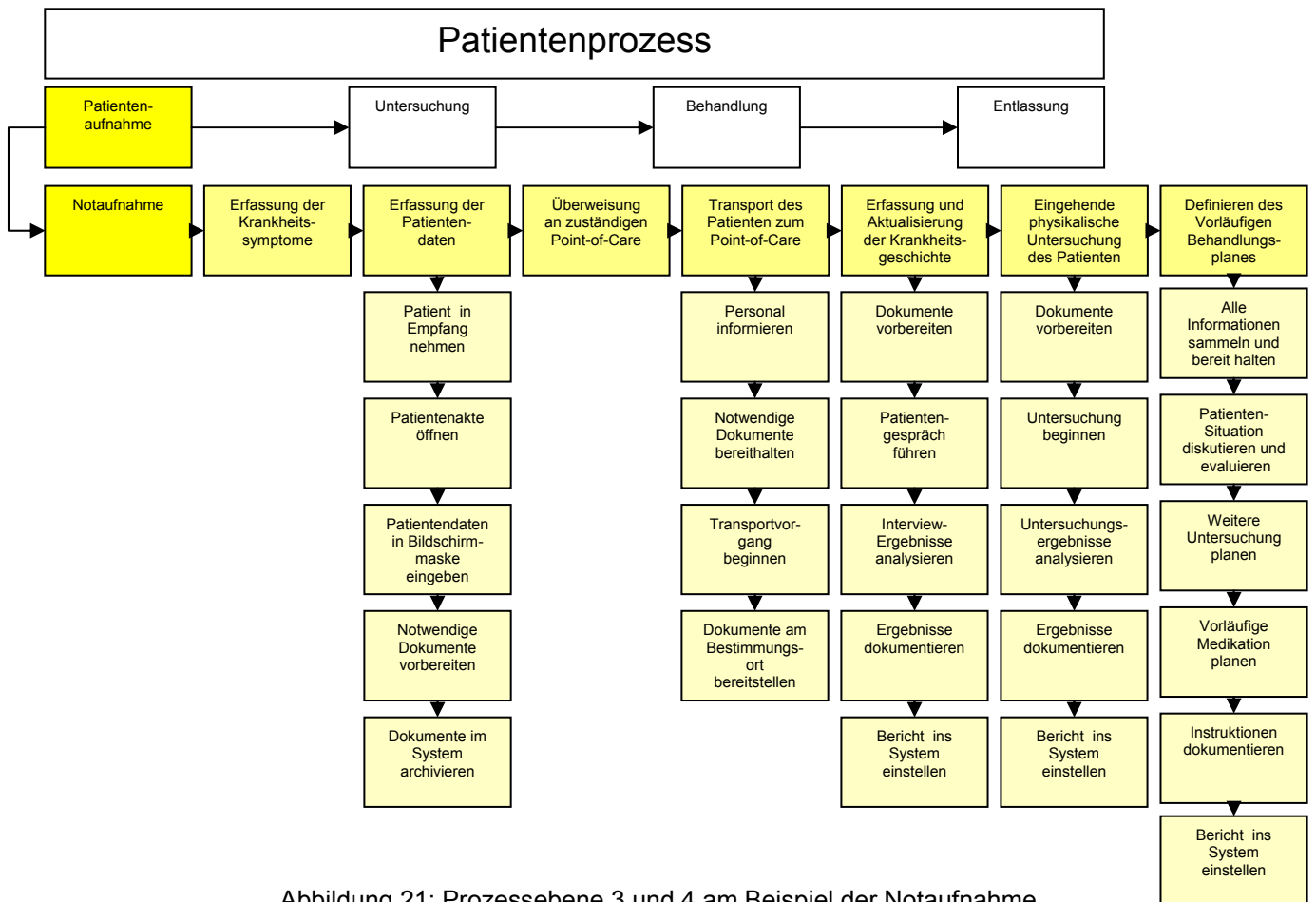


Abbildung 21: Prozessebene 3 und 4 am Beispiel der Notaufnahme

Wird nun die eben beschriebene Vorgehensweise der Prozessdarstellung auf die weiter oben dargestellte schematische Abbildung der Toolbox übertragen, so ergibt sich die Prinzipdarstellung in Abbildung 22.

Nachdem die Begriffe Prozesskategorien und Hierarchisierung von Prozessen erläutert wurden, wird nun auf die inhaltliche Beschreibung von Prozessen eingegangen. Dazu wird auf die Definitionen in Kapitel 3.2.1 zurückgegriffen.

Prozesse verbrauchen Zeit und Ressourcen. Sie stehen i.d.R. mit anderen Prozessen bzw. Subprozessen in Interaktion, erhalten Input und produzieren Output nach bestimmten Vorschriften. Um die für die Entwicklung von Modellen notwendigen inhaltlichen Beschreibungen und Interaktionen der einzelnen Prozessschritte strukturiert zu erfassen, werden i.d.R. standardisierte Templates<sup>56</sup> benutzt. Abbildung 23 zeigt das Beispiel eines Template zur Erfassung relevanter Informationen.

---

<sup>56</sup> Template = Formular zur Erfassung von Daten/Informationen.

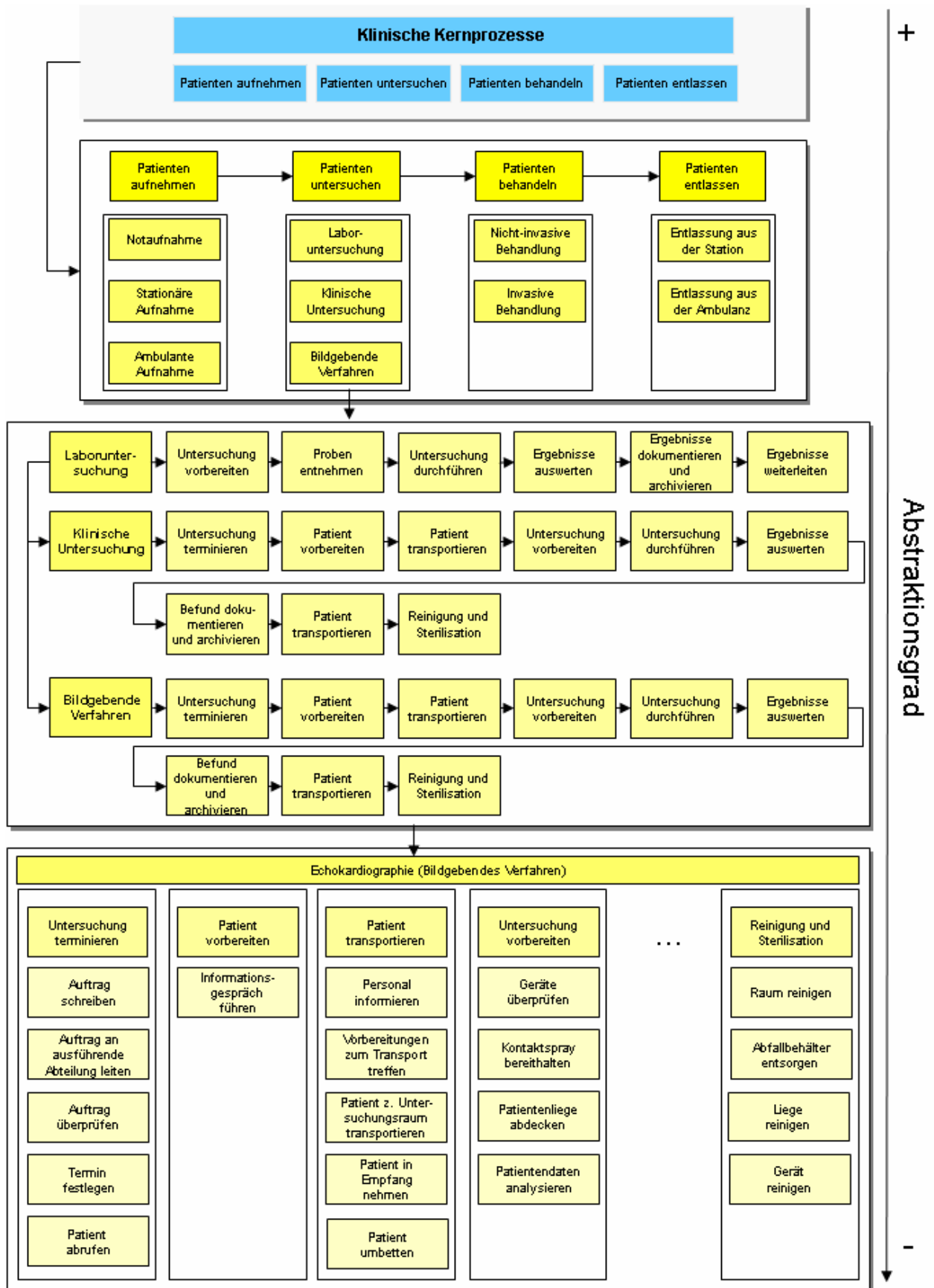


Abbildung 22: Aufbau der Toolbox mit Prozesskomponenten unterschiedlichen Abstraktionsgrades (Prinzipdarstellung)

<b>Name des Prozessschritts:</b> <Name des Prozessschrittes>	
<b>Beschreibung:</b> <Beschreibung>	
<b>Input:</b> ▪ <Beschreibung des Inputs>	<b>Benötigte Zeit (⌘)</b> <Zeitverbrauch>
<b>Output:</b> ▪ <Beschreibung des Outputs>	
<b>Ressourcen</b>	
<b>Personal:</b> ▪ <Name Personal>	<b>Material:</b> ▪ <Name Material>
	<b>Equipment:</b> ▪ <Name Equipment>
<b>Metriken</b>	
<b>Klinische:</b> ▪ <Name Metrik>	<b>Operationale:</b> ▪ <Name Metrik>
	<b>Finanzielle:</b> ▪ <Name Metrik>

Abbildung 23: Template zur inhaltlichen Beschreibung von Prozessen

### 3.2.3 Methoden zur Beurteilung und Verbesserung der Prozessqualität

Es ist unumstritten, dass qualifizierte Mitarbeiter und neueste Technologien einen signifikanten Beitrag zum Geschäftserfolg eines Unternehmens liefern. Doch auch die besten Mitarbeiter können ihre volle Leistung nur dann entfalten, wenn die Prozesse, in denen sie bestimmte Rollen und Aufgaben übernehmen, eine adäquate Qualität aufweisen [CaMe05 S.9f]. Bei der Prozessmodellierung und -optimierung ist deshalb ein entscheidender Parameter, wie Prozessqualität gemessen sowie kontinuierlich und nachhaltig verbessert werden kann.

In Kapitel 3.2.3 werden anhand der Methoden CMMI<sup>57</sup>, EFQM und Six Sigma drei Möglichkeiten zur Bewertung und Optimierung der Qualität von Prozessen vorgestellt. Die dargestellten Methoden sind international verbreitet<sup>58</sup> und anerkannt. Sie werden von zahlreichen Unternehmen zur Verbesserung der Prozessqualität erfolgreich eingesetzt.

<sup>57</sup> Capability Maturity Model Integration.

<sup>58</sup> Ca. 300 Unternehmen weltweit meldeten z.B. den Einsatz von CMMI im Zeitraum April 2002 bis Dezember 2004 an das SEI [CaMe05 S.4].



### 3.2.3.1 Capability Maturity Model Integration

Das Capability Maturity Model findet seinen Ursprung im Jahre 1986 [ACE04 S.13]. Das Department of Defense<sup>59</sup> gab den Auftrag an das Software Engineering Institute<sup>60</sup> der Carnegie Mellon Universität, ein Hilfsmittel zur Beurteilung der Leistungsfähigkeit von Software-Lieferanten zu entwickeln. Daraus entstand 1991 die erste Version des Capability Maturity Models.<sup>61</sup> Mit diesem Modell wollte das Department of Defense Probleme in den Griff bekommen, die in Zusammenhang mit der Einhaltung von Terminen, Qualität und Kosten bei der Entwicklung von komplexen Softwarelösungen standen. 1993 kam nach Einarbeitung des Inputs der Industrie eine überarbeitete Fassung des Capability Maturity Models<sup>62</sup> heraus. Aufgrund des Erfolges wurden weitere CMMs für andere Anwendungsgebiete entwickelt, die teils unterschiedliche Strukturen hatten, teils Redundanzen zum ursprünglichen CMM aufwiesen. Deshalb entschloss sich das SEI, die unterschiedlichen Modelle in ein integriertes Modell zu überführen [ACE04 S.41; Kneu03 S.7f]. Ferner wurde ein Rahmenwerk geschaffen, welches die strukturierte Integration weiterer zukünftiger Modelle ermöglicht [ACE04 S.46]. Die existierenden Methoden zum Durchführen von Prozess-Assessments wurden verfeinert und mit best-practice Beispielen angereichert. Daraus entstand das heute existierende Capability Maturity Model Integration – kurz CMMI genannt [CaMe05 S.14 u. S.16].

Das CMMI ist ein Referenzmodell, welches die Anforderungen an professionell geführte Unternehmen und deren Prozesse für bestimmte Themenbereiche definiert.<sup>63</sup> Dafür stellt es zu jedem Themenbereich sog. „best practices“ zur Verfügung, die quasi als Benchmark für Unternehmen dienen, die CMMI einsetzen. Dadurch, dass CMMI keine konkreten Prozessbeschreibungen enthält, ist es ein Modell, welches in den unterschiedlichsten Unternehmen angewandt werden kann. Primäres Ziel von CMMI ist die strukturierte, kontinuierliche Prozessverbesserung. Aufgrund seiner Struktur kann CMMI für

---

<sup>59</sup> Amerikanisches Verteidigungsministerium.

<sup>60</sup> Kurz SEI genannt; das SEI ist ein staatlich unterstütztes Forschungs- und Entwicklungszentrum.

<sup>61</sup> CMM Version 1.0.

<sup>62</sup> CMM Version 1.1.

<sup>63</sup> Z.B. Softwareentwicklung, Systementwicklung, integrierte Produkt- und Prozessentwicklung.

Prozessverbesserungen innerhalb eines Projektes, eines Teilbereiches in einem Unternehmen oder aber auch für die unternehmensweite Prozessverbesserung eingesetzt werden [CMMI05].

Das CMMI besteht aus 25 Prozessgebieten [ACE S.80], die in Abbildung 24 dargestellt sind.<sup>64</sup> Diese fassen die an ein Unternehmen gestellten Anforderungen – bezogen auf das jeweilige Thema – in strukturierter Form zusammen.

- Organizational Process Focus
- Organizational Process Definition
- Organizational Training
- Organizational Process Performance
- Organizational Innovation and Deployment
- Project Planning
- Project Monitoring and Control
- Supplier Agreement Management
- Risk Management
- Integrated Teaming
- Integrated Supplier Management
- Quantitative Project Management
- Integrated Project Management
- Requirements Management
- Requirements Development
- Technical Solution
- Product Integration
- Verification
- Validation
- Configuration Management
- Process and Product Quality Assurance
- Measurement and Analysis
- Decision Analysis and Resolution
- Organizational Environment for Integration
- Causal Analysis and Resolution

Abbildung 24: Prozessgebiete des CMMI

Zu jedem Prozessgebiet gehören Ziele, die zu erreichen sind. Jedem Ziel sind Praktiken zugeordnet, die zum Erreichen des Zieles beitragen. Die Anforderungen an ein Prozessgebiet sind erst dann erfüllt, wenn alle Ziele, die dem Prozessgebiet zugeordnet sind, vollständig erreicht sind.

Bei den Zielen unterscheidet CMMI zwischen spezifischen und generischen Zielen. Spezifische Ziele gelten nur für das jeweilige Prozessgebiet [ACE S.61; Kneu03 S.12 u. S.17f]. Generische Ziele gelten gleichermaßen für alle Prozessgebiete und sollen die Institutionalisierung<sup>65</sup> der Prozessgebiete im Unternehmen sicherstellen. Abbildung 25 zeigt am Beispiel des Prozessgebietes

---

<sup>64</sup> Auf die Übersetzung der englischen Begriffe, die spezifisch im CMMI verwendet werden, wurde bewusst verzichtet, um eine Verwässerung der Begriffe bei der Übersetzung ins Deutsche zu vermeiden.

<sup>65</sup> D.h. die dauerhafte Umsetzung.

„Organizational Innovation and Deployment“ die zugehörigen spezifischen und generischen Ziele und Praktiken.

<b>Prozessgebiet:</b>	<b>Organizational Innovation and Deployment (Maturity Level 5)</b>
<b>Spezifisches Ziel (1):</b>	<b>Select Improvements</b>
Praktiken:	Collect and Analyze Improvement Proposals Identify and Analyze Innovations Pilot Improvements Select Improvements for Deployment
<b>Spezifisches Ziel (2):</b>	<b>Deploy Improvements</b>
Praktiken:	Plan the Deployment Manage the Deployment Measure Improvement Effects
<b>Generisches Ziel:</b>	<b>Institutionalize a Defined Process</b>
Praktiken:	Establish a Defined Process Collect Improvement Information

Abbildung 25: Spezifische und generische Ziele und Praktiken am Beispiel des Prozessgebietes „Organizational Innovation and Deployment“

CMMI unterscheidet zwei Arten der Darstellung: Die stufenförmige<sup>66</sup> und die kontinuierliche<sup>67</sup> Darstellung [ACE S.71; Kneu03 S.11]. In der stufenförmigen Darstellung gibt es fünf Reifegrade [ACE04 S.72ff; Kneu03 S.14ff], die sog. „maturity levels“ (vgl. Abb. 26). Reifegrade stellen die Qualität der Prozesse eines Unternehmens dar. Im Reifegrad 1 sind die Prozesse in Unternehmen i.d.R. nicht definiert, eher „chaotisch“ und als ad-hoc charakterisiert. Im Reifegrad 2 sind die grundlegenden Prozesse zur Durchführung von Projekten definiert. Dahinter verbirgt sich die Philosophie, dass ein funktionierendes Projektmanagement die Grundlage für die weitere Verbesserung von Prozessen ist und ohne ein funktionierendes Projektmanagement definierte Prozesse nicht umgesetzt werden können.

---

<sup>66</sup> Engl.: Staged.

<sup>67</sup> Engl.: Continuous.

Maturity Level		Process Area
1	Initial	N/A
2	Managed	Requirements Management Project Planning Project Monitoring and Control Supplier Agreement Management Measurement and Analysis Process and Product Quality Assurance Configuration Management
3	Defined	Requirements Development Technical Solution Product Integration Verification Validation Organizational Process Focus Organizational Process Definition Organizational Training Integrated Project Management Risk Management Integrated Teaming Integrated Supplier Management Decision Analysis and Resolution Organizational Environment for Integration
4	Quantitatively Managed	Organizational Process Performance Quantitative Project Management
5	Optimizing	Organizational Innovation and Deployment Causal Analysis and Resolution

Abbildung 26: Reifegrade und zugehörige Prozessgebiete des CMMI staged [CaMe05 S. S.38]

Im Reifegrad 3 sind einheitliche Prozesse für die gesamte Organisation definiert und eingeführt. Hier verlagert sich der Fokus von einzelnen Projekten auf die Organisation als Ganzes. Im Reifegrad 4 werden als Entscheidungsgrundlage für Prozessverbesserungen Performance-Kennzahlen bzw. Kennzahlensysteme genutzt, um die Effizienz und Effektivität der standardisierten Unternehmensprozesse zu beurteilen. Unternehmen mit Reifegrad 5 legen ihren Fokus auf die kontinuierliche Verbesserung und Weiterentwicklung ihrer Prozesse.

Die einzelnen Reifegrade bauen aufeinander auf, so dass ein nächst höherer Reifegrad nur erreicht ist, wenn sowohl die Ziele aller vorausgehenden Reifegrade erzielt wurden, wie auch die spezifischen Ziele des betreffenden Reifegrades selbst. Abbildung 27 zeigt das stufenweise Vorgehensmodell.

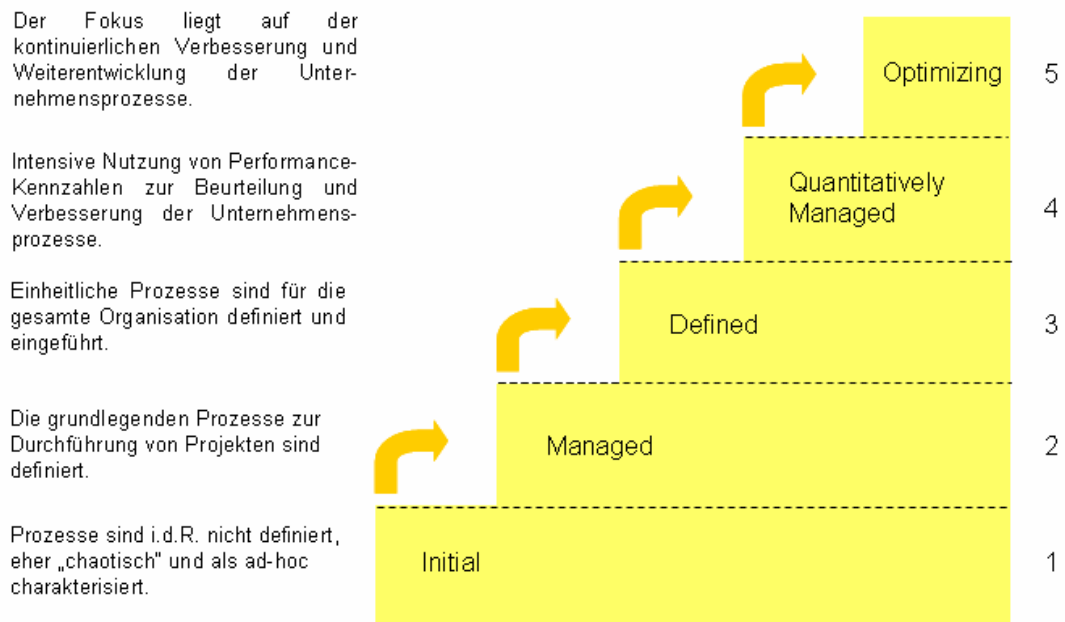


Abbildung 27: Stufenmodell des CMMI staged [ACE04 S.93]

Durch das stufenweise Vorgehen ist der Weg für Prozessverbesserungen im Unternehmen klar und strukturiert vorgegeben. Im stufenweisen Modell sind Vergleiche zwischen den organisatorischen Einheiten im Unternehmen möglich, da sich die Reifegrade auf das gesamte Unternehmen bzw. seine Teilbereiche beziehen.

In der kontinuierlichen Darstellung werden die Prozessgebiete des CMMI auf vier Kategorien verteilt [ACE04 S.106f]:

- Support
- Engineering
- Project Management
- Process Management

Eine Organisation hat hier keinen einheitlichen Reifegrad. Aufgrund der getrennten Bewertung je Prozessgebiet hat die Organisation einen Fähigkeitsgrad je Prozessgebiet, den sog. capability level. Abbildung 28 zeigt die sechs Fähigkeitsgrade, Abbildung 29 zeigt die Zuordnung der Prozessgebiete zu den vier Kategorien des CMMI continuous.

Zum Erreichen eines höheren capability levels muss die Organisation die Ziele des vorherigen capability levels vollständig erreicht haben. Jedes Prozessgebiet wird dabei einzeln mit einem capability level bewertet. Zum Erreichen des jeweiligen capability levels definiert das CMMI vorgegebene Ziele.

Fähigkeitsgrade des CMMI continuous	
0	Incomplete
1	Performed
2	Managed
3	Defined
4	Quantitatively Managed
5	Optimizing

Abbildung 28: Fähigkeitsgrade des CMMI continuous [CaMe05 S.36]

Category	Process Area
Support	Process and Product Quality Assurance Configuration Management Measurement and Analysis Decision Analysis and Resolution Organizational Environment for Integration Causal Analysis and Resolution
Engineering	Requirements Management Requirements Development Technical Solution Product Integration Verification Validation
Project Management	Project Planning Project Monitoring and Control Integrated Supplier Management Supplier Agreement Management Integrated Project Management Risk Management Integrated Teaming Quantitative Project Management
Process Management	Organizational Innovation and Deployment Organizational Process Performance Organizational Process Focus Organizational Process Definition Organizational Training

Abbildung 29: Kategorien und zugehörige Prozessgebiete des CMMI continuous  
 [CaMe05 S. S.33]

Die kontinuierliche Darstellung ermöglicht den Vergleich innerhalb einer Organisation bzw. zwischen Organisationen bezüglich eines bestimmten Prozessgebietes. Unternehmen haben hier die Möglichkeit, die Auswahl und Reihenfolge der Prozessgebiete, die sie optimieren möchten, selbst zu bestimmen.

Kontinuierliche und stufenförmige Form des CMMI sind inhaltlich gleich, unterscheiden sich jedoch in ihrem Aufbau. Abbildung 30 stellt die grundlegenden Unterschiede gegenüber.

Kontinuierliche Darstellung	Stufenförmige Darstellung
<p>Flexibilität in der Reihenfolge der Prozessverbesserungen</p> <p>Fokus liegt auf Prozessverbesserung innerhalb des jeweiligen Prozessgebietes</p> <p>Prozessverbesserungen innerhalb eines Prozessgebietes können unterschiedlich schnell vorangehen</p> <p>Problembereiche in einem Unternehmen können unabhängig vom aktuellen Fähigkeitsgrad adressiert werden</p>	<p>Vorgegebener Verbesserungspfad</p> <p>Fokus liegt auf Prozessverbesserung innerhalb der gesamten Organisation</p> <p>Gesamtergebnisse der Prozessverbesserung spiegeln sich in den Reifegraden wider</p> <p>Reifegrade können aufgrund organisationsweiter Gültigkeit als interner wie auch externer Vergleichsmaßstab herangezogen werden</p>

Abbildung 30: Unterschiede zwischen kontinuierlicher und stufenförmiger Darstellung  
[CaMe05 S.43]

### 3.2.3.2 EFQM

Die European Foundation for Quality Management – kurz EFQM genannt – wurde 1988 von vierzehn führenden europäischen Unternehmen gegründet. Die Zielsetzung der EFQM ist es, „Weltklasse-Ansätze für das Management europäischer Unternehmen bekannt zu machen, die zu nachhaltiger Excellence führen“ [EFQM\_F03 S.3]. Das EFQM-Modell für Excellence wurde „zur Bewertung und Verbesserung von Organisationen eingeführt“ und stellt die Basis dar, um einen nachhaltigen Wettbewerbsvorteil zu erreichen [EFQM\_F03 S.3]. Eines der Grundkonzepte, das hinter dem Modell steht, ist das Management mittels Prozessen und Fakten.<sup>68</sup> Die Philosophie hinter diesem Konzept besagt, dass exzellente Organisationen durch ein „Netzwerk untereinander abhängiger

<sup>68</sup> Weitere Grundkonzepte sind Ergebnisorientierung, Ausrichtung auf den Kunden, Führung und Zielkonsequenz, Mitarbeiterentwicklung und –beteiligung, kontinuierliches Lernen, Innovation und Verbesserung, Entwicklung von Partnerschaften, soziale Verantwortung [EFQM\_F03 S.5ff].



und miteinander verbundener Systeme, Prozesse und Fakten“ gesteuert werden [EFQM\_F03 S.6]. Die systematische Umsetzung der Unternehmensstrategie wird dabei durch ein klar strukturiertes und integriertes Netzwerk von Prozessen, welche pro-aktiv gemanagt und kontinuierlich verbessert werden, sichergestellt. Das EFQM-Modell besteht aus neun Kriterien, die in Abbildung 31 dargestellt sind.

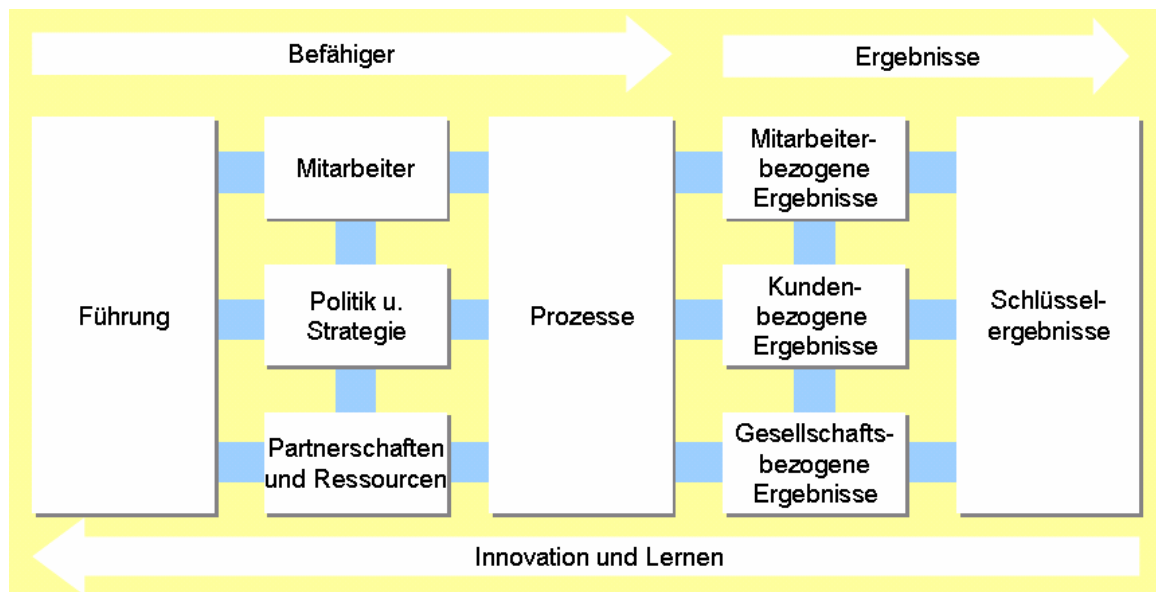


Abbildung 31: EFQM-Modell für Excellence [EFQM\_I03 S.5]

Das Modell unterscheidet die Kriterien nach sog. Befähigern<sup>69</sup> und nach Ergebnis-Kriterien. Zu den Befähigern<sup>70</sup> gehören die Art der Unternehmensführung, der Umgang mit Mitarbeitern, die Unternehmenspolitik und –strategie, Partnerschaften und Ressourcen sowie die zugehörigen Unternehmensprozesse. Prozesse werden dabei im Modell als eigenständiges Befähiger-Kriterium dargestellt. Die erzielten Ergebnisse werden im Modell unterschieden nach mitarbeiterbezogenen, kundenbezogenen, gesellschaftsbezogenen Ergebnissen sowie nach Schlüsselergebnissen [EFQM\_I03 S.5].

Neben dem Kriterium der „Prozesse“ werden zum Zweck der Vollständigkeit auch die übrigen Kriterien [EFQM\_I03 S.13-15] beschrieben.

<sup>69</sup> Engl.: Enabler.

<sup>70</sup> Befähiger-Kriterien befassen sich damit, wie bzw. wodurch die Ergebnisse erreicht werden.

### (1) Führung

Exzellente Führungskräfte stellen die Umsetzung der Mission und Vision des Unternehmens sicher, schaffen die dafür notwendige Unternehmenskultur und leben diese auch erfolgreich vor.

### (2) Politik und Strategie

Exzellente Organisationen leiten aus ihrer Mission und Vision die Unternehmenspolitik und –strategie ab. Diese sind wiederum an den Bedürfnissen ihrer Interessensgruppen ausgerichtet.

### (3) Mitarbeiter

Exzellente Organisationen entwickeln und fördern ihre Mitarbeiter. Sie tragen dazu bei, dass Mitarbeiter ihr gesamtes Potential entfalten können. Leistung der Mitarbeiter wird anerkannt und belohnt. Entsprechende Personalprozesse sind definiert und werden gelebt.

### (4) Partnerschaften und Ressourcen

Exzellente Organisationen planen und managen Partner, Lieferanten und interne Ressourcen. Sie nutzen diese zielgerichtet zur Umsetzung der Unternehmensstrategie sowie zum Erreichen optimaler Prozessabläufe.

### (5) Prozesse

Prozesse werden systematisch gestaltet, gemanagt und kontinuierlich verbessert. Dabei steht der Gedanke im Vordergrund, die Wertschöpfung für alle Stakeholder zu steigern. Bei der Entwicklung von Produkten und Dienstleistungen werden die Bedürfnisse und Erwartungen der Kunden ausreichend berücksichtigt. Adäquate Services für die vermarkteten Produkte und Dienstleistungen werden in ausreichendem Maß angeboten. Kundenbeziehungen werden aktiv gemanagt und kontinuierlich vertieft.

Für den Bereich der Prozesse nennt die EFQM die folgenden Punkte, die im Rahmen einer Bewertung nach dem EFQM-Modell geprüft werden:

- Prozesse werden systematisch gestaltet und gemanagt.

- Prozesse werden nach Bedarf und unter Nutzung von Innovationen verbessert, um Kunden und anderen Interessensgruppen voll zufrieden zu stellen und die Wertschöpfung für sie zu steigern.
- Produkte und Dienstleistungen werden auf Basis der Bedürfnisse und Erwartungen der Kunden entworfen und entwickelt.
- Kundenbeziehungen werden gemanagt und vertieft.

(6)-(9) Kundenbezogene, mitarbeiterbezogene, gesellschaftsbezogene Ergebnisse sowie Schlüsselergebnisse

Exzellente Organisationen führen zu o.g. Themenbereichen Leistungsindikatoren ein, um die Performance in dem jeweiligen Thema kontinuierlich zu messen und zu verbessern. Die erzielten Ergebnisse sind im Vergleich zu anderen Unternehmen ausgezeichnet.

Bei der Bewertung der Befähiger-Kriterien ist entscheidend, welches Vorgehen die Organisation bei jedem Kriterium anwendet und das Ausmaß, in dem das jeweilige Konzept in der gesamten Organisation umgesetzt ist. Bei den Ergebniskriterien wird bewertet, was die Organisation in Bezug auf die einzelnen Kriterien leistet. Dazu wird überprüft, welche Kennzahlen die Organisation verwendet, um die Ergebnisse zu messen, und die Gründe, warum diese Kennzahlen gewählt wurden. Für jede Kennzahl wird die Bedeutung für die Organisation überprüft. Um die Qualität der Ergebnisse beurteilen zu können, wird die Entwicklung der Ergebnisse im Zeitablauf überprüft. Die erzielten Ergebnisse werden mit den ursprünglich geplanten Zielen verglichen, um den Grad der Zielerreichung zu messen. Soweit möglich werden die Ergebnisse mit anderen Unternehmen im Rahmen eines Benchmarking verglichen. Die Organisation muss den Nachweis erbringen, welche Vorgehensweisen zu den Ergebnissen geführt haben [EFQM\_B97 S. 19f; EFQM\_A97 S. 10f]. Die nachfolgenden Abbildungen 32 und 33 fassen die Bewertungskriterien für die Befähiger- und Ergebniskriterien mit der zugehörigen Bewertungsskala zusammen. Dabei werden jeweils für alle Unterpunkte der Befähiger das Vorgehen und die Umsetzung, für alle Unterpunkte der Ergebnis-Kriterien die erzielten Ergebnisse selbst und der

Umfang der Ergebnisse bewertet. Daraus wird dann im Anschluss das arithmetische Mittel für jeden Befähiger und für jedes Ergebnis-Kriterium gebildet.

Vorgehen	Bewertung	Umsetzung
Anekdotisch oder ohne Wertschöpfung	0%	Wenig effektive Anwendung.
Einige Nachweise für fundierte Ansätze und auf Vorbeugung beruhende Systeme. Wird gelegentlich geprüft. Teilweise Integration in die normale Geschäftstätigkeit.	25%	Bei etwa einem Viertel des Potentials angewandt, wenn man alle relevanten Bereiche und Tätigkeiten berücksichtigt.
Nachweis für fundiertes, systematisches Vorgehen und auf Vorbeugung beruhende Systeme. Wird regelmäßig auf geschäftliche Effektivität überprüft. Gute Integration in die normale Geschäftstätigkeit und Planung.	50%	Bei etwa der Hälfte des Potentials angewandt, wenn man alle relevanten Bereiche und Tätigkeiten berücksichtigt.
Klarer Nachweis für fundiertes systematisches Vorgehen und auf Vorbeugung beruhende Systeme. Klarer Nachweis für Verfeinerung und verbesserte geschäftliche Effektivität durch Überprüfungszyklen. Gute Integration in die normale Geschäftstätigkeit und Planung.	75%	Bei etwa drei Viertel des Potentials angewandt, wenn man alle relevanten Bereiche und Tätigkeiten berücksichtigt.
Klarer Nachweis für fundiertes systematisches Vorgehen und auf Vorbeugung beruhende Systeme. Klarer Nachweis für Verfeinerung und verbesserte geschäftliche Effektivität durch Überprüfungszyklen. Vorgehen ist vollkommen in die tägliche Arbeit integriert. Könnte als Vorbild für andere Organisationen dienen.	100%	Beim gesamten Potential in allen relevanten Bereichen und Tätigkeiten angewandt.

Abbildung 32: Bewertungsmatrix der Befähiger-Kriterien<sup>71</sup> [EFQM\_B97 S. 21]

<sup>71</sup> Bei der Bewertung können auch Werte, die zwischen den in der Tabelle angegebenen Bewertungsstufen liegen, herangezogen werden.

Ergebnisse	Bewertung	Umfang
Anekdotisch	0%	Ergebnisse betreffen wenige relevante Bereiche und Tätigkeiten.
Einige Ergebnisse weisen positive Trends und/oder zufrieden stellende Leistungen auf. In einigen Fällen günstige Vergleiche mit den eigenen Zielen.	25%	Ergebnisse betreffen einige relevante Bereiche und Tätigkeiten.
Viele Ergebnisse weisen positive Trends und/oder anhaltend gute Leistungen über mindestens drei Jahre auf. In vielen Bereichen günstige Vergleiche mit eigenen Zielen. Einige Vergleiche mit externen Organisationen. Einige Ergebnisse sind auf das Vorgehen zurückzuführen.	50%	Ergebnisse betreffen viele relevante Bereiche und Tätigkeiten.
Die meisten Ergebnisse weisen deutlich positive Trends und/oder anhaltend hervorragende Leistungen über mindestens drei Jahre auf. Günstige Vergleiche mit eigenen Zielen in vielen Bereichen. Günstige Vergleiche mit externen Organisationen in vielen Bereichen. Viele Ergebnisse sind auf das Vorgehen zurückzuführen.	75%	Ergebnisse betreffen die meisten relevanten Bereiche und Tätigkeiten.
Deutlich positive Trends und/oder anhaltend hervorragende Leistungen seit mindestens fünf Jahren in allen Bereichen. Ausgezeichnete Vergleiche mit eigenen Zielen und externen Organisationen in den meisten Bereichen. „Klassenbester“ in vielen Tätigkeitsbereichen. Ergebnisse sind eindeutig auf das Vorgehen zurückzuführen. Positive Anzeichen, dass Spitzenposition beibehalten wird.	100%	Ergebnisse betreffen alle relevante Bereiche und Tätigkeiten.

Abbildung 33: Bewertungsmatrix der Ergebnis-Kriterien<sup>72</sup> [EFQM\_B97 S. 22]

Um nun die Gesamtpunktzahl zu berechnen, werden die für die Befähiger und Ergebnis-Kriterien ermittelten Werte mit den Faktoren, die den einzelnen Befähigern und Ergebnis-Kriterien zugeordnet sind, multipliziert. Die einzelnen Faktoren sind in Abbildung 34 dargestellt.

<sup>72</sup> Bei der Bewertung können auch Werte, die zwischen den in der Tabelle angegebenen Bewertungsstufen liegen, herangezogen werden.

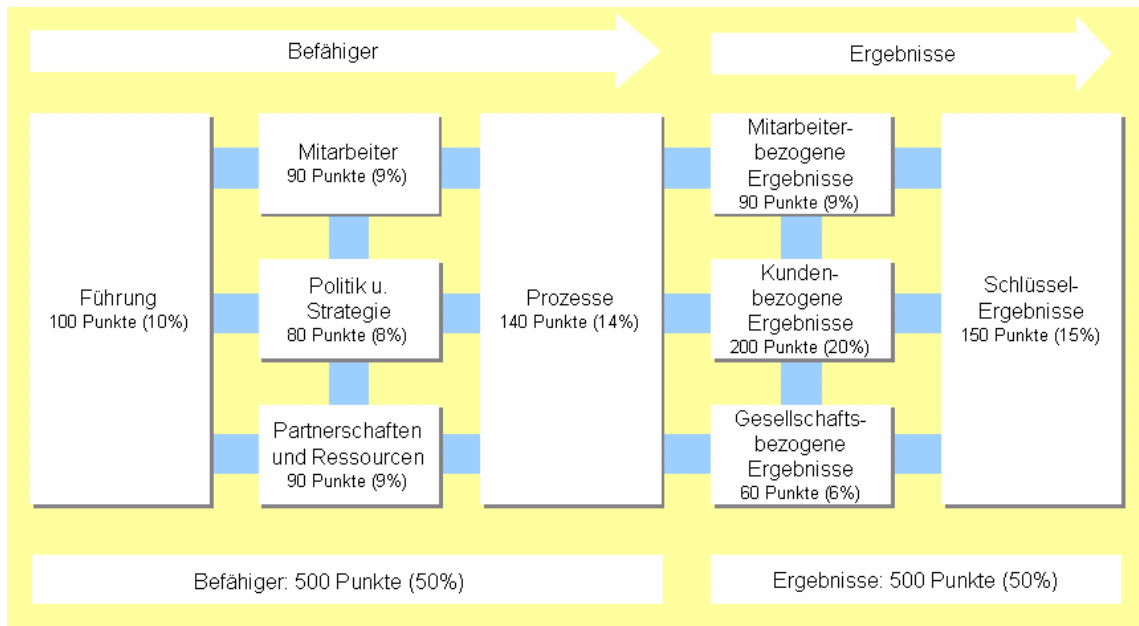


Abbildung 34: Faktorwerte der Befähiger- und Ergebniskriterien gem. EFQM-Modell  
[EFQM\_I03 S.12]

Positiv zu bewerten ist der klare, modulare Aufbau des EFQM-Modells sowie die im Modell dargestellte Ursache-Wirkungsbeziehung der einzelnen Kriterien. Die Schwäche des EFQM-Modells liegt darin, dass keine konkreten Methoden und Tools zur Verfügung gestellt werden, um die gewünschten Ergebnisse bzw. Ergebnisverbesserungen zu erreichen. Die Bewertung der Zielerreichung auf Basis der Bewertungsmaßstäbe kann aufgrund teilweise fehlender, objektiv messbarer Werte, nur durch die subjektive Wahrnehmung des jeweiligen Bewerter erfolgen.

### 3.2.3.3 Six Sigma

Six Sigma ist eine systematische und formalisierte Methodik, die das Ziel hat, durch von Teams geführte Verbesserungsprojekte die Kosten von Unternehmen zu reduzieren bzw. deren Umsätze zu erhöhen. Dies soll erreicht werden, indem Fehler im Prozessablauf behoben bzw. vermieden werden, die Leistung der Prozesse kontinuierlich gemessen und quantifiziert wird, und indem versucht wird, Schwachstellen in den Prozessen zu beheben, bevor diese zu ungewollten Problemen führen [BMB02 S.13]. Dabei soll die Optimierung der Qualität aller

Prozesse im Unternehmen an den Kundenbedürfnissen ausgerichtet werden. Die Definition des Institute of Medicine (IOM) unterstreicht diese Aussage für das Gesundheitswesen. Es definiert Qualität im Gesundheitswesen als den Grad, bis zu dem Anbieter von Gesundheitsdienstleistungen in der Lage sind, die gewünschten (Behandlungs-)Ergebnisse zu erzielen, und inwieweit sich diese Anbieter auf dem neuesten Stand der Medizin befinden [BMB02 S.13]. Die Kundenanforderungen<sup>73</sup> stehen bei Six Sigma im Mittelpunkt. Diese Anforderungen werden critical to quality (CTQ) genannt und bezeichnen das, was für die Qualität wesentlich ist [PNC01 S.44f]. Sobald die Anforderungen klar definiert sind, werden die Fehler/Defekte gezählt, die bei einem Produkt oder einem Prozess auftreten. Fehler/Defekte sind alle Ereignisse, in denen die Anforderungen an ein Produkt oder einen Prozess nicht erfüllt werden [PNC01 S.45].

Die Six-Sigma-Methodik eignet sich besonders zur kontinuierlichen Verbesserung von (Prozess-)Variationen<sup>74</sup>, Durchlaufzeiten und Nutzungsgraden von Einsatzfaktoren (vgl. Abb. 35) [MKB01 S.1ff; Berg04 S.1]. Alle drei Faktoren nehmen eine bedeutende Stellung bei der Beurteilung der Leistung von Prozessen ein.

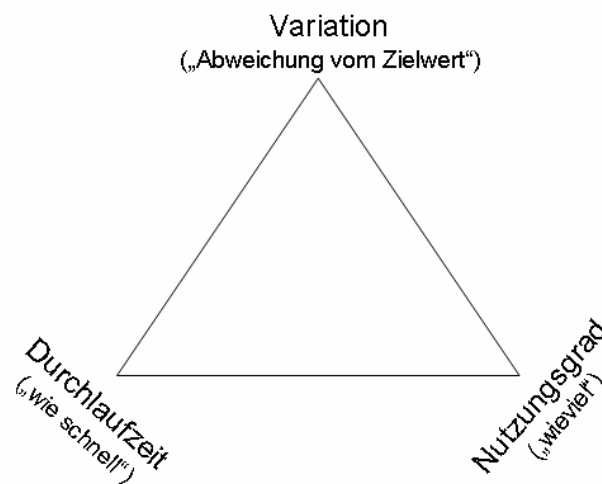


Abbildung 35: Leistungs- und Verbesserungs-dreieck für Prozesse [MKB01 S.3]

<sup>73</sup> Und damit letztendlich die Qualität der erbrachten Leistung.

<sup>74</sup> Abweichungen vom Zielwert.

Der Buchstabe Sigma<sup>75</sup> kommt aus dem griechischen Alphabet und ist im vorliegenden Fall die Maßzahl für Prozessvariation. In der Statistik bezeichnet Sigma ( $\sigma$ ) die Standardabweichung, die ein Indikator für die Summe an Variationen ist [PNC01 S.40]. Im Sinne der Six Sigma Definition entspricht die Leistung eines Prozesse genau dann 6 Sigma, wenn die Variation eines Prozessmerkmals so gering ist, dass bei einer Million Fehlermöglichkeiten nur 3,4 Fehler auftreten.

Fehler, die im Gesundheitswesen auftreten können und leider auch auftreten, sind z.B. falsche oder unvollständig aufgenommene Daten bei der Patientenaufnahme, fehlerhafte Daten in der Patientenakte, falsche oder unvollständige Diagnosen, falsche Medikation, Fehler bei der Abrechnung an die Krankenkasse, verzögerte Entlassung von Patienten und Abweichen von Leitlinien.

Die Verbesserungsmethodik von Six Sigma, DMAIC genannt, besteht dabei aus den fünf Hauptschritten Define, Measure, Analyse, Improve, Control [Selz03 S.66f; PNC01 S.48ff; Berg04 S.17 u. 24 u. 27f; ReYu03 S.95ff].

(1) Define (Definieren):

In der Define-Phase erfolgt die Auswahl der Prozesse, die Gegenstand eines Six Sigma Projektes sein sollen. Das Problem wird identifiziert, Ziele werden gesteckt. Die Auswahl der zu verbessernden Prozesse sollte nach Dringlichkeit erfolgen. In der Define-Phase werden die Tools SIPOC<sup>76</sup>, VoC<sup>77</sup> und CTQ<sup>78</sup> eingesetzt. Mit SIPOC werden in Form einer strukturierten Übersichtsdarstellung die wesentlichen Inhalte und Elemente der betrachteten Prozesse beschrieben. Über Kundeninterviews werden die allgemeinen Kundenanforderungen ermittelt (VoC). Aus diesen Anforderungen werden anschließend messbare Größen generiert und die kritischen Qualitätsmerkmale in einem CTQ-Treiberbaum erfasst.

---

<sup>75</sup> Geschrieben:  $\sigma$ .

<sup>76</sup> Supplier Input Process Output Customer.

<sup>77</sup> Voice of Customer.

<sup>78</sup> Critical to Quality.



(2) Measure (Messen):

In der Measure-Phase werden die Faktoren, die Einfluss auf den Prozess haben, identifiziert. Daten zur aktuellen Prozessqualität werden gesammelt. Die Leistung in Bezug auf die (Kunden-)Anforderungen wird gemessen.

(3) Analyse (Analysieren):

In der Analyse-Phase werden die Ursachen für Prozessvariationen identifiziert. Die vorangegangenen Messungen liefern die Grundlage der Analyse des Zustandekommens der Variationen. Es kann dann überlegt werden, welche Maßnahmen zu ergreifen sind, um Variationen zu reduzieren. Dafür werden z.B. best-practices herangezogen. Wertschöpfende und nicht-wertschöpfende Tätigkeiten sowie Prozessengpässe werden identifiziert. Die anfangs definierten Anforderungen werden – wenn nötig – verfeinert.

(4) Improve (Verbessern):

In der Improve-Phase werden Ideen entwickelt, um die Problemursachen zu beseitigen. Es folgt die Entscheidung, welche Verbesserungen konkret umgesetzt werden. Die entwickelten Lösungen werden getestet. Sofern nötig werden neue oder angepasste Prozesse implementiert.

(5) Überprüfen (Control):

Im Rahmen von Messungen der Prozessleistung wird überprüft, ob und inwieweit die umgesetzte Maßnahme zum Erfolg geführt hat. Die Reduktion der Variationen und die Höhe der Kosteneinsparungen werden berechnet.

Eine interessante Fragestellung, mit der sich auch das SEI intensiv beschäftigt hat [SiFo04], ist die nach der Möglichkeit, Six Sigma komplementär zu CMMI zu benutzen, quasi als „Enabler“ und zum beschleunigten Erreichen eines höheren Prozessreifegrades bzw. Fähigkeitsgrades. In einer von der SEI durchgeführten Untersuchung [SiFo04 S.7ff] wurde deutlich, dass Six Sigma durch seine klare, rationale Vorgehensweise die Entscheidungsfindung in Unternehmen positiv beeinflusst, durch die Einbindung des Managements Veränderungsprozesse in

Unternehmen anstößt und nicht zuletzt nachvollziehbare, quantifizierte Ergebnisse liefert. Die von Six Sigma zur Verfügung gestellten Tools tragen strukturiert und zielgerichtet zur Zielerreichung bei. All das wirkt sich natürlich auch positiv auf das Erreichen eines höheren Reife- bzw. Fähigkeitsgrades gem. der CMMI-Systematik aus. Die nachstehende Abbildung 36 zeigt beispielhaft, wo die Anwendung von DMAIC, der Verbesserungsmethode von Six Sigma, einen Beitrag zum Erreichen eines bestimmten Prozessreifegrades im CMMI-Modell leisten kann [Wil05 S.12f].

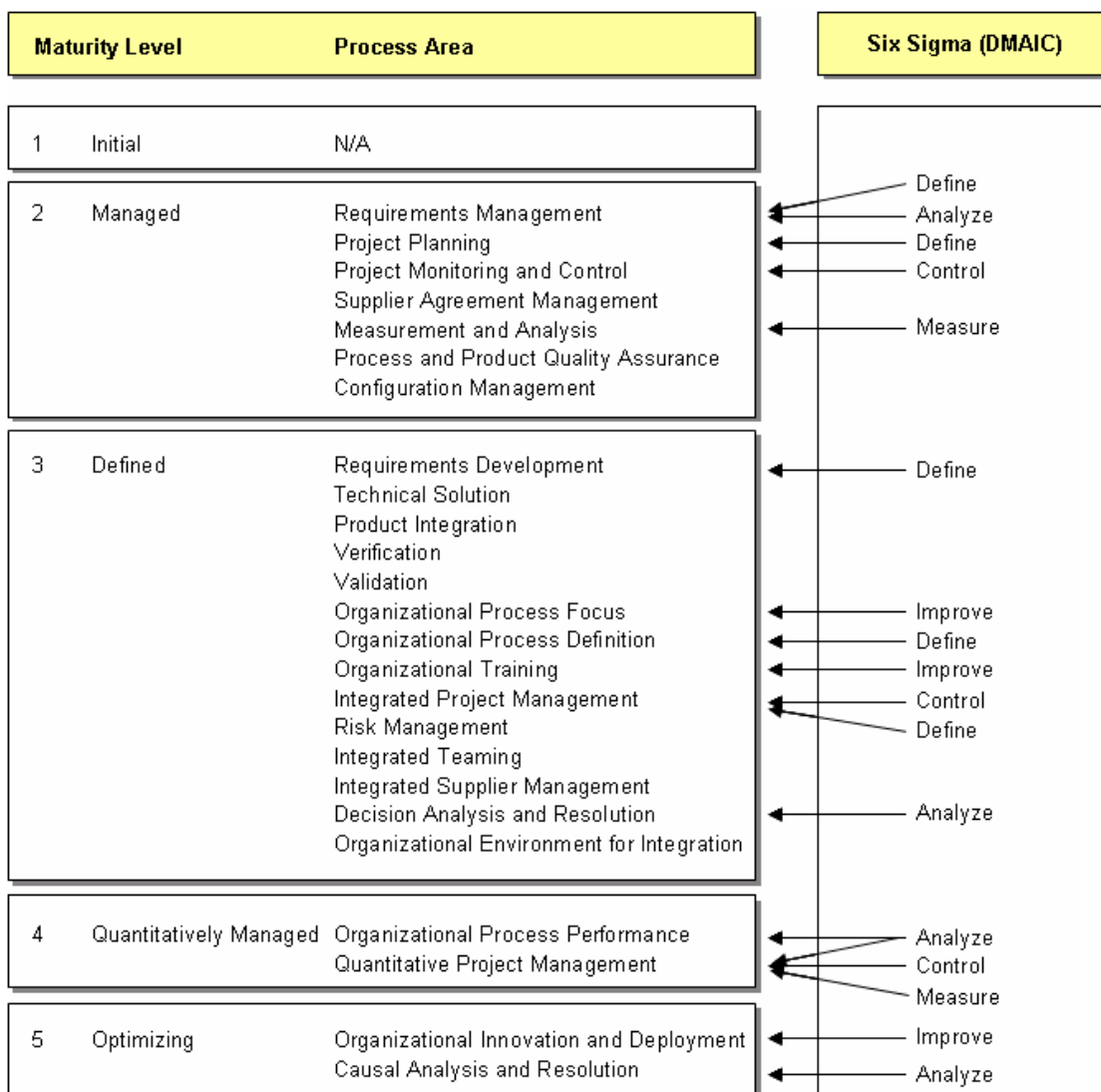


Abbildung 36: Six Sigma als unterstützende Methode für CMMI

Six Sigma gehört mit zu den umfassendsten Methoden, die zur Beurteilung und Verbesserung der Prozessqualität eingesetzt werden können. Durch seinen mathematisch-statistischen Schwerpunkt sind durch die Objektivierbarkeit auch

die Ergebnisse unterschiedlicher Unternehmen gut miteinander vergleichbar. Ferner stellt die Methode eine umfangreiche Toolbox zur Optimierung von Prozessen zur Verfügung und kann zudem – wie oben beschrieben – auch komplementär zu anderen Methoden eingesetzt werden.

### **3.3 Prozessmodelle**

Unter einem Prozessmodell wird in dieser Arbeit die Repräsentation eines realen Prozesses in einem Modell verstanden. Bei der Übertragung bzw. Konstruktion eines Prozessmodells fließen bei der Abbildung des realen Prozesses subjektive Interpretationen des Betrachters ein<sup>79</sup>. Insbesondere zeitliche und logische Zusammenhänge einzelner Prozessaktivitäten werden vom Menschen bei der Modellbildung auf Basis seiner subjektiven Wahrnehmung konstruiert.

Zur Darstellung von Prozessen werden gemeinhin gerichtete Graphen verwendet [BeSc96 S.53; Peer98 S.67]. Gerichtete Graphen sind Netzwerke aus Knoten und verbindenden Kanten, bei dem die Kanten eine Orientierung in eine Richtung (Pfeile) haben [Müll73 S.238]. Aktivitäten sind der Kern der Betrachtung von Prozessmodellen. Ein wesentliches Merkmal ist hier die Existenz einer Sicht auf die zeitlichen und/oder kausalen Interaktionen zwischen den Aktivitäten des betreffenden Prozesses [DIN96 S.18]. Zur Darstellung können verschiedene Techniken bzw. Modellierungssprachen eingesetzt werden, wie zum Beispiel Petri-Netze, ereignisgesteuerte Prozessketten und die UML<sup>80</sup>. Diese werden in den folgenden Kapiteln 3.3.1 bis 3.3.3 in ihren Grundzügen erläutert.

#### **3.3.1 Petri-Netze**

Ein Petri-Netz<sup>81</sup> ist ein gerichteter Graph, dessen Knoten passiven oder aktiven Charakter besitzen. Diese Knoten – auch Komponenten genannt – können durch gerichtete Kanten zu einem Netzwerk verknüpft werden [Baum90 S.42; Brau84

---

<sup>79</sup> Vgl. Kapitel 3.1.

<sup>80</sup> Unified Modeling Language.

<sup>81</sup> Nach Carl Adam Petri.

S.31ff; Peer98 S.68; Krei04 S.47]. Daraus ergeben sich Netze aus Stellen und Transitionen. Stellen sind passive Systemteile und werden als runde Knoten dargestellt. Sie übernehmen Lager- und Speicherfunktionen und können Zustände annehmen. Transitionen sind die aktiven Systemteile und werden als rechteckige Knoten dargestellt. Sie dienen der Erzeugung, Bearbeitung und Weiterleitung beliebiger Dinge im Netz. Mit Hilfe von Pfeilen werden die Systemkomponenten verknüpft. Durch Pfeilgewichte wird angegeben, wie viele Marken aus einer Stelle genommen werden (Demarkierung) und um wie viele Marken die nachgeschaltete Stelle ergänzt wird (Markierung), wenn eine Transition schaltet [Reis82 S.8; Reis85 S.29ff; Baum90 S.50; Pete81 S.7ff; Krei04 S.48].

Transitionen, in die keine Stelle eingeht, sind Markengeneratoren. Transitionen ohne Nachfolger sind Markenvernichter. Die nachfolgenden Abbildungen zeigen anhand eines vereinfachten Beispiels die Wirkungsweise von Petri-Netzen. Abbildung 37 stellt ein Petri-Netz vor dem Schaltvorgang dar. Es besteht aus drei Lagerknoten (=Stellen) sowie einem Produktionssystem (Transition). Für die Produktion von 6 Einheiten eines Gutes O-1, die beim Produktionsprozess hergestellt werden, sind eine Einheit vom Gut I-1 und drei Einheiten vom Gut I-2 notwendig. Die jeweils benötigten Mengen sind als Pfeilgewichte an den Kanten dargestellt. Lager I-1 ist initial mit drei Einheiten an Gut I-1, Lager I-2 ist initial mit vier Einheiten an Gut I-2 vorbelegt.

Beim ersten Schaltvorgang wird Lager I-1 um eine Einheit dekrementiert, Lager I-2 um drei Einheiten. Lager O-1 enthält nach dem Schaltvorgang sechs Einheiten vom Gut O-1 (vgl. Abb. 38).

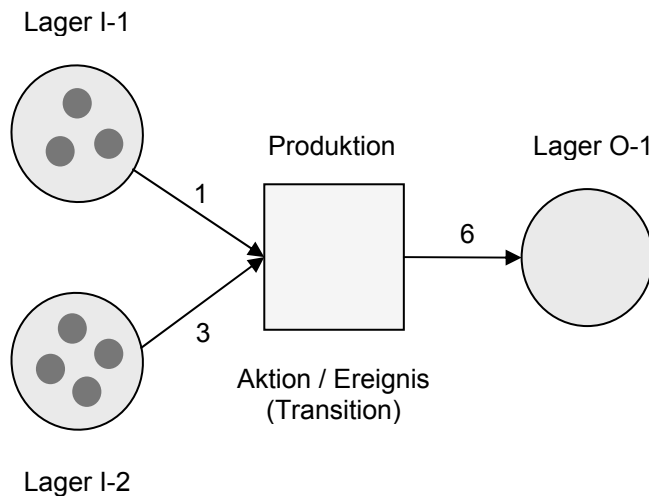


Abbildung 37: Produktionssystem als Netz aus Stellen und Transitionen vor dem ersten Schaltvorgang

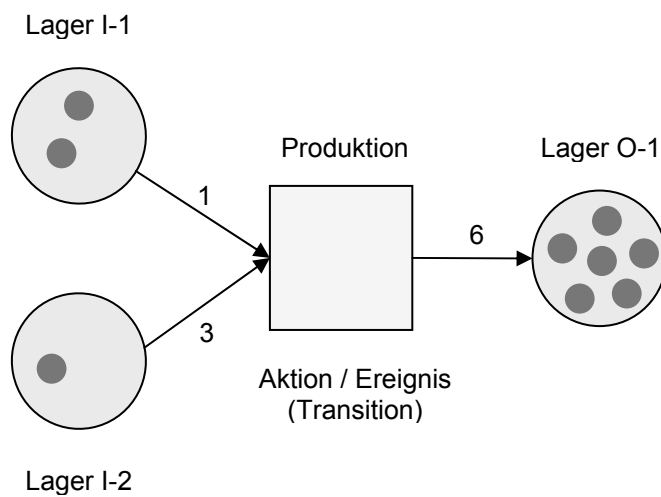


Abbildung 38: Zustand des Produktionssystems nach dem ersten Schaltvorgang

Algorithmen zur Ermittlung des optimalen Ablaufs existieren für Petri-Netze per se nicht. Bei steigender Modellkomplexität sinkt durch die Beschränkung in den graphischen Darstellungsmöglichkeiten von Prozessen die Einsatzmöglichkeit von Petri-Netzen in Ihrer Ursprungsform. Jedoch ist die grundlegende Methodik

der Petri-Netze nicht selten integraler Bestandteil anderer formaler Sprachen<sup>82</sup>, die zur Beschreibung von Prozessen dienen, sowie von Simulationssystemen<sup>83</sup>, die das Grundprinzip von Petri-Netzen übernommen haben.

### 3.3.2 Ereignisgesteuerte Prozessketten

Ereignisgesteuerten Prozessketten versuchen, unterschiedliche Sichten integriert zu betrachten (Funktionssicht, Daten-/Informationsfluss, Organisationssicht). Ereignisse<sup>84</sup> lösen Aktivitäten aus. Aktivitäten werden durch Funktionen dargestellt, welche durch Transformation einen Eingangszustand in einen Endzustand (=neuer Zustand) überführen. Mittels gerichteter Pfeile werden zeitlich-logische Abhängigkeiten zwischen Ereignissen und Funktionen beschrieben (=Kontrollfluss). Ferner stellen ereignisgesteuerte Prozessketten Verknüpfungsoperatoren<sup>85</sup> zur Verfügung. Diese beschreiben die logischen Verbindungen zwischen Ereignissen und Funktionen (vgl. Abb. 39 und Abb. 40).

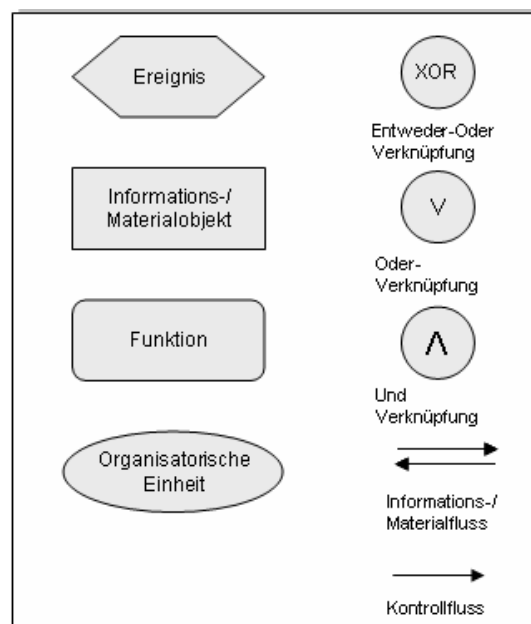


Abbildung 39: Symbole zur graphischen Beschreibung ereignisgesteuerter Prozessketten

<sup>82</sup> Wie z.B. ereignisorientierte Prozessketten (EPK).

<sup>83</sup> Wie z.B. AMTOS [Niem90].

<sup>84</sup> Konkretes Eintreten eines Zustandes.

<sup>85</sup> Entweder-Oder-Verknüpfung / Oder-Verknüpfung / Und-Verknüpfung.

Bei einer konjunktiven Verknüpfung ist eine Gesamtaussage wahr, wenn beide Teilaussagen wahr sind. Bei einer disjunktiven Verknüpfung ist die Gesamtaussage wahr, wenn genau eine der Teilaussagen wahr ist. Bei einer adjunktiven Verknüpfung ist die Gesamtaussage wahr, wenn mindestens eine der Teilaussagen wahr ist [KFG99 S.219ff].

ZUSTAND		VERKNÜPFUNGSART		
A1	A2	Konjunktiv $\wedge$	Disjunktiv	Adjunktiv $\vee$
1	1	1	0	1
1	0	0	1	1
0	1	0	1	1
0	0	0	0	0

Abbildung 40: Verknüpfungsoperatoren und deren Aussageform

Ereignisgesteuerte Prozessketten übernehmen diverse Elemente des Petri-Netz-Ansatzes, wie z.B. die Verknüpfung der Komponenten mittels Pfeilen. Durch den angebotenen Umfang an Elementen zur Beschreibung sowie die Möglichkeit zur Hierarchisierung von Prozessen ist eine realitätsnahe Abbildung von Prozessen möglich. Ereignisgesteuerte Prozessketten werden zur Modellierung von Geschäftsprozessen speziell von ARIS<sup>86</sup>, einer Softwarelösung der IDS Scheer AG [Sche06], genutzt. Da es sich bei ereignisgesteuerten Prozessketten um eine semi-formale Sprache handelt, wird der Nachteil zum einen in dem sich daraus ergebenden hohen Gestaltungsspielraum gesehen, zum anderen darin, dass keine hinreichenden Regeln für eine Modellausführung existieren [Sine03 S.8 u. S.10; KaKu05 S.2; VoFi03 S.3 u. S.8].

<sup>86</sup> ARIS=Architektur integrierter Informationssysteme.

### 3.3.3 Unified Modeling Language

Im Oktober 1994 begann die Rational Software Corporation mit der Entwicklung einer einheitlichen, standardisierten Modellierungssprache. Im Juni 1996 wurde die erste Version der UML, einer grafischen Sprache zur Beschreibung objektorientierter Modelle, veröffentlicht. Diese enthielt bestehende, bereits akzeptierte Diagrammtypen verschiedener objektorientierter Methoden, die entsprechend angepasst, vereinfacht oder ergänzt wurden. Auch bei den Darstellungsmitteln wurde auf bewährte Methoden zurückgegriffen. Beispielsweise enthalten die Aktivitätsdiagramme der UML Ansätze der Datenflußdiagramme und der Petri-Netze [GGB02 S.36]. Das neue an UML sind also nicht die Inhalte an sich, sondern die „Vereinheitlichung zu einer einzigen standardisierten Sprache mit einer formal definierten Bedeutung“ [GGB02 S.36]. UML differenziert in der Modellierung von Systemen zwischen einer externen Sicht und einer internen Sicht [GGB02 S.48f]. In der externen Sichtweise wird das betreffende System von außen betrachtet. Der Fokus liegt hier auf den Geschäftsprozessen, an denen Außenstehende<sup>87</sup> beteiligt sind, im Falle von Unternehmen also Prozesse aus der Sicht der Kunden, Geschäftspartner oder Lieferanten. Die interne Sicht beschäftigt sich mit den unternehmensinternen Geschäftsprozessen, den ihnen zugeordneten Ressourcen sowie der dahinter stehenden Organisationsstruktur. UML bietet verschiedene Diagrammtypen an, um interne<sup>88</sup> und externe<sup>89</sup> Sichten in einheitlicher Weise abzubilden. Die wichtigsten Diagrammtypen sind nachfolgend aufgelistet [GGB02 S.55f u. S.97f u. S.176ff; Krei04 S.8f; Miel02 S.42ff; Rum04 S.41ff u. S.241ff]:

- In Anwendungsfalldiagrammen werden die Akteure für das betrachtete System und deren Beziehung untereinander dargestellt.
- Aktivitätsdiagramme beschreiben die Prozesse des betrachteten Systems. Die Interaktionen zwischen den Akteuren werden dargestellt.

---

<sup>87</sup> Kunden, Lieferanten, Geschäftspartner.

<sup>88</sup> Darstellung der internen Sicht mittels Aktivitätsdiagrammen, Paketdiagrammen und Klassendiagrammen.

<sup>89</sup> Darstellung der externen Sicht mittels Anwendungsfalldiagrammen, Aktivitätsdiagrammen und Sequenzdiagrammen.



- Sequenzdiagramme<sup>90</sup> zeigen den zeitlichen Ablauf von Interaktionen. Diese Diagrammart hat ihren Fokus vor allem auf Nachrichten, die zwischen den Prozessbeteiligten ausgetauscht werden.
- Paketdiagramme beschreiben die Organisationseinheiten in Form sog. Pakete.
- Klassendiagramme beschreiben die Beziehungen zwischen Personen, Geschäftsobjekten und Organisationseinheiten.
- Zustandsdiagramme zeigen das Verhalten der einzelnen Objekte. Sie beschreiben alle Zustände, die ein Objekt annehmen kann sowie die Ereignisse, die die Zustandsänderungen auslösen.

Zur grafischen Darstellung von Diagrammen existieren in UML eine Reihe von Elementen [GBB02 S.58 u. S.76f u. S.90 u. S.99f u. S.105f u. S.157; Krei04 S. 18f], die auszugsweise<sup>91</sup> in den nachfolgenden Tabellen 4a bis 4c dargestellt sind.

---

<sup>90</sup> Ähnlich dem Sequenzdiagramm stellen auch Kollaborationsdiagramme Interaktionen dar.

Kollaborationsdiagramme orientieren sich am statischen Klassendiagramm, eine explizite Zeitachse fehlt.

<sup>91</sup> Die Auswahl wurde auf die gebräuchlichsten und für die Themen der vorliegenden Arbeit relevanten Elemente beschränkt.




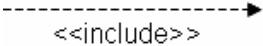


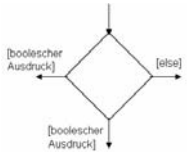
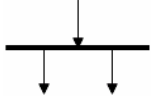
Beschreibung des Elements	Grafische Darstellung
Akteure (z.B. Kunden, Lieferanten, Geschäftspartner) repräsentieren eine Rolle, die jemand einnimmt, wenn er mit einem System interagiert.	
Assoziationen stellen die Beziehung zwischen einem Akteur und einem Geschäftsvorfall/Anwendungsfall dar. Sie geben lediglich an, ob ein Akteur an der Durchführung des Geschäftsvorfalles beteiligt ist.	
Geschäftsanwendungsfälle beschreiben die Interaktionen zwischen dem Akteur und dem System („Welche Leistung nimmt der Akteur in Anspruch?“).	
Include-Beziehungen geben an, dass ein Geschäftsanwendungsfall in einem anderen Geschäftsanwendungsfall enthalten ist.	
Aktivitäten stellen einzelne Schritte im Prozessablauf dar.	
Pfeile beschreiben den Fluss der Aktivitäten. Eingehende Pfeile lösen eine Aktivität aus.	
Rauten stellen Verzweigungen dar, die mindestens einen Eingang und zwei oder mehr Ausgänge haben.	
Aufspaltungen erlauben das Parallelisieren von Aktivitäten.	

Tabelle 4a: Elemente der UML

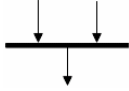


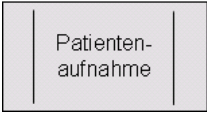
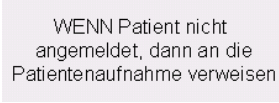
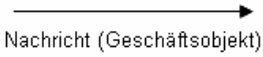
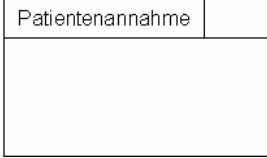
Beschreibung des Elements	Grafische Darstellung
Zusammenführen von zwei oder mehr parallelen Aktivitäten.	
Der Startzustand ist der Startpunkt des Aktivitätsdiagramms.	
Der Endzustand gibt an, wenn die Ausführung der Aktivitäten beendet ist.	
Einzelne Aktivitäten können einem Verantwortungsbereich zugeordnet werden. Verantwortungsbereiche repräsentieren z.B. organisatorische Einheiten.	
Kommentare dienen in den einzelnen Diagrammen der Erläuterung.	
Der Pfeil gibt an, in welche Richtung Nachrichten versendet werden. In Klammern wird das Geschäftsobjekt (z.B. eine Patientenakte, ein Röntgenbild) angegeben, welches mit der Nachricht übermittelt wird.	
Eine Organisationseinheit wird als „Paket“ dargestellt. Der Name der Organisationseinheit steht links oben, Elemente (z.B. Mitarbeiter) werden direkt in das Kästchen eingetragen.	

Tabelle 4b: Elemente der UML


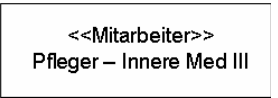

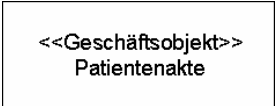
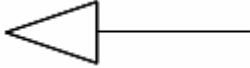
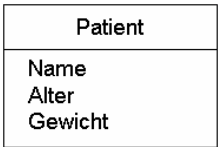
Beschreibung des Elements	Grafische Darstellung
<p>Zur Darstellung von Mitarbeitern können wahlweise zwei unterschiedliche Symbole verwendet werden. Mit dem Begriff Mitarbeiter werden in UML die Rollen von Personen beschrieben, die an einem Geschäftsprozess beteiligt sind.</p>	 
<p>Zur Darstellung von Geschäftsobjekten können wahlweise zwei unterschiedliche Symbole verwendet werden.</p>	 
<p>Eine Generalisierung ist eine Beziehung zwischen einem allgemeinen Element und zwei oder mehr speziellen Elementen. Sie dient der Hierarchisierung.</p>	
<p>Eine Klasse bildet ein Konzept, eine Menge an Personen, Produkten oder Ideen ab. Dazugehörige Attribute repräsentieren die Merkmale einer Klasse.</p>	

Tabelle 4c: Elemente der UML

Für jedes der weiter oben beschriebenen Diagramme wird nun unter Einsatz der eben dargestellten grafischen Elemente ein Beispiel konstruiert und erläutert. Abbildung 41 zeigt ein Anwendungsfalldiagramm.

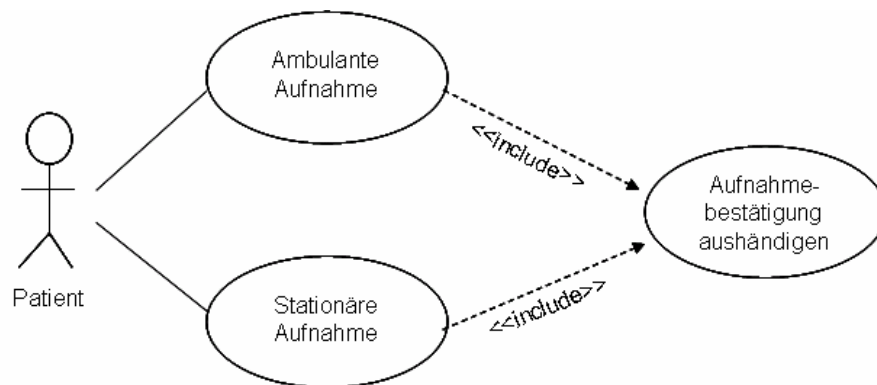


Abbildung 41: Anwendungsfalldiagramm

Das obige Beispiel zeigt das Grundprinzip einer Patientenaufnahme. Der Patient (=Akteur) kann ambulant oder stationär aufgenommen werden. Die include-Beziehung gibt an, dass bei beiden Formen der Patientenaufnahme (=Geschäftsanwendungsfall) eine Aufnahmebestätigung ausgehändigt wird.

Das Aktivitätsdiagramm in Abbildung 42 beschreibt den vereinfachten Ablauf einer Patientenaufnahme. Es beginnt mit dem schwarzen Punkt (=Startzustand) und setzt sich entlang der Pfeile fort. Das Aktivitätsdiagramm ist in zwei Verantwortungsbereiche aufgeteilt: Dem Verantwortungsbereich des Patienten und dem Verantwortungsbereich der Patientenaufnahme. Der Patient ist verantwortlich für das Vorzeigen seiner Krankenversicherungskarte. Die Patientenaufnahme ist verantwortlich für das Einlesen der Daten, die auf der Versichertenkarte gespeichert sind, sowie für die Überprüfung des vereinbarten Aufnahmetermins. Wurde der Termin korrekt vereinbart, werden dem Patienten die Krankenversicherungskarte sowie die Einweisungspapiere ausgehändigt. Wurde kein Termin vereinbart, wird dem Patienten die Krankenversicherungskarte zurückgegeben und ein Aufnahmetermin abgestimmt.

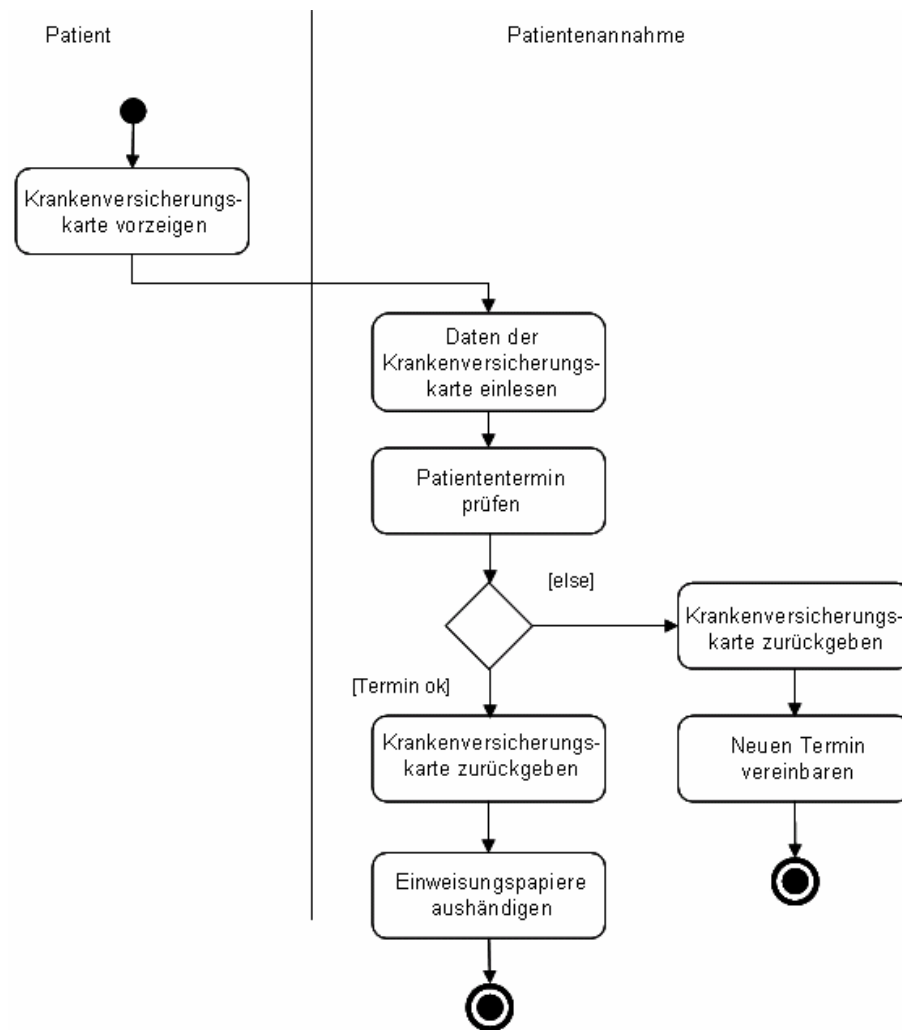


Abbildung 42: Aktivitätsdiagramm am Beispiel der Patientenaufnahme

Abbildung 43 zeigt für den vereinfachten Fall der stationären Patientenaufnahme das zugehörige Sequenzdiagramm, welches seinen Fokus auf die Darstellung von Interaktionen zwischen den beteiligten Parteien sowie auf den Austausch von Nachrichten legt.

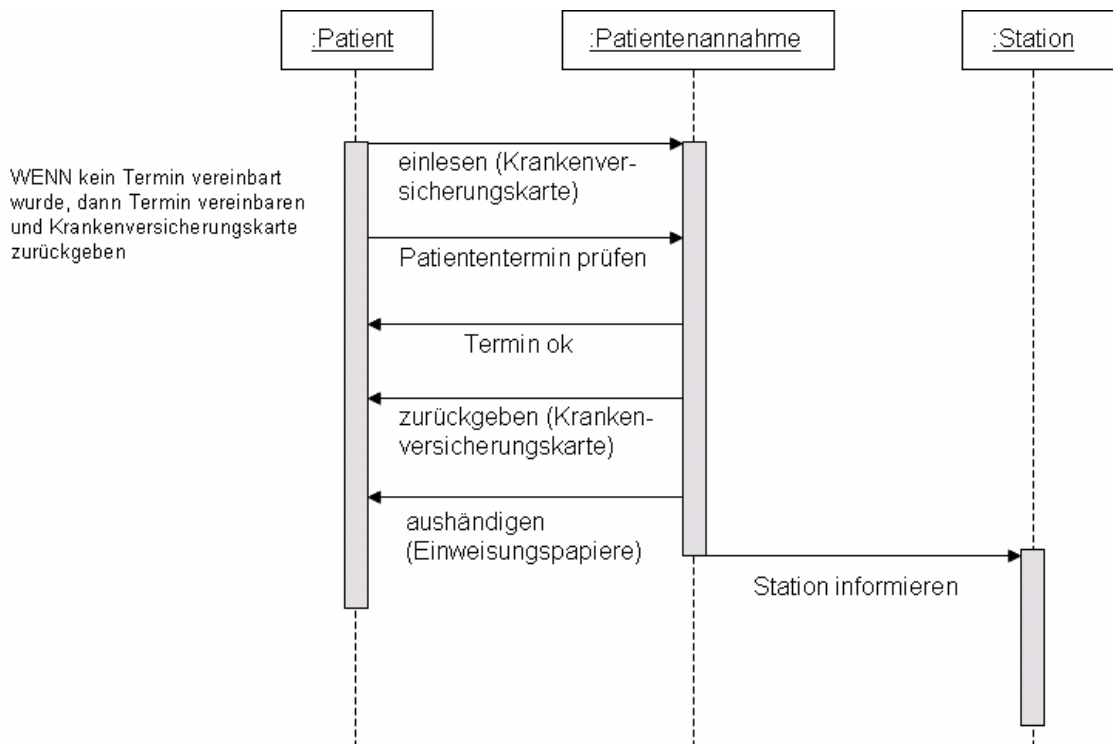


Abbildung 43: Sequenzdiagramm am Beispiel der Patientenaufnahme

Die Sequenz beginnt damit, dass der Patient seine Krankenversicherungskarte übergibt und die Patientenannahme die Daten einliest. Die Aufforderung „einlesen“ ist die Nachricht, die Krankenversicherungskarte ist das Geschäftsobjekt, welches übergeben wird. Wurde ein Termin vereinbart, wird dem Patienten die Krankenversicherungskarte zurückgegeben und die Einweisungspapiere ausgehändigt. Anschließend wird die aufnehmende Station über die Ankunft und Aufnahme des Patienten informiert. In Kommentarfeldern<sup>92</sup> können Hinweise für die Bearbeitung des Geschäftsfalls angegeben werden. Echte Verzweigungen (IF-THEN-ELSE) können in Sequenzdiagrammen nicht angegeben werden.

Die genauen Aktivitäten, die durch den Erhalt von Nachrichten in der Patientenannahme angestoßen werden, sind im Sequenzdiagramm nicht ersichtlich. Hierfür dienen die Aktivitätsdiagramme.

Das nachfolgende Paketdiagramm (vgl. Abb. 44) stellt einen vereinfachten Aufbau der Organisationseinheit „Zentrale Patientenaufnahme“ dar.

<sup>92</sup> Links in der Abbildung 43 zu sehen.

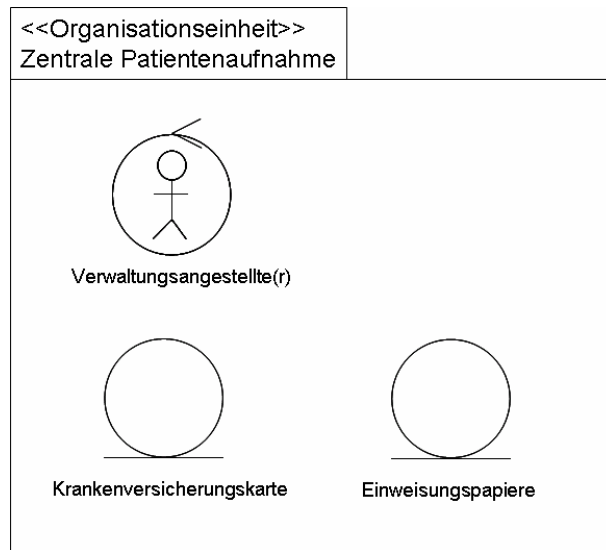


Abbildung 44: Paketdiagramm am Beispiel der Organisationseinheit „Zentrale Patientenaufnahme“

Der Name der Organisationseinheit wird in das linke obere Kästchen eingetragen. In dem darunter befindlichen Kasten werden die beteiligten Mitarbeiter (im Beispiel der Verwaltungsangestellte) sowie die beteiligten Geschäftsobjekte (im Beispiel die Krankenversicherungskarte und die Einweisungspapiere) eingetragen. Das Mitarbeiter-Symbol gibt im Paketdiagramm die jeweilige Rolle an, nicht jedoch die Anzahl der Mitarbeiter, die diese Rolle tatsächlich ausfüllen.

Das in Abbildung 45 dargestellte Klassendiagramm zeigt die Zusammenhänge zwischen den Mitarbeitern der Patientenaufnahme und den beteiligten Geschäftsobjekten. Der Verwaltungsangestellte, der ein Mitarbeiter der zentralen Patientenaufnahme ist, prüft den Überweisungsschein (=Geschäftsobjekt) und übergibt die Einweisungspapiere (=Geschäftsobjekt).



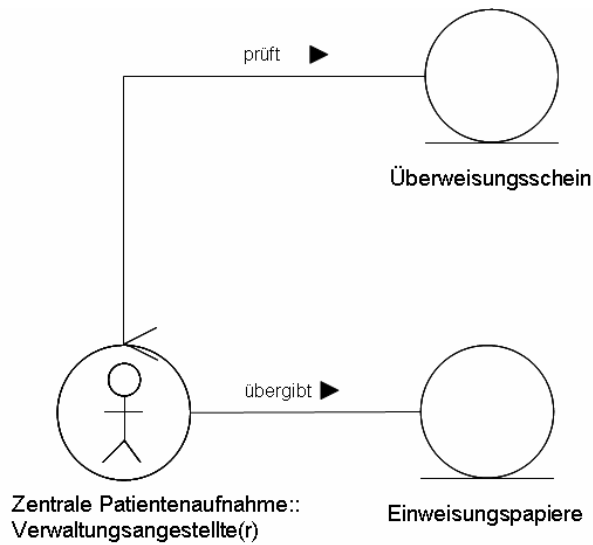


Abbildung 45: Vereinfachtes Klassendiagramm am Beispiel der Patientenaufnahme

Das nachstehende Zustandsdiagramm zeigt alle möglichen Zustände, die das Objekt „Kernspintomograph“ im System annehmen kann (vgl. Abb. 46). Bevor der Kernspintomograph Teil des betrachteten Systems werden kann, muss er bestellt (=Zustand „bestellt“) und geliefert werden. Nach Lieferung hat das Gerät den Zustand „in Wartung“, da er nach Aufstellung und vor Inbetriebnahme einer Komplettprüfung unterzogen wird. Wird das Gerät aktuell nicht gewartet, kann es die Zustände „einsatzbereit“ bzw. „in Betrieb“ (=Durchführen einer Diagnose am Patienten) annehmen.

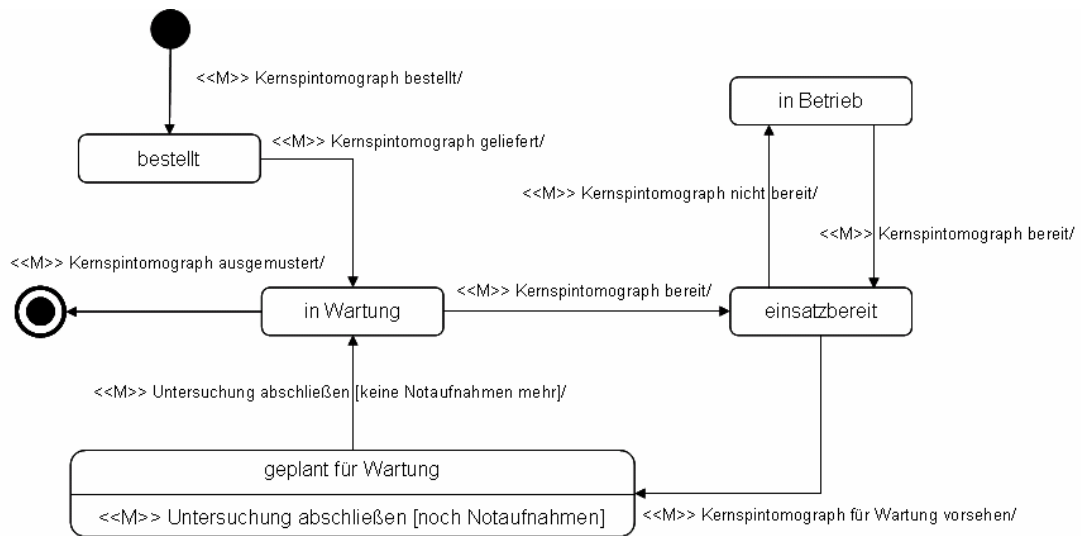


Abbildung 46: Zustandsdiagramm am Beispiel eines Kernspintomographen

Ist eine Wartung geplant (Zustand „geplant für Wartung“), muss eine eventuell laufende Untersuchung erst beendet werden, bevor die Wartung beginnen kann. Die in eckigen Klammern angegebene Bedingung<sup>93</sup> „noch Notaufnahmen“ gibt an, dass eine Wartung nicht durchgeführt werden kann, wenn Notfallpatienten eingeliefert wurden, die umgehend untersucht werden müssen. Erst wenn das Gerät für Notaufnahmen nicht mehr zur Verfügung stehen muss, kann es den Zustand „in Wartung“ annehmen.

Der Vorteil von UML als Modellierungssprache wird zum einen in der Vereinheitlichung der Terminologie und der Standardisierung<sup>94</sup> der Notationen gesehen. Beides schafft Transparenz, klare Strukturen und erleichtert letztendlich die Verständigung und Zusammenarbeit der an der Modellierung beteiligten Personen [GBB02 S.13]. Zum anderen können trotz der Mächtigkeit von UML nicht nur komplexe Sachverhalte, sondern auch einfache Modelle mit UML entwickelt werden. Jedoch liegt es letztendlich an dem Geschick des Anwenders, die richtigen und für den jeweiligen Zweck sinnvollen Komponenten der UML einzusetzen - was aufgrund des Umfangs durchaus eine Herausforderung darstellen kann.

<sup>93</sup> Auch Wächterbedingung genannt.

<sup>94</sup> Hinter der Standardisierung steht die Object Management Group, ein internationaler Verband zur Förderung offener Standards für objektorientierte Anwendungen.

#### **4. Grundlegende Komponenten von Simulationssystemen – Ein Überblick**

Simulationssysteme müssen unterschiedlichste Aspekte berücksichtigen, um Anwendern eine effiziente Modellbildung und -analyse zu ermöglichen. In diesem Zusammenhang ist die entscheidende Frage zu beantworten, wer die Anwender von Simulationsmodellen sind. Im Idealfall sollte die Entwicklung von Simulationsmodellen von Personen vorgenommen werden, die zum einen fundierte Kenntnisse über die zu modellierenden Prozesse haben, zum anderen sollten diese zugleich auch die Anwender des betreffenden Simulationsmodells sein. Anwender sind i.d.R. das Management und die Fachkräfte des jeweiligen Unternehmens bzw. Unternehmensbereiches, in dem Prozesssimulation zum Einsatz kommen soll. Im produzierenden Gewerbe sind dies z.B. Produktionsleiter oder Qualitätsmanager, in Krankenhäusern z.B. Ärzte, Pflegepersonal oder die Verwaltungsleitung. Anwender von Simulationssystemen müssen in die Lage versetzt werden, Modelle weitgehend eigenständig zu entwickeln und auszuwerten – selbst ohne fundiertes IT-Wissen. Nur so wird eine ausreichende Akzeptanz der Prozesssimulation bei diesen Personengruppen erreicht. Um dies zu ermöglichen, sollten Simulationssysteme u.a. folgende Aspekte berücksichtigen:

Komponenten von Simulationssystemen müssen klar und verständlich, modular aufgebaut und wieder verwendbar sein. Modelle sollten - soweit möglich - ohne weitergehende Programmierkenntnisse entwickelt werden können. Dies wird u.a. durch Parametrisierbarkeit von Modellkomponenten erreicht. Dem Anwender sollten verschiedene Standardauswertungen zur Verfügung stehen, wie z.B. zur Analyse der Stillstandszeiten von Maschinen sowie von Ressourcen- und Kapazitätsengpässen, die im Prozessablauf auftreten.

Wie dies konkret aussehen kann, wird in den nachfolgenden Kapiteln 4.1 und 4.2 beschrieben<sup>95</sup> und anhand relevanter Beispiele verdeutlicht.

---

<sup>95</sup> In Anlehnung an das Simulationssystem AMTOS.

## 4.1 Basiskomponenten zur Modellierung von Unternehmensprozessen

### 4.1.1 Automaten

Automaten dienen der Darstellung von zeitverbrauchenden Aufgaben (=Tasks), wie dies bei Unternehmensprozessen der Fall ist. Automaten können wie folgt unterteilt werden (vgl. Abb. 47) [Niem90 S.172]:

- (1) Initiale Automaten haben zu mindestens einem Buffer eine Sender-Beziehung, aber zu keinem Buffer eine Empfänger-Beziehung. Sie können in einem Simulationsmodell für die Generierung von Aufträgen (z.B. Produktionsaufträge) eingesetzt werden.
- (2) Transiente Automaten haben zu mindestens einem Buffer eine Empfänger-Beziehung und zu mindestens einem Buffer eine Sender-Beziehung. Diese Automatenart nimmt Aufträge von vorgelagerten Prozessschritten an, verarbeitet sie weiter und leitet sie dann an den nächstfolgenden Prozessschritt.
- (3) Terminale Automaten haben zu mindestens einem Buffer eine Empfänger-Beziehung, aber zu keinem Buffer eine Sender-Beziehung. Sie können in einem Simulationsmodell zum Löschen von bearbeiteten Aufträgen eingesetzt werden. Vor dem Löschen können informationsrelevante Daten in eine Datei geschrieben und gespeichert werden.
- (4) Isolierte Automaten haben zu keinem Buffer eine Empfänger- bzw. Sender-Beziehung. Sie werden z.B. zur Darstellung von Leitsystemen eingesetzt.

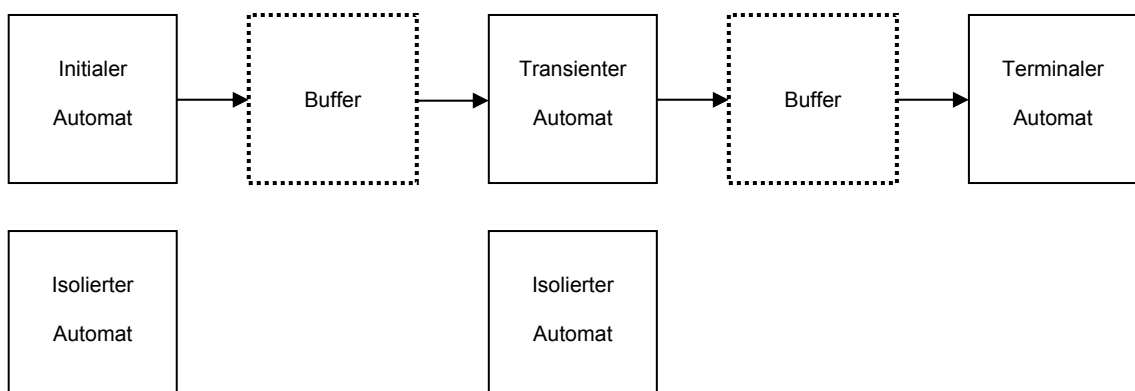


Abbildung 47: Arten von Automaten – Beispielhafte Darstellung

Zu semantischen Unterstützung bei der Modellentwicklung und späteren Auswertung sollten Automaten, die Bestandteil des entwickelten Simulationsmodells sind, mit benutzerindividuellen Bezeichnern versehen werden können (vgl. Abb. 48). Der Vorteil von Bezeichnern liegt darin, dass sich Anwender durch den geringeren Abstraktionsgrad und den Bezug zu ihrer „Realität“ besser mit dem Modell identifizieren können.

Bezeichner für Automaten müssen mehrfach vergeben werden können. Deshalb muss Simulationssoftware programmintern mit Automatennummern arbeiten, die eindeutig sind und den jeweiligen Automaten unabhängig von dessen benutzerindividuellen Bezeichnung identifiziert.

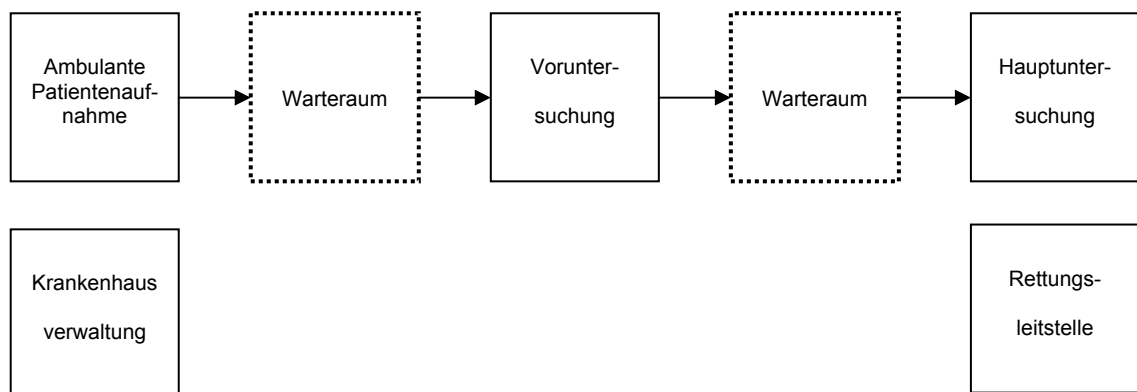


Abbildung 48: Beispiel für benutzerindividuelle Bezeichnung von Automaten

Im Folgenden wird darauf eingegangen, wie Automaten parametrisiert werden können. Die hierfür gewählten Bezeichnungen für die verwendeten Parameter sind beispielhaft und können in unterschiedlichen Simulationssystemen unterschiedlich benannt sein.

Die Anzahl möglicher Startvorgänge bei Automaten wird nachfolgend mit dem Parameter COUNT bezeichnet. Wird dem Parameter COUNT ein bestimmter Wert zugewiesen, wird bei jedem Start des Automaten der Wert COUNT um 1 dekrementiert. Der Automat kann so oft starten, bis der dekrementierte Count-Wert den Wert 0 erreicht. Wird kein Wert zugewiesen, soll der Automat unendlich oft starten können.<sup>96</sup> Ein Beispiel verdeutlicht die praktische Bedeutung dieses Parameters: Röntengeräte sollen nach einer bestimmten Anzahl von

<sup>96</sup> In Abbildung 49 verdeutlicht mit dem Zeichen  $\infty$ .

Aufnahmen<sup>97</sup> einer Gerätewartung unterzogen werden, bevor weitere Aufnahmen gemacht werden. Beträgt das Wartungsintervall z.B. 2.000 Aufnahmen, so würde der COUNT-Wert im Modell auf 2.000 eingestellt.

Der Parameter DURATION bestimmt die Laufzeit eines Automaten. Die Vorgangsdauer kann vom Anwender für jeden Automaten individuell gesetzt werden. Die standardmäßige Vorbelegung ist 1. Ein variables Setzen des Parameters über Usercode<sup>98</sup> ist ebenso möglich wie eine feste, voreingestellte Dauer des betrachteten Automaten. Nachfolgendes Beispiel zeigt die praktische Relevanz: Ein Patient kommt in die ambulante Aufnahme eines Krankenhauses. Für die Aufnahme von persönlichen Daten und der Krankheitsgeschichte benötigt das Personal im Mittel 5 Minuten. Die Vorgangsdauer des entsprechenden Automaten wird vom Anwender auf 5 festgelegt. 5 Zeiteinheiten entsprechen in diesem Fall 5 Minuten realer Zeit. Da es sich hier lediglich um einen Mittelwert handelt, muss die Möglichkeit bestehen, Streubereiche für den Wert der Vorgangsdauer anzugeben. Die Einführung des Parameters DURATION\_MOD ermöglicht es, gleichverteilt um den Mittelwert DURATION zu „streuen“. Über einen Zufallszahlen-Generator werden dann Vorgangsdauern erzeugt, die zwischen  $(DURATION-DURATION\_MOD)$  und  $(DURATION+DURATION\_MOD)$  liegen.

In der Kardiologie können die beiden Parameter zum Beispiel wie folgt eingesetzt werden: Je nach Schweregrad der Herzerkrankung und damit zusammenhängender Co-Morbiditäten bewegt sich die Behandlungsdauer des betreffenden Patienten innerhalb einer gewissen Streubreite um einen vorgegebenen Mittelwert. Dieses Szenario kann durch Belegen der oben beschriebenen Parameter im Modell dargestellt werden.

Im Kontext der Vorgangsdauer eines Automaten fehlt nun noch ein Parameter, der die Restlaufzeit des Automaten ausgibt. Im Folgenden ist dieser als REMAINDER\_DURAT bezeichnet. Beim Start des betreffenden Automaten enthält der Parameter REMAINDER\_DURAT die für die aktuelle Task berechnete Laufzeit vermindert um den Wert 1, da der Automat bereits gestartet ist. Während des Modelllaufs wird der Parameter in jedem Zeitschritt um 1 dekrementiert. Der Automat wird beendet, wenn der Wert REMAINDER\_DURAT = 0 ist. Während

---

<sup>97</sup> Dies entspricht in einem Modell der Anzahl von Startvorgängen des betrachteten Automaten.

<sup>98</sup> Individuelle Entwicklung von Programmcode im Simulationsmodell.

des Simulationslaufs kann der Anwender beobachten, wann eine Task beendet wird und damit für den nächsten Auftrag frei ist.

Durch den Parameter `WAITING_TIME` kann der Anwender angeben, wie viele Zeiteinheiten ein Automat nach Beendigung seiner Task inaktiv bleiben soll, bevor er wieder gestartet werden kann. Die Vorbesetzung mit dem Wert 0 bedeutet, dass der Automat sofort nach Beendigung einer Task erneut gestartet werden kann. In der Praxis kann dieser Parameter zum Beispiel in folgendem Fall angewandt werden: Ein Automat wird im Modell zur Abbildung eines Operationssaals benutzt. Die Vorgangsdauer kann - je nach Art und Dauer der Operation - über den Parameter `DURATION` gesetzt werden. Ist die OP abgeschlossen, muss der Operationssaal gereinigt und desinfiziert werden. Während dieser Zeit kann der Operationssaal nicht genutzt werden. Benötigt der Reinigungsprozess z.B. 30 Minuten, so würde der Parameter `WAITING_TIME` bei Minutentaktung des Simulationsmodells auf 30 Zeiteinheiten gesetzt.

Insbesondere bei der Simulation von klinischen Prozessen ist ein weiterer Wert zur Beurteilung von Prozessen von Bedeutung: Das spätest zulässige Ende eines Prozessschrittes. Tritt bei einem Patienten Herzstillstand auf, muss dieser innerhalb von wenigen Minuten wiederbelebt werden, da sonst der Hirntod eintreten kann. Wird dieser Prozessschritt nicht rechtzeitig durchgeführt und abgeschlossen, kann der Patient sterben. Um diese laufzeitkritischen Prozesse statistisch auszuwerten, wird der Parameter `DEADLINE` eingeführt. Er überwacht die Einhaltung der Laufzeiten einzelner Automaten. Bei einer Vorbelegung `DEADLINE = 0` bleibt der Wert in späteren Auswertungen unberücksichtigt.

Der Parameter `VARCOST_MOD` modifiziert die variablen Personalkosten aufgabenspezifisch. Vor Berechnung der variablen Personalkosten werden diese mit dem `VARCOST_MOD`-Faktor des jeweiligen Automaten multipliziert. So lassen sich in den variablen Kosten auf flexible Art und Weise Zu- und Abschläge berücksichtigen, wie z.B. Nachtzulagen, Sonn- und Feiertagszulagen, Schmutzzulagen und Gefahrenzulagen. In der klinischen Anwendung können z.B. unterschiedlich hohe Sätze für Kassen- und Privatpatienten berücksichtigt werden.

Abbildung 49 zeigt den Aufbau einer Bildschirmmaske zur Parametrisierung der einzelnen Automaten. Der in der Abbildung verwendete englische Begriff „Task“

ist hier mit dem entsprechenden Automaten, der diesen Prozessschritt abbildet, gleichgesetzt. Eine Anpassung der Parameter sollte auch während der Simulation eines Modells möglich sein. Dies gewährt höchstmögliche Flexibilität. Nach dem Anhalten des Simulationslaufs können Parameter verändert werden, die Simulation kann dann ab dem Zeitpunkt, an dem sie gestoppt wurde, mit veränderten Parametern weiterlaufen. Parameter können alternativ über Usercode während des Simulationslaufs gesetzt bzw. verändert werden.

Task Parameters		Time: 120			Model: <name of model>
Task Parameter	Patienten- aufnahme	Vorunter- suchung	Hauptunter- suchung	Behandlung	<name>
COUNT	∞	∞	∞	∞	0
DURATION	122	12	3	7	0
DURATION_MOD	0	0	5	0	0
WAITING_TIME	0	0	0	0	0
REMAINDER_DURAT	0	2	4	0	0
CONFIG	0	0	0	0	0
FATHER	0	0	0	0	0
EARNINGS	0	0	0	125	0
COST	12	5	18	25	0
PRIORITY	0	0	0	0	0
DEADLINE	0	0	0	0	0

Abbildung 49: Bildschirmmaske zur Parametrisierung von Automaten<sup>99</sup>

Um genauere Informationen über das Laufzeitverhalten von Automaten zu erhalten, kann die nachfolgende Bildschirmsicht genutzt werden (vgl. Abb. 50). Sie enthält neben den oben bereits beschriebenen Parametern Aussagen zu den eingesetzten Ressourcen, dem benötigten (Produktions-)Material sowie notwendiges Equipment zum Ausführen des Prozesses. Rote Textfarbe kennzeichnet aufgetretene Engpässe. Im gezeigten Beispiel kann der Automat nicht anlaufen, da notwendiges Personal (im vorliegenden Fall „Herr Huber“)

<sup>99</sup> Abbildung in Anlehnung an [Niem90 S. 31].



sowie notwendiges Equipment (im vorliegenden Fall „Stethoskop“) nicht verfügbar ist.

Status of Task <internal number>/<name>		Time: 120	Model: <name of model>				
Parameter		Resources					
COUNT	32767	<b>Personnel</b>					
DURATION	12	Skilled	Maier	Müller	Huber	Schneider	<name>
DURATION_MOD	1	Avail./Busy	1	1	0	1	0
WAITING_TIME	0	<b>Equipment</b>					
REMAINDER_DURAT	0	Necessary	Kanülen	Behälter	Stethoskop	<name>	
CONFIG	0	Availability	2	1	0	0	
FATHER	0	<b>Material</b>					
EARNINGS	0	Necessary	Narkotika	B-Blocker	<name>	<name>	
COST	0	Availability	20	32	0	0	
PRIORITY	0	<b>Statistics</b>					
DEADLINE	0	<b>Time-related</b>		<b>Bottleneck/Shortage</b>			
VARCOST_MOD	0	Active	80	Personnel	3		
		Blocked	10	Equipment	5		
		Idle	30	Material	1		
		Active/Total Time	67%	Buffer	7		

Abbildung 50: Bildschirmmaske zur Beobachtung des Laufzeitverhaltens der Automaten<sup>100</sup>

Die „Statistics“ zeigen die jeweils aktuelle Auswertung für die aktive Zeit des Automaten (active), Stillstandszeiten (idle bzw. blocked) sowie das prozentuale Verhältnis der aktiven Zeit des betrachteten Automaten zur Gesamtlaufzeit des Simulationsmodells. Ferner werden die Anzahl der aufgetretenen Engpässe (=Bottlenecks) und deren Ursachen gezeigt. In obigem Beispiel konnte der Automat mehrfach nicht rechtzeitig starten, 3x wegen Personalengpass, 5x wegen fehlendem Equipment, 1x wegen fehlendem Material und 7x wegen vollständig belegter Buffer.

<sup>100</sup> Abbildung in Anlehnung an [Niem90, S. 26].

#### 4.1.2 Input-Output-Verknüpfung von Automaten

Input-Output-Verknüpfungen von Automaten werden im Modell grafisch durch Pfeile dargestellt. Sie dienen zudem der Synchronisation von Automaten. Automaten können ihren Output über einen oder mehrere Pfeilverbindungen an nachgelagerte Automaten oder explizite Buffer liefern. Umgekehrt können Automaten von einer oder mehreren Pfeilverbindungen Input empfangen. Pfeilverbindungen sind zugleich implizite Buffer. Ihnen können Anfangsbestände zugeordnet werden, ferner können Input-Output-Mengen vorbelegt werden. Standardmäßig sind die impliziten Buffer mit den Werten maximale Kapazität = 1, (Anfangs-)Bestand=0, Input-Menge=1 und Output-Menge=-1 vorbelegt. In der nachstehenden Matrix (vgl. Abb. 51) lassen sich die oben beschriebenen Werte individuell einstellen:

Buffer – Input-Output-Matrix		Time: 0	Model: <name of model>			
Task Buffer	Capacity	Automat 1	Automat 2	Automat 3	Automat 4	Automat 5
Buffer 1	2	2	-2	0	0	0
Buffer 2	1	0	1	0	-1	0
Buffer 3	1	0	0	1	-1	0
Buffer 4	2	0	0	0	2	-2
Buffer x	0	0	0	0	0	0

Abbildung 51: Aufnahmekapazität der impliziten Buffer sowie Input-Output-Beziehungen zu den Automaten

Die beispielhafte Abbildung der Input-Output-Beziehungen mit den entsprechenden Pfeilgewichten sieht wie folgt aus:

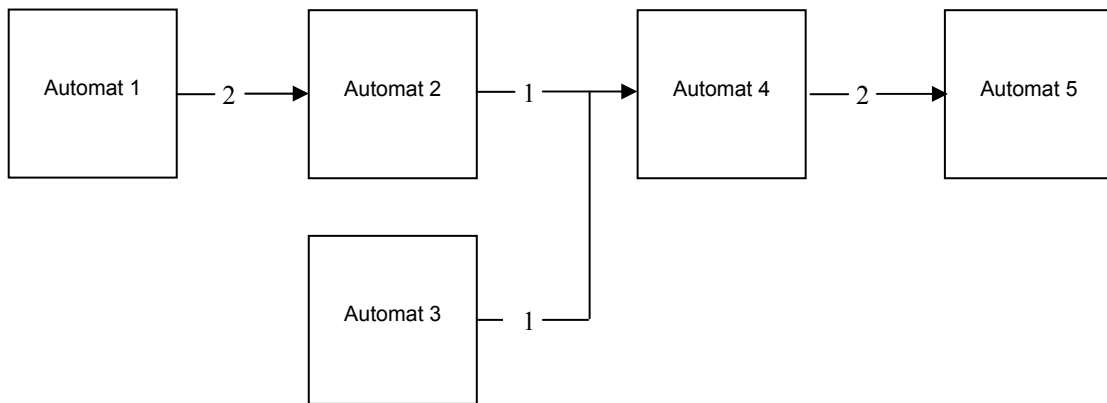


Abbildung 52: Beispielhafte Darstellung von Pfeilgewichten im Rahmen von Input-Output-Beziehungen zwischen Automaten

Positive Werte stellen den Input der Liefer-Task dar, negative den Wert, um den die startende Empfänger-Task den vorgelagerten impliziten Buffer dekrementiert. Obiges Beispiel stellt folglich Input-Gewichte dar, also den Wert, um den die Liefer-Tasks den impliziten nachgelagerten Buffer inkrementiert.

Neben den Input-Output-Werten (=Pfeilgewichte) müssen den impliziten Buffern weitere Parameter zugeordnet werden können. Dazu gehören Kapazitätsgrenzen und Wartezeiten. Mittels Kapazitätsgrenzen wird das maximale Fassungsvermögen der impliziten Buffer festgelegt. Zwei praktische Beispiele sollen dies verdeutlichen. In Produktionsprozessen existieren neben den klassischen, „expliziten“ Lagern sog. Zwischenlager, welche sich beim Übergang von einem Produktionsabschnitt zum nächsten wieder finden. Hier besteht eine begrenzte (Platz-)Kapazität, um Bauteile für eine begrenzte Zeit zu puffern. Dies ist u.a. in der Automobilindustrie der Fall. Trotz einer Just-In-Time Fertigung werden Sicherheitspuffer eingebaut, welche es den Automobilherstellern ermöglichen, die Produktion selbst bei verspäteter Anlieferung von Teilen noch für eine bestimmte Zeit aufrecht zu erhalten. Diese Puffer für Halbfertigprodukte befinden sich direkt vor bzw. bei der Maschine des jeweils nächsten Produktionsschritts. Es gibt hier also kein eigenes Lager.

In klinischen Abteilungen werden Gänge bzw. Vorräume zu den einzelnen Räumen häufig dafür benutzt, zu reinigende bzw. bereits gereinigte Betten für kurze Zeit abzustellen, also quasi „zwischenzulagern“. Auch hier ist die räumliche

Kapazität begrenzt. Für die o.g. Beispiele sind implizite Buffer die geeignetere grafische Darstellungsform als explizite Buffer, da es sich in den vorliegenden Beispielen nicht um Lagerräume im klassischen Sinn (z.B. eine Lagerhalle oder ein in einem separaten Raum befindliches Vorratslager) handelt.

Ein weiterer Parameter im Rahmen der Verknüpfung von Automaten sind Wartezeiten. Wartezeiten dienen der verzögerten Weitergabe von Transaktionen im Simulationsmodell. Werte  $n > 0$  stellen Wartezeiten dar. Nachfolgende Prozesse starten  $n$  Zeiteinheiten nach Beendigung des vorausgehenden Prozesses. Mittels dieses Parameters können z.B. Transportzeiten von Patienten nach erfolgter Behandlung von einem Behandlungsraum A zum nächsten Behandlungsraum B bzw. ins Patientenzimmer dargestellt werden (=implizite Form der Abbildung von Wartezeiten). Transportzeiten können alternativ auch durch einen Automaten dargestellt werden (=explizite Form der Abbildung von Wartezeiten).

### 4.1.3 Buffer

In Abgrenzung zu Kapitel 4.1.2<sup>101</sup> handelt es sich in Kapitel 4.1.3 um explizite Buffer. Diese bieten neben den impliziten Buffern zusätzliche funktionale Möglichkeiten bei der Modellierung von Prozessen. Explizite Buffer können von einem oder mehreren Automaten beliefert werden und an einen oder mehrere Automaten liefern. Dadurch ist eine Differenzierung der Input- und Outputmengen möglich. Abbildung 53 zeigt ein exemplarisches Beispiel: Buffer 1 wird vom Automaten 1 mit zwei Einheiten, vom Automaten 2 mit einer Einheit und vom Automaten 3 mit drei Einheiten eines Gutes beliefert. Er selbst beliefert wiederum den Automaten 4 mit einer Einheit und den Automaten 5 mit zwei Einheiten. Buffer 2 wird von Automat 4 mit einer Einheit beliefert.

---

<sup>101</sup> Beschreibung von impliziten Buffern in Form von Pfeilverbindungen zwischen Automaten.

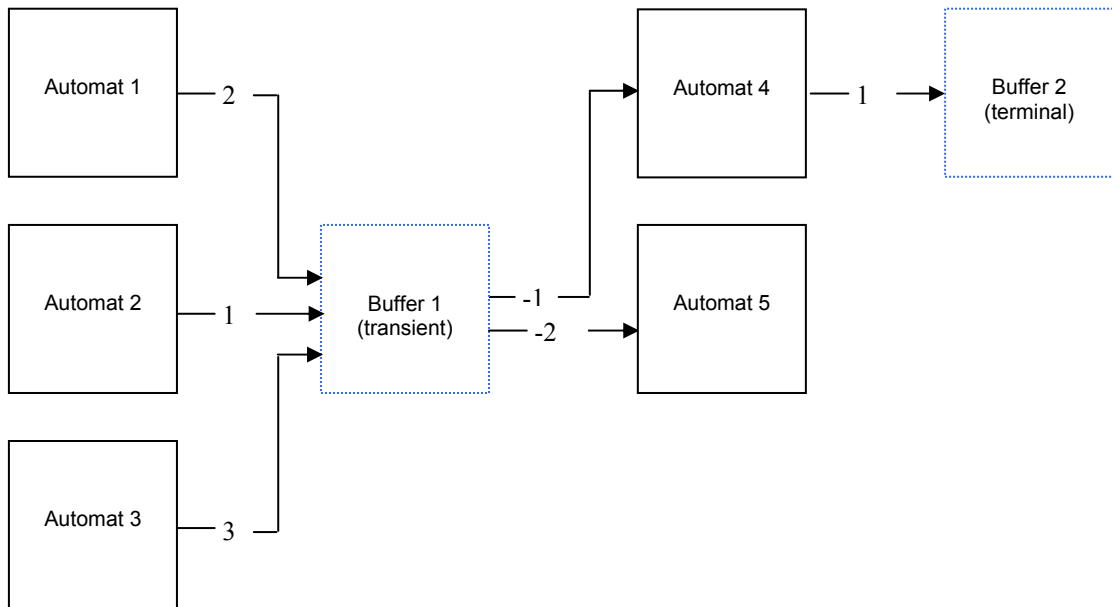


Abbildung 53: Beispielhafte Darstellung von Input-Output-Beziehungen zwischen Automaten und expliziten Buffern

Die Erfassung der Parameter erfolgt analog zu den impliziten Buffern (vgl. Abb. 54).

Buffer - Input-Output-Matrix		Time: 0					Model: <name of model>
Task Buffer	Capacity	Automat 1	Automat 2	Automat 3	Automat 4	Automat 5	
Buffer 1	10	2	1	3	-1	-2	
Buffer 2	3	0	0	0	1	0	

Abbildung 54: Aufnahmekapazität der expliziten Buffer sowie Input-Output-Beziehungen zu den Automaten

Explizite Buffer können initial, transient und terminal sein. Initiale Buffer haben keinen Eingang, aber ein bis mehrere Ausgänge. Transiente Buffer haben mindestens einen Eingang sowie mindestens einen Ausgang. Terminale Buffer haben mindestens einen Eingang, aber keinen Ausgang.

#### **4.1.4 Ressourcen**

Die Verfügbarkeit von Ressourcen ist Voraussetzung für die Ausführung von Aufgaben in Prozessen. Abwesenheit von Personal, z.B. durch Urlaub oder Krankheit, fehlendes Material oder Kapital sowie Stillstandszeiten von Maschinen durch Wartung oder Reparatur, haben Auswirkungen auf die Abarbeitung von im Prozessablauf anfallenden Aufgaben.

Um Prozesse unter Berücksichtigung notwendiger Ressourcen abbilden zu können und um die Ressourcenverfügbarkeit im Modell zu steuern, wird in den nachfolgenden Kapiteln für jede Ressourcengruppe eine individuelle Ressourcenverwaltung dargestellt und erläutert.

##### **4.1.4.1 Personal**

Personal wird mit Hilfe einer Personalmatrix den einzelnen Automaten zugeordnet. Dazu wird wie folgt vorgegangen: In der Personalmatrix werden alle am Prozess beteiligten Personen erfasst. Jede Person kann einem oder mehreren Automaten als notwendige Ressource zugeordnet sein. Personen können einer Personengruppe (z.B. Chirurg, Pflegepersonal, Verwaltung) zugeordnet werden. Bestimmte Personen aus einer Personengruppe können für bestimmte Prozessschritte unterschiedliche Präferenzen erhalten. Personen mit höherer Präferenz werden beim Einsatz für den entsprechenden Prozessschritt den Personen mit niedrigerer Präferenz vorgezogen. Personen und Personengruppen können in Arbeitsschichten eingeteilt werden. Dazu stehen Parameter zur Eingabe von Arbeitsschichten zur Verfügung. Ferner kann die Anzahl notwendiger Personen aus jeder Personengruppe für jeden einzelnen Prozessschritt angegeben werden. Dazu stehen vier Parameter zur Verfügung, die individuell belegt werden können (vgl. Abb. 55).

Personnel Matrix		Time: 0			Model: <name of model>	
Task	Aufnahme	Vorunter- suchung	Hauptunter- suchung	Behandlung	<name>	<name>
Person						
Maier	a120	-----	-----	-----	-----	-----
Müller	b110	-----	-----	-----	-----	-----
Schneider	a120	-----	-----	-----	-----	-----
Huber	-----	c110	-----	-----	-----	-----
Bauer	-----	c110	-----	-----	-----	-----
Meyer	-----	-----	c110	-----	-----	-----
Hübner	-----	-----	-----	e110	-----	-----
<name>	-----	-----	-----	-----	-----	-----

Abbildung 55: Personalmatrix<sup>102</sup>

Der erste Parameter (alphanumerisch) bezeichnet die Personengruppe (z.B. Chirurgen oder Pflegepersonal), der die betreffende Person angehört. Der zweite Parameter (numerisch) gibt die Präferenz der Person für die jeweilige Task an. Die Verfügbarkeitsprüfung erfolgt in der Reihenfolge der Präferenz. Der dritte Parameter gibt an, wie viele Personen aus der jeweiligen Personengruppe für die betreffende Task notwendig sind. Der vierte Parameter gibt die Arbeitsschicht an. Bezeichnungen für die Parameter Personengruppe und Arbeitsschicht können in der nachfolgend gezeigten Eingabemaske vergeben werden.

<sup>102</sup> Abbildung in Anlehnung an [Niem90 S.20].

Personnel Parameter – Groups and Shifts		Time: 0	Model: <name of model>
Personengruppe		Arbeitsschicht	
(1)	Chirurgen	(1)	Frühschicht
(2)	Pflegepersonal	(2)	Spätschicht
(3)	Kinderärzte	:	:
:	:	:	:
(n)	<name>	(n)	<name>

Abbildung 56: Eingabemaske für Personengruppe und Arbeitsschicht

Für die Personaleinsatzplanung stehen personenbezogene Kalender zur Verfügung. Hier können für jede einzelne Person oder Personengruppe An- und Abwesenheiten geplant werden. Zur Verfügung stehen Parameter für Urlaub (v=vacation), Krankheit (i=ill), Anwesenheit (p=present), Abwesenheit (a=absent) und Sonn-/Feiertage (h=holiday).

Personnel Calendar		Model: <name of model>									
Name: Huber / 5		Aktuelles Datum: 11. Nov. 2004									
Tag	M	D	M	D	F	S	S	.	.	.	D
Mon	1	2	3	4	5	6	7	.	.	.	31
Jan	p	p	p	p	p	h	h	.	.	.	p
Feb	p	p	p	p	p	h	h	.	.	.	p
Mar	p	p	p	p	p	h	h	.	.	.	p
.											
Dez	p	p	p	p	p	h	h	.	.	.	p

Abbildung 57: Personenbezogener Kalender<sup>103</sup>

Nach gleichem Schema ist ein übergeordneter betriebsbezogener Kalender verfügbar. Dort können z.B. Betriebsschließungstage eingetragen werden.

<sup>103</sup> Abbildung in Anlehnung an [Niem90 S.19].



Um Ressourcen in einem Modell realitätsgetreu abbilden zu können, sind noch weitere Parameter notwendig. Personen können u.U. mehrere Aufgaben gleichzeitig ausführen. In einem Produktionsbetrieb kann ein Mitarbeiter für die Überwachung von 1 bis n Maschinen zuständig sein. Ein Financial Controller kann für 1 bis n Abteilungen oder 1 bis n Länder bzw. Regionen verantwortlich sein. Um diese Fälle im Modell abzubilden, wird der Parameter CAPACITY eingeführt. Bei einer Kapazität von  $CAPACITY = 1$  kann die betreffende Person zu einem bestimmten Zeitpunkt nur einer Task zugeordnet werden. Bei einer Kapazität von  $CAPACITY = n$  kann die betreffende Person für n Tasks arbeiten bzw. zuständig sein.

Der Parameter EFFICIENCY legt die Leistungsfähigkeit einer Person fest [Peer98 S.107]. Folgendes Beispiel erläutert die Verwendung dieses Parameters: Für das Controlling der Abteilung „Innere Medizin“ einer Klinik benötigt ein erfahrener Senior Controller für das Erstellen der Kostenstellenberichte bei voller Leistungsfähigkeit ( $EFFICIENCY = 1$ ) vier Arbeitstage. Ein Junior Controller benötigt für die gleiche Aufgabe acht Arbeitstage. Um die Aufgabe in der gleichen Zeit zu erfüllen, wie ein Senior Controller, werden 2 Junior Controller benötigt. Sie haben in diesem Fall jeweils die halbe Leistungsfähigkeit eines Senior Controllers und damit eine  $EFFICIENCY = 0.5$ .

Personal kann fixe und variable Kosten verursachen. Letztere entstehen nach Umfang der Beauftragung von Personen. Im Modell entspricht dies der Zuordnung der Person zu einem Prozessschritt sowie dessen Abarbeitung. Als klassische Fixkosten sind Lohn- und Lohnnebenkosten zu nennen.

Die nachfolgende Abbildung zeigt die entsprechende Bildschirmmaske zur Erfassung und Nachverfolgung der oben beschriebenen Parameter.

Personnel Matrix				Time: 0	Model: <name of model>		
Task Person	Aufnahme	Vorunter- suchung	Hauptunter- suchung	Capacity	Occupancy	Fixed Cost	Variable Cost
	Maier	a120	-----	-----	1	0	150
Müller	b110	-----	-----	1	0	120	1.2
Schneider	a120	-----	-----	1	0	210	3.0
Huber	-----	c110	-----	1	0	90	5.8
Bauer	-----	c110	-----	1	0	70	0
Meyer	-----	-----	c110	1	0	145	0
Hübner	-----	-----	-----	1	0	64	0
<name>	-----	-----	-----	1	0	0	0

Abbildung 58: Erweiterte Personalmatrix

#### 4.1.4.2 Maschinen, Werkzeuge und Hilfsmittel

Der Aufbau der Bildschirmmaske zur Erfassung und Verwaltung der Parameter für Maschinen, Werkzeuge und Hilfsmittel (vgl. Abb. 59) - im Folgenden als Equipment bezeichnet - sowie die Wirkungsweise im Modell ist identisch zur Personalmatrix bzw. zur Ressource „Personal“. Auf das entsprechende Kapitel 4.1.4.1 wird verwiesen.

Equipment Matrix				Time: 0	Model: <name of model>	
Task Equipment	Aufnahme	Vorunter- suchung	Hauptunter- suchung	Behandlung	<name>	<name>
	Kantüle	a110	a110	-----	-----	-----
Behälter	a110	-----	a110	-----	-----	-----
Stethoskop	a110	-----	-----	-----	-----	-----
<name>	-----	-----	-----	-----	-----	-----

Abbildung 59: Equipment Matrix – Zuordnung des Equipments zu den einzelnen Automaten

#### 4.1.4.3 Material

Der Verbrauch von Material, darin eingeschlossen Hilfs- und Betriebsstoffe, kann mit Hilfe der nachfolgenden Eingabemaske gesteuert werden. Jeder Task wird das notwendige Material und die Verbrauchsmenge zugeordnet. In der nachfolgenden Abbildung ist dies durch negative Werte dargestellt. Erzeugt ein Automat Material und erhöht damit den Materialbestand, wird dies durch Werte > 0 dargestellt.

Material Matrix		Time: 0			Model: <name of model>	
Task / Material	Aufnahme	Vorunter- suchung	Hauptunter- suchung	Behandlung	<name>	<name>
Beta-Blocker	-2.00	-1.00	0.00	0.00	0.00	0.00
Aspirin	0.00	0.00	-1.00	0.00	0.00	0.00
Kochsalzlg.	0.00	0.00	0.00	-2.70	0.00	0.00
Verbandsmat.	0.00	0.00	0.00	-5.00	0.00	0.00
Einwegtuch	0.00	0.00	-1.00	-1.00	0.00	0.00
<name>	0.00	0.00	0.00	0.00	0.00	0.00

Abbildung 60: Materialmatrix zur Steuerung des Verbrauchs von Material bzw. zur Herstellung eines Gutes

Weitere Parameter für Material oder Materialgruppen sind der Preis (PRICE), ein Anfangsbestand (SUPPLY), welcher dann gemäß den angegebenen Verbrauchsvorschriften reduziert bzw. bei Produktion des Gutes erhöht wird, sowie ein Mindestlagerbestand. Bei Erreichen bzw. Unterschreiten des Mindestlagerbestands kann dann automatisch eine Bestellung ausgelöst werden (ORDER POINT). Dies lässt sich durch einen Automaten realisieren, der in jedem Zeitschritt prüft, ob der Mindestbestand erreicht oder unterschritten wurde (SUPPLY <= ORDER POINT). Er löst dann gegebenenfalls eine Bestellung aus. Um festzulegen, zu welchem Zeitpunkt das Material verbraucht wird, stehen die Werte FIRST, LAST und ALL zur Verfügung. FIRST bedeutet, dass das Material nur im ersten Zeitschritt - direkt nach dem Start des Automaten - verbraucht wird. LAST bedeutet, dass das Material im letzten Zeitschritt des Automaten verbraucht wird. Mit ALL wird Material in jedem Zeitschritt, den der Automat

durchläuft, verbraucht. Die Abbildung 62 zeigt die entsprechende Benutzeroberfläche zur Parametrisierung.

Material Matrix		Time: 0			Model: <name of model>		
Task Material	Aufnahme	Vorunter- suchung	Hauptunter- suchung	Behandlung	Supply	Price	Order Point
Beta-Blocker	----	First	----	----	100.00	2.00	25.00
Aspirin	----	----	First	----	100.00	0.75	20.00
Kochsalzlg.	----	----	----	All	100.00	3.60	40.00
Verbandsmat.	----	----	----	Last	100.00	0.45	80.00
Einwegtuch	----	----	First	First	100.00	0.80	20.00
<name>	----	----	----	----	0.00	0.00	0.00

Abbildung 61: Materialmatrix mit Angaben zum Verbrauchszeitpunkt, Materialpreis, Bestandsmenge und Bestellzeitpunkte

#### 4.1.4 Transaktionen

Transaktionen beschreiben in der vorliegenden Arbeit parametrisierbare, mobile Modellkomponenten. Übertragen auf das klinische Umfeld stellen Transaktionen im Kontext der vorliegenden Arbeit Patienten mit bestimmten Krankheitsbildern dar. Die Bezeichnung der Transaktion erfolgt durch Vergabe eines Transaktionsnamens/-bezeichners. Für Transaktionen, die im Modell Patienten repräsentieren, kann dies z.B. eine eindeutige Patientenummer sein. So wird zum einen die Verfolgung und Identifizierung von Transaktionen während des Modelllaufs, zum anderen die Erstellung statistischer Auswertungen erleichtert.

Transaktionen durchlaufen Prozesse nach bestimmten Regeln und Vorschriften. Den Parameter, der festlegt, welchen Weg die jeweilige Transaktion durch das Modellnetz nimmt, wird NEXT PATH genannt. Dieser Parameter wird mit der Pfeilnummer des im Netz festgelegten nachfolgenden Pfeils belegt. Durch individuelle Belegung der Pfeilnummer durch den Anwender kann bei Bedarf der Weg der Transaktion durch das Netz unabhängig von der Netzstruktur geleitet

werden. Es entsteht damit eine hohe Flexibilität bei der Verarbeitung von Transaktionen.

In Modellen werden Transaktionen standardmäßig i.d.R. nach dem FIFO-Prinzip (first-in first-out) abgearbeitet. Die Transaktion, die als erstes eintrifft, wird auch als erste weiterverarbeitet. Dieses Prinzip kann durch Vergabe von Prioritäten (Parameter PRIORITY) verändert werden. Gleiches erreicht man durch Verwendung entsprechender Warteschlangen (z.B. LIFO-Warteschlangen<sup>104</sup>, FIFO-Warteschlangen).

Der Parameter DEADLINE gibt den spätest möglichen Fertigstellungstermin des Auftrages an. Er dient statischen Zwecken und erleichtert spätere Analysen von Transaktionen.

Bei Aufträgen, die eine Mindestlagerzeit / Mindestwartezeit haben, wird der Parameter MIN\_STORING\_TIME belegt. Bei Angabe einer MIN\_STORING\_TIME > 0 kann die betreffende Transaktion erst dann aus dem Buffer entnommen und an die nächste Task weitergeleitet werden, wenn die Mindestwartezeit erreicht ist.

Die Zeiten, zu denen ein Automat zur Bearbeitung anstehende Transaktionen nicht annehmen kann, weil er z.B. aufgrund eines Ressourcenengpasses nicht anlaufen kann, sollten automatisch mitprotokolliert werden. Diesem Zweck soll der Parameter BLOCKED\_TIME dienen.

Dem Anwender sollten neben den o.g. vorgegebenen Parametern weitere Parameter zur freien Belegung zur Verfügung stehen (USER\_PARAM\_i). Diese sollten je nach Anwendungsfall individuell belegt und für spätere statistische Auswertungen genutzt werden können.

Abbildung 62 zeigt die Bildschirmmaske, welche die oben beschriebenen Parameter widerspiegelt.

---

<sup>104</sup> LIFO = last-in, first-out.

<b>Transactions</b>		<b>Time: 0</b>		<b>Model: &lt;name of model&gt;</b>		
Name	Next Path	Priority	Deadline	Min_Storing_Time	$\Sigma$ Blocked_Time	User_Param_i
Transaktion 1	3	0	0	0	6	0
Transaktion 2	1	1	0	20	0	0
Transaktion 3	5	0	120	0	0	0
Transaktion x	0	0	0	0	0	0

Abbildung 62: Transaktionsmatrix mit zugehörigen Parametern [Niem90a S.150]

Zur Verarbeitung von Transaktionen können fünf verschiedene Automatentypen zum Einsatz kommen (vgl. Abb. 63) [Niem90a S.132f und S.159; vgl. auch Peer98 S.116ff].

- (1) Der GENERATE-Automat erzeugt Transaktionen und gibt diese an den nachfolgenden Automaten bzw. an den nachfolgenden Buffer weiter.
- (2) Der STANDARD-Automat ist ein zeitverbrauchender Automat. Transaktionen werden nach der Bearbeitungsdauer weitergegeben.
- (3) Ein ASSEMBLY-Automat fasst mehrere Transaktionen zu einer einzigen zusammen. In Produktionsprozessen kann er eingesetzt werden, wenn mehrere Bauteile aus vorgelagerten Prozessschritten zu einem Teil zusammgebaut werden. In klinischen Prozessen können mittels eines Assembly-Automaten z.B. einzelne Informationen aus unterschiedlichen Behandlungsschritten in einem Arztbrief zu einer Gesamtinformation zusammengefasst werden.
- (4) Der SPLIT-Automat verhält sich umgekehrt zum Assembly-Automaten. Er dupliziert eine Transaktion gemäß den im Modell festgelegten Output-Beziehungen. Der Split-Automat kann im klinischen Prozess für die differenzierte Betrachtung der Behandlung eines Patienten sowie dem dazugehörigen Informations-/Datenfluss eingesetzt werden.
- (5) Der ASSEMBLY/SPLIT-Automat kombiniert die Eigenschaften und Wirkungen der oben beschriebenen gleichnamigen Automatentypen.

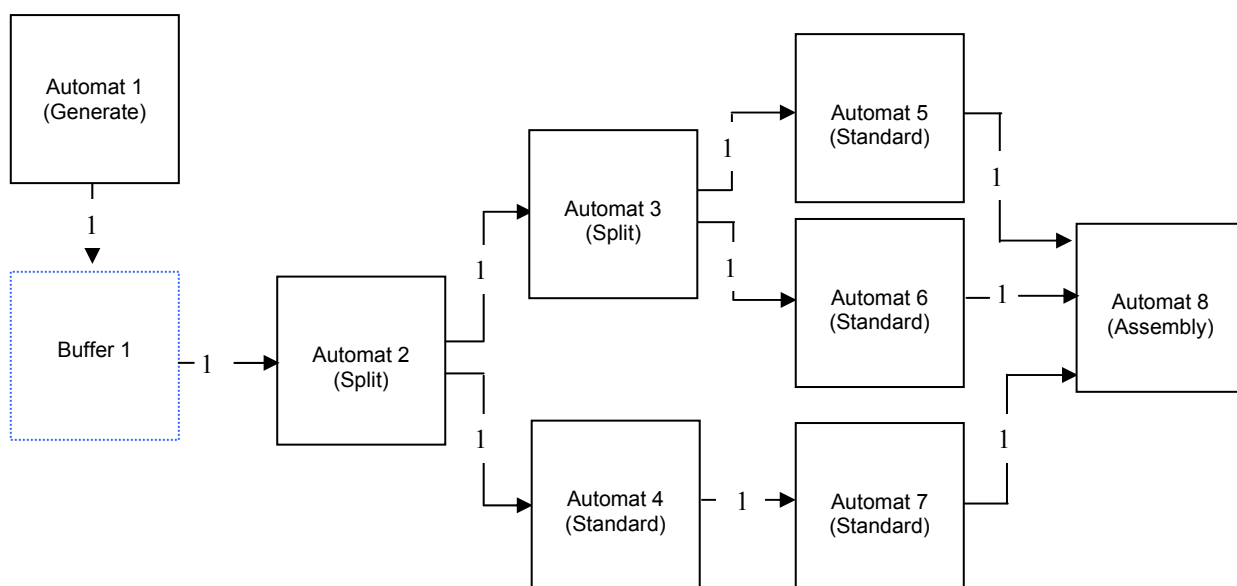


Abbildung 63: Automatentypen Generate, Standard, Split und Assembly

Um im Verlauf der Prozesssimulation zeitbezogene „Verhaltensmuster“ für jede Transaktion festzuhalten, muss der Anwender in die Lage versetzt werden, entsprechende Parameter für spätere Auswertungen bereits bei der Modellierung individuell zu definieren. Abbildung 64 zeigt beispielhaft eine Bildschirmmaske zur Eingabe User-individueller, zeitbezogener Parameter.

Time-related individual User Parameter		Model: <name of model>
Name	Start of Measurement (Task / Buffer)	End of Measurement (Task / Buffer)
Behandlungsdauer	Patientenaufnahme	Patientenentlassung
Door-to-Balloon	Notaufnahme	Dilatation
Cut-Suture-Time	Cutting	Suturing
Door-to-Needle	Notaufnahme	Lysetherapie
<User Parameter z>	<name>	<name>

Abbildung 64: Maske zum Festlegen zeitbezogener, User-individueller Parameter

Für jeden Parameter wird neben dem Namen der gewünschte Start- und Endzeitpunkt für die Zeitmessung vom Anwender festgelegt. Start- und

Endzeitpunkte der Messung werden durch Eingabe des Buffers oder des Automaten bestimmt. Trifft die Transaktion auf den Buffer bzw. Automaten, beginnt bzw. endet die Zeitmessung.

#### **4.2 Basiskomponenten zur Darstellung und Analyse entscheidungsrelevanter Informationen**

Die Entwicklung eines adäquaten Berichtswesens bedingt vorab die Klärung diverser Fragestellungen, die sich im Zusammenhang mit der Bereitstellung von Informationen ergeben. Um den Umfang dieser Thematik einzugrenzen, konzentrieren sich die nachfolgenden Betrachtungen auf die Anforderungen des Top- und Middle-Managements in Kliniken sowie auf die klinischen Kernprozesse. Aus dieser Einschränkung ergeben sich als Berichtsempfänger in Kliniken die Klinikleitung, die Professoren, die leitenden Oberärzte sowie die Pflegedienstleitung.

Jeder Berichtsempfänger bzw. jede Empfängergruppe verfolgt spezifische Interessen und setzt damit unterschiedliche Schwerpunkte in Bezug auf relevante Informationen. Inhalt, Aggregationsgrad und Aufbereitung von Informationen müssen deshalb auf die jeweiligen Empfänger abgestimmt sein. Generell gilt: Je höher die hierarchische Position des Berichtsempfängers im Unternehmen, desto aggregierter sind die bereitgestellten Informationen. Vice versa: Je niedriger die Position in der hierarchischen Unternehmensstruktur, desto spezifischer und granularer sind die benötigten Informationen. Informationen sind hier z.B. kostenstellen- oder produktbezogen.

Das Top-Management, also die Klinikleitung, sollte zur Steuerung der Klinik u.a. die Umsätze, Kosten und Gewinne der einzelnen klinischen Bereiche, die durchschnittlichen Patientenverweildauern, die Bettenbelegung, den Grad der Einhaltung von Leitlinien, Patientenzufriedenheit, Mitarbeiterzufriedenheit und Innovationsgrad erhalten. Die Leiter der klinischen Abteilungen, weiter oben als Middle-Management bezeichnet, benötigen ebenfalls die o.g. Informationen, jedoch bezogen auf ihren eigenen Tätigkeitsbereich, z.B. der Kardiologie, Radiologie, Onkologie etc. Vertieft man die Bereitstellung



entscheidungsrelevanter Informationen am Beispiel der Kardiologie, ergeben sich weitere Kennzahlen. Diese sind z.B. die Geräteauslastung in der Abteilung, Engpässe während der Behandlung<sup>105</sup>, der „Case Mix“ und die damit verbundenen „Cost per Case“, Door-to-Balloon-Time<sup>106</sup>, Door-to-Needle-Time, Revaskularisierungsgrad (TIMI-Fluss)<sup>107</sup>, Entlassungsmedikation und Infektionsraten.

Trotz einiger aus Gründen der Entscheidungs- und Steuerungsrelevanz zwingend notwendiger Kennzahlen, gibt es von Klinik zu Klinik selbstverständlich individuelle Unterschiede im Berichtswesen. Diese müssen in der Praxis von Fall zu Fall geklärt werden und können somit nicht als Standard in ein Simulationswerkzeug einfließen.

Der Grad der Verdichtung von Daten für unterschiedliche Berichtsempfänger ist in Abbildung 65 schematisch dargestellt.

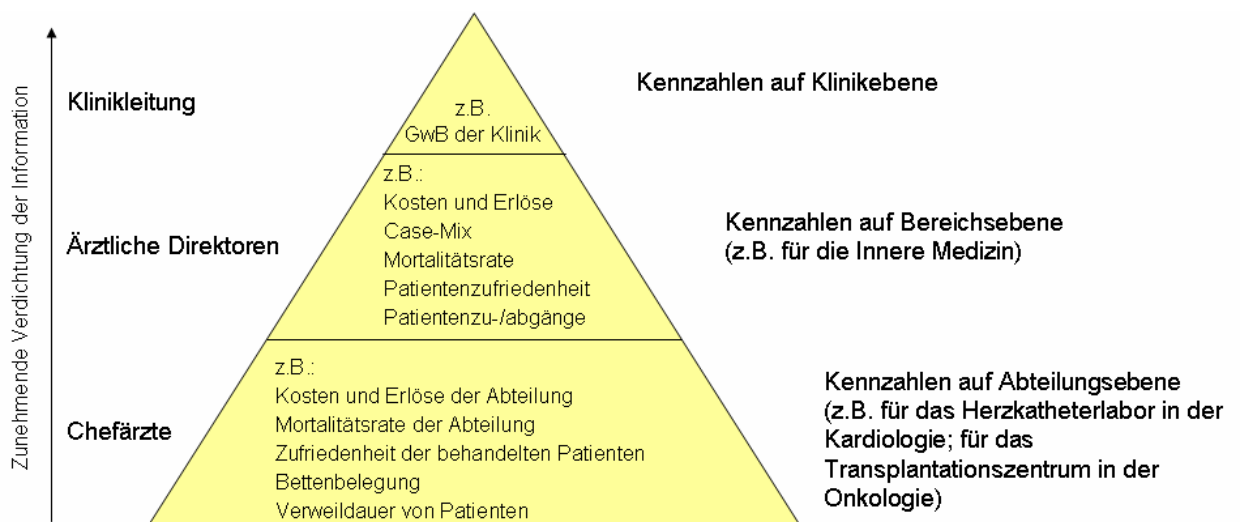


Abbildung 65: Schematische Darstellung der adressatenabhängigen Verdichtung von Daten

Zur detaillierten Analyse des Modellverhaltens müssen Statistiken zu jedem Zeitpunkt während des Simulationslaufs mit den jeweils aktuellen Werten angezeigt werden können. Dabei sollte es dem Anwender möglich sein, Daten

<sup>105</sup> Z.B. aufgrund von Personalmangel oder Ausfall von Geräten.

<sup>106</sup> Zeitspanne zwischen dem Eintreffen des Patienten in der Notaufnahme und dem Wiedereröffnen des Infarktgefäßes.

<sup>107</sup> Revaskularisation = Chirurgische Verbesserung der Durchblutung minderversorgter Gewebe durch Gefäßchirurgie.

unterschiedlich stark zu verdichten. In einer hierarchischen Modellstruktur geschieht dies bottom-up.

In den nachfolgenden Kapiteln werden die bisherigen Erkenntnisse bei der Entwicklung von standardisierten Auswertungen, die ein Simulationssystem bereitstellen sollte, einfließen.

## **4.2.1 Ressourcenbezogene Statistiken**

### **4.2.1.1 Personal**

Als erstes wird auf das Thema der Mitarbeiterauslastung eingegangen. Jeder Betrieb mit Gewinnerzielungsabsicht ist daran interessiert, seine Mitarbeiter während ihrer Arbeitszeit möglichst zu 100% auszulasten, entsprechende Ruhepausen eingerechnet. Bei Unterauslastung arbeitet der Betrieb ineffizient. Andererseits riskiert er bei übermäßig hoher Auslastung die verspätete Fertigstellung von Aufträgen aufgrund von Kapazitätsengpässen. Die Mitarbeiterzufriedenheit kann bei extremer Arbeitsbelastung<sup>108</sup> des Personals sinken. In der klinischen Praxis ist dies ein Balance-Akt: Kliniken können ihren „Auftragseingang“ i.d.R. nur bedingt steuern. Notfallpatienten, die z.B. mit einem Herzinfarkt eingeliefert werden, können nicht erst in den nächsten Stunden oder gar Tagen behandelt werden. Sie müssen umgehend therapiert werden. Überstunden des Klinikpersonals gehören deshalb zum Alltag in Krankenhäusern. Würde auf der anderen Seite zu viel Personal vorgehalten, sinkt die Rentabilität der betreffenden Klinik. Deshalb ist eine solide, möglichst flexible und dem Bedarf angepasste Personalplanung entscheidend. Die in den nachfolgenden Abbildungen gezeigten Statistiken können hierfür relevante Information bereitstellen.

Das Aktivitätsverteilungsdiagramm in Abbildung 66 zeigt, wie viel Prozent des Personals während der Arbeitszeit zu wie viel Prozent aktiv waren, also gemäß ihren Skills und ihrer Aufgaben im Prozessablauf aktiv eingesetzt waren. Je besser die Ergebnisse, desto weiter befinden sie sich im linken Bereich der x-

---

<sup>108</sup> Z.B. häufige Überstunden; Wochenendarbeit; Zusatzschichten.

Achse (Activity in %) und im oberen Bereich der y-Achse (% of Personnel) in der Grafik.

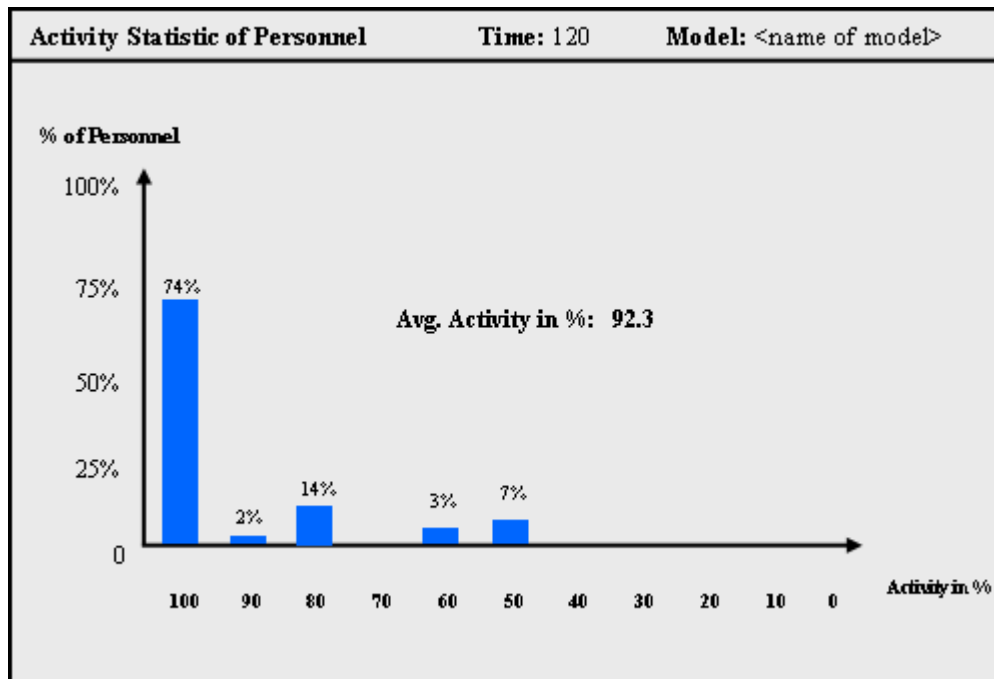


Abbildung 66: Aktivitätsverteilungsdiagramm für das Personal

Die Werte aus Abbildung 66 können auch in Anlehnung an eine Lorenzkurve<sup>109</sup> dargestellt werden. Die nachstehende Grafik zeigt Konzentrationstendenzen bezogen auf die Auslastung des Personals.

<sup>109</sup> Siehe dazu [BaBa89 S.24f; SBR05 S.33f].

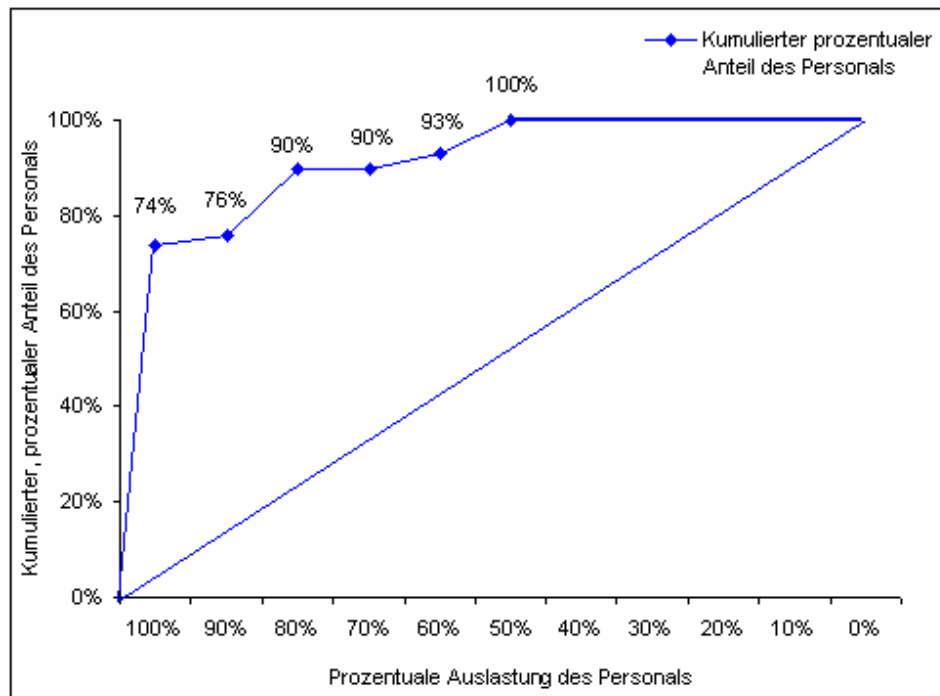


Abbildung 67: Gegenüberstellung des kumulierten, prozentualen Anteils des Personals und der zugehörigen prozentualen Auslastung

Abbildung 68 zeigt eine detaillierte personenbezogene Auswertung. Hier wird die kumulierte aktive Zeit absolut (Active Time) und in Prozent der Gesamtzeit berechnet, ferner die aufgelaufenen fixen und variablen Personalkosten (VarCost und FixCost). Für jeden Mitarbeiter wird zudem angezeigt, wie oft er für einen Engpass im Prozessablauf (Bottleneck) und damit für eine Prozessunterbrechung verantwortlich war.

<b>Personnel Results</b>						
Name	Active Time	Active Time in %	Bottleneck	Var.Cost	Fix Cost	Total Cost
Maier	55	46%	2	120	1.400	1.520
Müller	32	27%	0	80	990	1.070
Schneider	100	83%	1	56	2.090	2.146
Huber	12	10%	0	112	1.200	1.312
<b>TOTAL</b>	<b>199</b>	<b>51.5%</b>	<b>3</b>	<b>368</b>	<b>5.680</b>	<b>6.048</b>

Abbildung 68: Personenbezogene Auswertung bezüglich aktiver Zeiten, Engpässe und Kosten

Gibt es mehrere Mitarbeiter mit gleichem Skillset, können diese zu Personengruppen zusammengefasst werden. Die Statistik in Abbildung 69 zeigt, welche Personengruppe durch mangelnde Verfügbarkeit Engpässe im Prozess ausgelöst hat und wie lange die durchschnittliche Unterbrechungsdauer betrug. Die Statistik in Abbildung 70 zeigt das Beschäftigungsverhalten der jeweiligen Personengruppe.

Personnel Group Results – Bottleneck Statistic								
Task	Notnahme		Voruntersuchung		Hauptuntersuchung		Behandlung	
	No.	Average Duration	No.	Average Duration	No.	Average Duration	No.	Average Duration
Chirurg	0	0	0	0	0	0	3	8
Kardiologe	0	0	4	9	1	5	0	0
Anästhesist	0	0	0	0	0	0	1	3
Pfleger	0	0	0	0	0	0	0	0
Krankenschwester	2	18	1	5	0	0	0	0
<b>TOTAL</b>	<b>2</b>	<b>18</b>	<b>5</b>	<b>8</b>	<b>1</b>	<b>5</b>	<b>4</b>	<b>6</b>

Abbildung 69: Prozessunterbrechungen durch mangelnde Verfügbarkeit von Personal einer bestimmten Personengruppe

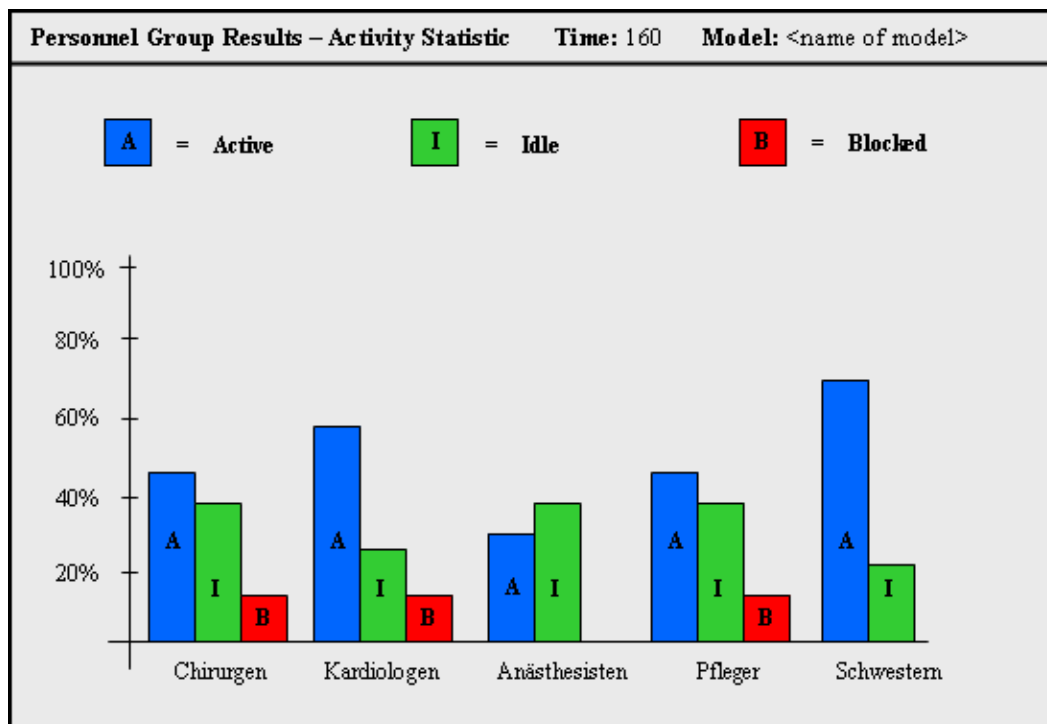


Abbildung 70: Beschäftigungsverhalten einzelner Personengruppen

Die Abbildungen 71 und 72 dienen der detaillierten Beobachtung und Analyse von Personalengpässen je Prozessschritt und je Person während und nach Abschluss des Simulationslaufs. Die Zustandsvariablen active, blocked und idle

dienen zur Beschreibung des Zustandes für jedes Matrixfeld (x, y) und in jedem Zeitschritt t. Abbildung 72 zeigt das Einsatzverhalten einer bestimmten Person im Zeitverlauf der Simulation.

Beide Darstellungen eignen sich aufgrund ihrer Granularität besonders zum „Fine-Tuning“ von Prozessen.

Personnel Activity		Time: 120		Model: <name of model>		
Task Person	Aufnahme	Vorunter- suchung	Hauptunter- suchung	Behandlung	<name>	<name>
	Maier	idle	idle	idle	active	idle
Müller	idle	blocked	active	idle	idle	idle
Schneider	active	idle	idle	idle	idle	idle
Huber	idle	idle	idle	busy	idle	idle
<name>	idle	idle	idle	idle	idle	idle

Abbildung 71: Matrix zur Feinanalyse während des Simulationslaufs<sup>110</sup>

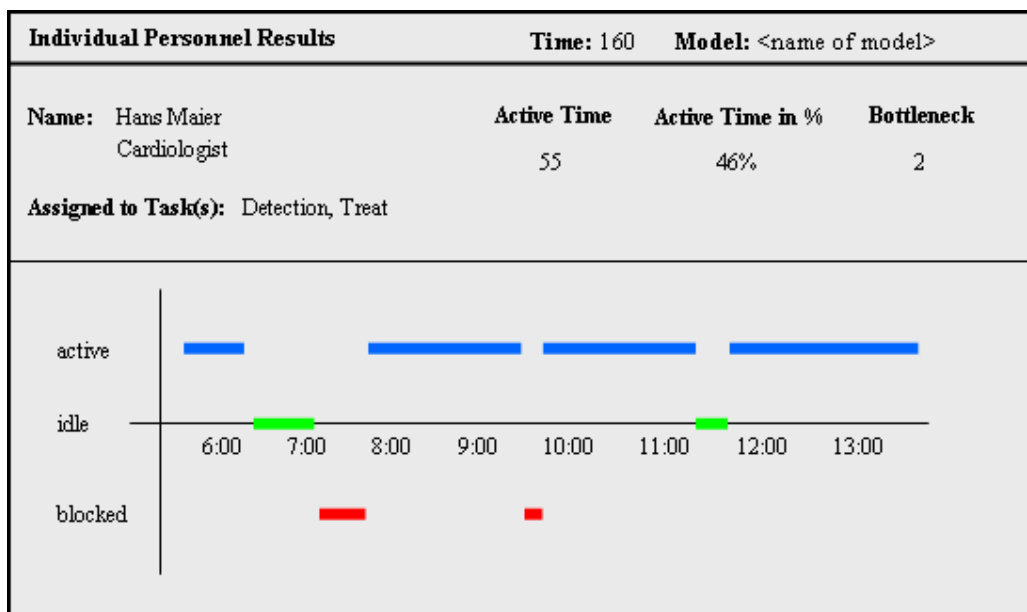


Abbildung 72: Matrix zur Feinanalyse des Einsatzverhaltens einer bestimmten Person

<sup>110</sup> Abbildung in Anlehnung an [Niem90 S.32].

### 4.2.1.2 Maschinen, Werkzeuge und Hilfsmittel

Analog zu den personenbezogenen Auswertungen wurden equipment-bezogene Statistiken entwickelt. Auf die entsprechenden Erläuterungen im Kapitel 4.2.1.1 wird verwiesen. Die nachstehenden Abbildungen zeigen die Bildschirmmasken, die analog zu den Masken, welche im vorherigen Kapitel gezeigt wurden, aufgebaut sind.

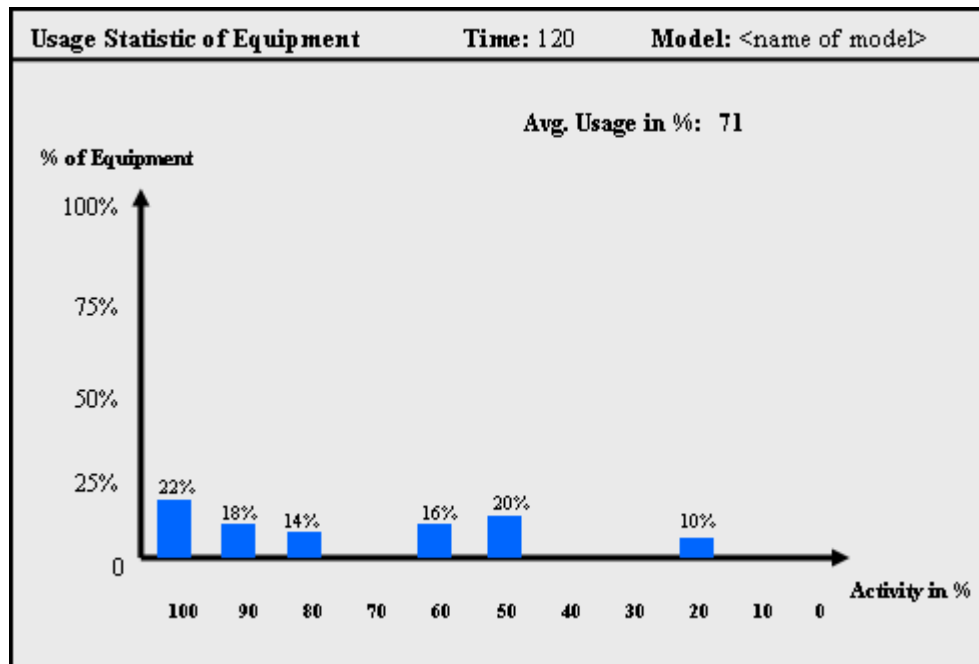


Abbildung 73: Aktivitätsverteilungsdiagramm für das Equipment

Equipment Results		Time: 120	Model: <name of model>			
Equipment Name	Active Time	Active Time in %	Bottleneck	Var.Cost	Fix Cost	Total Cost
MR	55	46%	2	120	1.400	1.520
CT	32	27%	0	80	990	1.070
Angio	100	83%	1	56	2.090	2.146
<b>TOTAL</b>	<b>187</b>	<b>51.5%</b>	<b>3</b>	<b>256</b>	<b>4.480</b>	<b>4.736</b>

Abbildung 74: Equipmentbezogene Auswertung bezüglich aktiver Zeiten, Ressourcenengpässe und Kosten



Equipment Results – Bottleneck Statistic								
Task Equipment	Notaufnahme		Voruntersuchung		Hauptuntersuchung		Behandlung	
	No.	Average Duration	No.	Average Duration	No.	Average Duration	No.	Average Duration
CT	0	0	0	0	0	0	3	8
MR	0	0	4	9	1	5	0	0
Angio	0	0	0	0	0	0	1	3
Sonogr.	0	0	0	0	0	0	0	0
TOTAL	2	18	5	8	1	5	4	6

Abbildung 75: Prozessunterbrechungen durch mangelnde Verfügbarkeit von Equipment

#### 4.2.1.3 Material

Auswertungen zum Materialeinsatz sind analog zu den Auswertungen, die in den beiden vorherigen Kapiteln 4.2.1.1 und 4.2.1.2 dargestellt sind. Auf die dortigen Ausführungen wird verwiesen.

Neu ist die Matrix zum Materialverbrauch. Sie zeigt für jeden einzelnen Prozessschritt den bis zum gewählten Zeitpunkt aufgelaufenen Verbrauch des betreffenden Materials (vgl. Abb. 76).

Die Abbildungen 77 und 78 zeigen Beispiele von materialbezogenen Auswertungen, die analog zu den Bildschirmmasken in den beiden vorherigen Kapiteln aufgebaut sind.

<b>Material Consumption</b>		<b>Time: 120</b>		<b>Model: &lt;name of model&gt;</b>		
<i>Task</i> <i>Material</i>	Aufnahme	Vorunter- suchung	Hauptunter- suchung	Behandlung	<name>	<name>
	Beta-Blocker	-183	0	0	-23	0
Verbandsmat.	0	0	-14	-56	0	0
Einwegtücher	0	0	0	-32	0	0
<name>	0	0	0	0	0	0

Abbildung 76: Materialverbrauch je Prozessabschnitt

<b>Material Results</b>		<b>Time: 120</b>		<b>Model: &lt;name of model&gt;</b>		
Material	Active Time	Active Time in %	Bottleneck	Var.Cost	Fix Cost	Total Cost
Verbandsmat.	55	46%	2	120	1.400	1.520
Beta-Blocker	32	27%	0	80	990	1.070
<b>TOTAL</b>	<b>87</b>	<b>21.5%</b>	<b>2</b>	<b>200</b>	<b>1.390</b>	<b>2.590</b>

Abbildung 77: Materialbezogene Auswertung bezüglich aktiver Zeiten, Ressourcenengpässe und Kosten

Material Results – Bottleneck Statistic								
Task	Notnahme		Voruntersuchung		Hauptuntersuchung		Behandlung	
	No.	Average Duration	No.	Average Duration	No.	Average Duration	No.	Average Duration
Beta-Blocker	2	18	0	0	0	0	3	8
Verbandsmat.	0	0	4	9	1	5	0	0
TOTAL	2	18	4	9	1	5	3	8

Abbildung 78: Prozessunterbrechungen durch mangelnde Verfügbarkeit von Material

#### 4.2.1.4 Ressourcenübergreifende Statistiken

In den vorangegangenen Kapiteln 4.2.1.1 bis 4.2.1.3 wurden Statistiken dargestellt, die sich auf eine bestimmte Ressource bzw. Ressourcenart bezogen haben. Die folgenden Statistiken betrachten nun mehrere Ressourcenarten gleichzeitig. Um eine möglichst umfassende Sicht zu bekommen, wurde in den Darstellungen zusätzlich zu den Ressourcen die Buffer mit einbezogen.

Abbildung 79 zeigt für jedes mögliche Tupel die im Zeitverlauf prozentual aufgetretenen Engpasssituationen. Die Werte sind über die Zeit kumuliert. Aus den vier Perspektiven Personal, Material, Equipment und Buffer ergeben sich insgesamt 15 Tupel. Es gilt:

(1)  $P := \text{Personal}, M := \text{Material}, E := \text{Equipment}, B := \text{Buffer}$

(2)  $M = \{k \in n\text{-Tupel für } n = 1, \dots, 4 \mid n\text{-Tupel aus } \{P, M, E, B\}\}$

(3)  $S = \{x \in M \text{ mit der Eigenschaft: Nachgefragte Menge von } x_{t+1} > \text{Bestandsmenge von } x_t\}$

$$(4) \frac{\sum_{t=0}^n \sum_{y \in S} y}{\sum_{t=0}^n \sum_{b \in M} b} = \text{Prozentualer Anteil verursachter Engpässe durch das betrachtete } n\text{-Tupel}$$

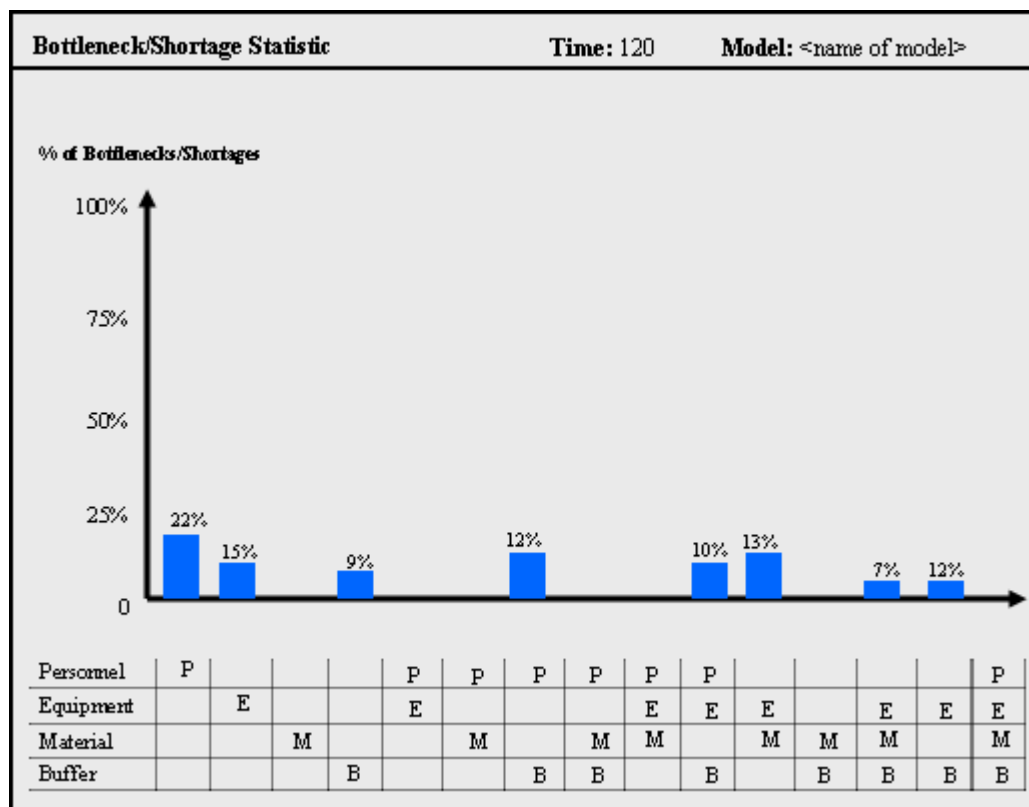


Abbildung 79: Prozentual aufgetretene Engpässe und dafür verantwortliche Ursachen<sup>111</sup>

Die in Abbildung 80 gezeigte Statistik gibt Auskunft darüber, wie oft und wodurch ein Engpass verursacht wurde und an welcher Stelle im Prozess der Engpass aufgetreten ist. Die Matrix zeigt die Situation im jeweils aktuellen Zeitschritt sowie die über die Zeit kumulierten Werte.

<sup>111</sup> Abbildung nach [Brin87 S.123].

<b>Bottleneck/Shortages</b>		<b>Time: 120</b>		<b>Model: &lt;name of model&gt;</b>		
		<b>Actual Status</b>				
<i>Task</i>	<i>Bottleneck</i>	Aufnahme	Vorunter- suchung	Hauptunter- suchung	Behandlung	<name>    <name>
Personnel		0	1	0	1	
Equipment		0	0	1	1	
Material		0	0	0	0	
Buffer		0	0	0	1	
		<b>Total / Cumulated</b>				
<i>Task</i>	<i>Bottleneck</i>	Aufnahme	Vorunter- suchung	Hauptunter- suchung	Behandlung	<name>    <name>
Personnel		0	4	0	3	
Equipment		0	0	7	4	
Material		0	0	0	0	
Buffer		0	0	0	5	

Abbildung 80: Auslöser für Engpässe je Prozessschritt

#### 4.2.2 Automatenbezogene Statistiken

Analog zu den ressourcenbezogenen Statistiken zeigt Abbildung 81 für Automaten eines Modells die kumulierte Aktivitätsverteilung im Zeitablauf.

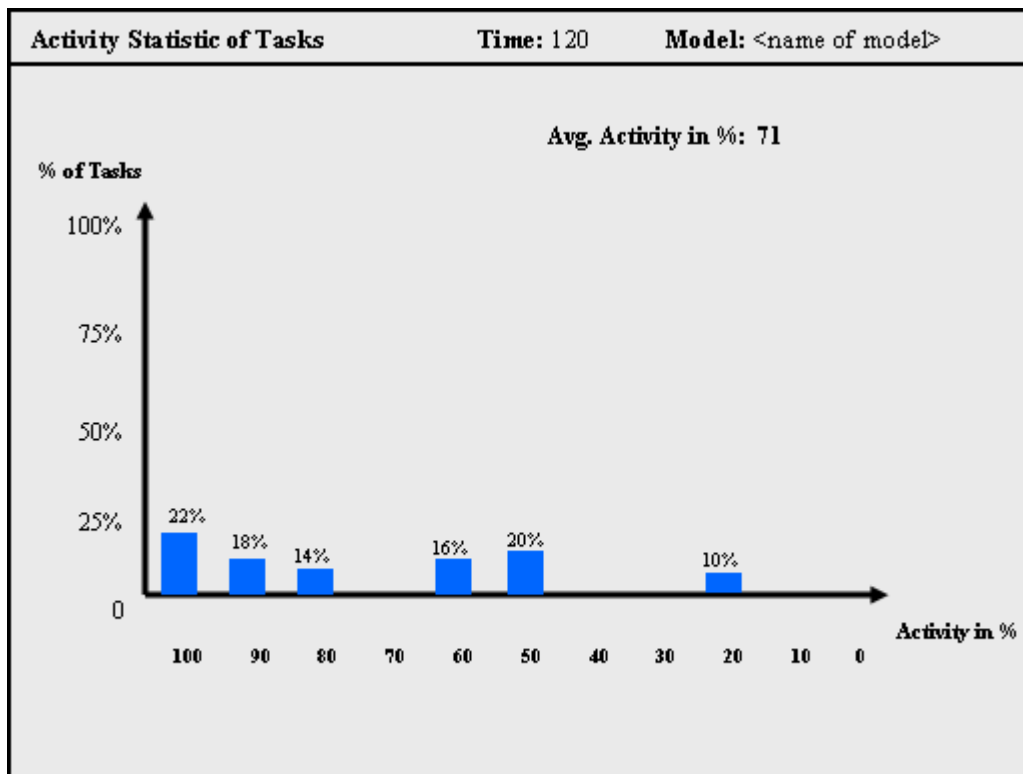


Abbildung 81: Aktivitätsverteilungdiagramm der Automaten

Um Informationen zum Aktivitätsverhalten der einzelnen Automaten zu erhalten, können zusätzlich zum Aktivitätsverteilungdiagramm die jeweiligen Automaten und deren Aktivitätsprofil angezeigt werden. Dies ist in Abbildung 82 dargestellt. Die Skala für „Activity in %“ wurde im vorliegenden Beispiel in 10er-Schritte unterteilt.

Activity Statistic of Tasks - Details		Time: 120	Model: <name of model>
Task	Activity in %	Activity in %	
	100%	50%	
EKG	90%	40%	
CT, MRT	80%	Ultraschall	30%
	70%	20%	
Laboruntersuchung	60%	Mammographie	10%
		0%	

Abbildung 82: Aktivitätsprofil je Automat

Die in Abbildung 83 dargestellte Bildschirmmaske dient zum einen der Parametrisierung der betreffenden Task<sup>112</sup>, zum anderen zeigt sie die für Engpässe verantwortlichen Ressourcen – in der Abbildung in Rot dargestellt. Das Laufzeitverhalten der Task wird in der Rubrik „Time-related Statistics“ dargestellt. Dabei bezeichnet ‚Active‘ die Anzahl der Zeitschritte, die ein Automat aktiv war. ‚Blocked‘ bezeichnet den Stillstand eines Automaten wegen Ressourcenmangels. ‚Idle‘ bezeichnet die kumulierten Wartezeiten des Automaten bis zur jeweils nächsten Ausführung.

In der Rubrik Bottleneck/Shortage werden für die vier möglichen Ursachen von Blockaden die jeweils kumulierten Werte gezeigt.

<sup>112</sup> Siehe auch Kapitel 4.1.1.

Status of Task <internal number>/<name>		Time: 120	Model: <name of model>				
Parameter		Resources					
COUNT	32767	<b>Personnel</b>					
		Skilled	Maier	Müller	<b>Huber</b>	Schneider	<name>
DURATION	12	Avail./Busy	1	1	<b>0</b>	1	0
DURATION_MOD	1	<b>Equipment</b>					
		Necessary	Kanülen	Behälter	<b>Stethoskop</b>	<name>	
WAITING_TIME	0	Availability	2	1	<b>0</b>	0	
REMAINDER_DURAT	0	<b>Material</b>					
CONFIG	0	Necessary	Narkotika	B-Blocker	<name>	<name>	
FATHER	0	Availability	20	32	0	0	
EARNINGS	0	<b>Statistics</b>					
COST	0	<b>Time-related</b>		<b>Bottleneck/Shortage</b>			
		Active	80	Personnel	3		
PRIORITY	0	Blocked	10	Equipment	5		
DEADLINE	0	Idle	30	Material	1		
VARCOST_MOD	0	Active/Total Time	67%	Buffer	7		

Abbildung 83: Automatenbezogene Informationen

Das in Abbildung 84 dargestellte Gantt-Diagramm zeigt das Laufzeitverhalten der einzelnen Automaten während der Simulation an. Aktive Automaten werden mit einem Balken (Blau) dargestellt, geblockte Automaten mit Rauten (Rot). Automaten, die auf neue Aufträge warten, sind mit Quadraten (Grün) gekennzeichnet.



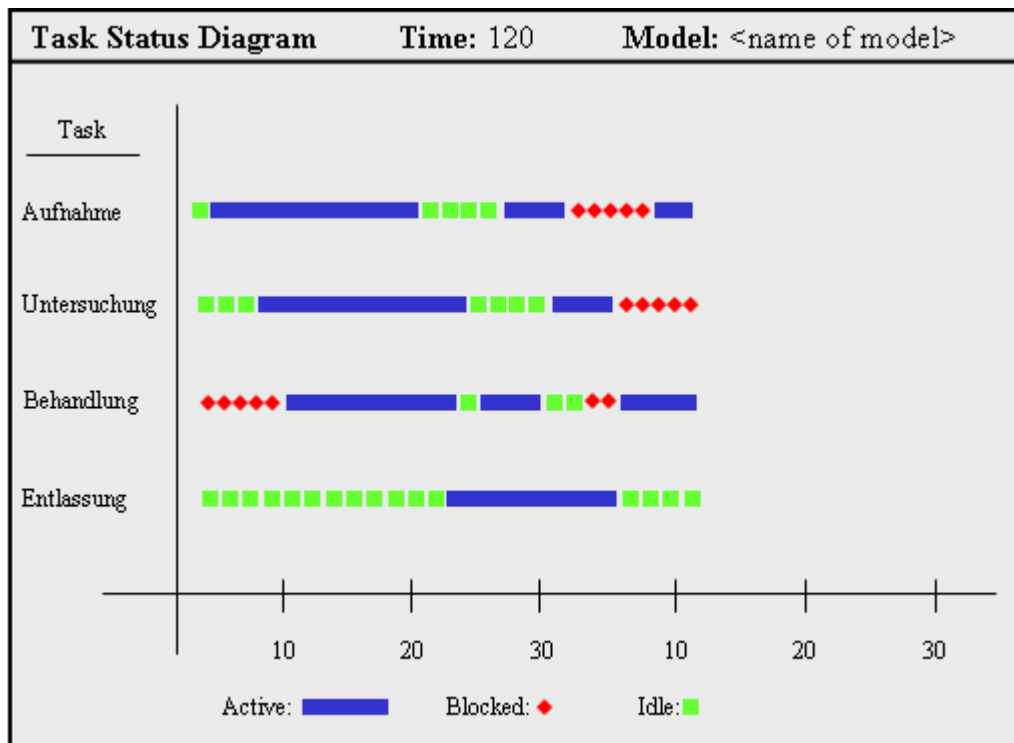


Abbildung 84: Laufzeitverhalten von Automaten im Gantt-Diagramm

Durch die Möglichkeit, in jedem Zeitschritt der Simulation eine aktualisierte Statistik am Bildschirm zu verfolgen, kann direkt beim Auftreten unerwünschter Ereignisse in die Prozessanalyse und –optimierung eingestiegen werden.

#### 4.2.3 Bufferbezogene Statistiken

Ein Simulationssystem sollte Auswertungen bezüglich Kapazitäten und Kosten der verwendeten Buffer zur Verfügung stellen. Kapazitätsbezogene Informationen, die für klinische Fragestellungen von Bedeutung sind, sind die minimale, maximale und durchschnittliche Belegung der Buffer. Ist die Kapazitätsgrenze eines Buffer erreicht, kann ein vorgelagerter Automat den Inhalt des Buffers nicht mehr inkrementieren<sup>113</sup>. Der Buffer blockt den entsprechenden Vorgang (Status BLOCKED). Es gilt:

$$\text{Aktueller Füllstand}_{\text{Buffer}, t} + \text{Zugänge}_{\text{Buffer}, t+1} \leq \text{Capacity}_{\text{Buffer}}$$

<sup>113</sup> Übertragen auf den klinischen Alltag bedeutet dies, dass z.B. alle Betten einer Station vollständig belegt sind und keine weiteren Patienten aufgenommen werden können.

Minimale, maximale und durchschnittliche Blockadezeiten werden je Buffer ermittelt und fortgeschrieben. Ferner sollte sich die Möglichkeit bieten, variable und fixe Kosten je Buffer festzulegen. Diese werden im Modellablauf über die Zeit kumuliert (vgl. Abb. 85 und Abb. 86).

<b>Buffer Results (1)</b>		<b>Time: 120</b>		<b>Model: &lt;name of model&gt;</b>			
Buffer	Average Content	Maximum Content	Minimum Content	Capacity	Blocked	Min. Time Blocked	Max. Time Blocked
Warteraum1	3	6	0	10	0	0	0
Warteraum2	4	8	0	8	4	12	22
<hr/>							
TOTAL							

Abbildung 85: Kapazitäts- und kostenbezogene Informationen für Buffer (1)

<b>Buffer Results (2)</b>					
Buffer	Average Time Blocked	# of Patients	Var.Cost	Fix Cost	Total Cost
Warteraum1	3	90	120	1.400	1.520
Warteraum2	4	75	80	990	1.070
<hr/>					
TOTAL			200	1.390	2.590

Abbildung 86: Kapazitäts- und kostenbezogene Informationen für Buffer (2)

Die nachstehende Statistik liefert Informationen über die Verweildauer von Transaktionen (vgl. Abb. 87). Im vorliegenden Beispiel sind die Transaktionen nach den Hauptkrankheitsbildern der Kardiologie eingeteilt. Für jede dieser Gruppen wird die Anzahl der Patienten sowie die durchschnittliche, minimale und maximale Verweildauer in dem betrachteten Buffer ermittelt. Diese Auswertung

ist aufgrund der Notwendigkeit der Reduktion der Verweildauern von Patienten für Kliniken besonders wichtig.<sup>114</sup>

<b>Buffer Results - Details</b>		<b>Time:</b> 120	<b>Model:</b> <name of model>	
<b>Buffer:</b> Patient Room 101		<b>Capacity:</b> 4		
Station: Cardiology		<b>Total # of Patients:</b> 98		
<hr/>				
<b>Avg. Content:</b>	2	<b>Avg. Time Blocked:</b>	5	<b>Avg. Length of Stay:</b> 5
<b>Min. Content:</b>	0	<b>Min. Time Blocked:</b>	1	<b>Min. Length of Stay:</b> 1
<b>Max. Content:</b>	4	<b>Max. Time Blocked:</b>	12	<b>Max. Length of Stay:</b> 7
<hr/>				
<b>Case Mix:</b>	<b># of Patients:</b>	<b>Avg. Length of Occupancy:</b>	<b>Min. Length of Occupancy:</b>	<b>Max. Length of Occupancy:</b>
Arrhythmia	42	4	2	7
Congestive Heart Failure	30	5	1	6
Valvular Heart Disease	8	7	2	4
Coronary Artery Disease	18	6	2	2

Abbildung 87: Detailinformationen je Buffer

#### 4.2.4 Transaktionsbezogene Statistiken

Wie in Kapitel 4.1.4 beschrieben, sollten vom Modellnutzer zeitbezogene Parameter individuell angelegt und im Anschluss an den Simulationslauf ausgewertet werden können. Die nachstehende Statistik zeigt an einem Beispiel aus der Kardiologie für drei individuell festgelegte Parameter die minimale, maximale und durchschnittliche Zeitdauer vom Anfangspunkt a zum Endpunkt b im Modellnetz der Simulation. Als Anfangs- und Endzeitpunkt der Messung werden Tasks oder Buffer angegeben.

<sup>114</sup> Siehe auch Kapitel 1.1.

Statistics of Time-related User Parameter			Time: 160	Model: <name of model>	
Name	Start of Measurement (Task / Buffer)	End of Measurement (Task / Buffer)	Minimum	Maximum	Average
Behandlungsdauer	Patientenaufnahme	Patientenentlassung	5	12	7
Door-to-Balloon	Notaufnahme	Dilatation	12	20	14
Door-to-Needle	Notaufnahme	Lysetherapie	5	12	8

Abbildung 88: Auswertung zeitbezogener User-Parameter

## 5. Standardisierte Beschreibung klinischer Prozesse

In Kapitel 5 werden klinische Prozesse als Teil des übergeordneten Patientenprozesses<sup>115</sup> ausgehend von einem klinischen Behandlungspfad abgeleitet, in standardisierter Form beschrieben und in die entsprechende Ablaufsequenz gebracht. Als Ausgangsbasis dient der Behandlungspfad für das akute Koronarsyndrom<sup>116</sup>, da dieser wesentliche Elemente des Patientenprozesses enthält und daher die geeignete Basis für die weitere Prozessbeschreibung darstellt. Der Fokus bei diesem Behandlungspfad wird auf die Diagnose und die Ersttherapie gelegt. Dies ist für die standardisierte Beschreibung der für diese Arbeit relevanten klinischen Prozesse ausreichend. Supportprozesse werden vernachlässigt<sup>117</sup>, da sich die vorliegende Arbeit auf den Patientenprozess im engeren Sinn fokussiert. Bei der inhaltlichen Beschreibung des Behandlungspfades wird an den Stellen abstrahiert, an denen sonst eine tiefergehende medizinische Diskussion unvermeidbar wäre. Der Abstraktionsgrad wird dabei so gewählt, dass es auf die standardisierten Prozessbeschreibungen keine negativen Einflüsse im Sinne von fehlenden bzw. lückenhaften Prozessschritten gibt.

### 5.1 Klinische Behandlungspfade

Der Begriff des klinischen Behandlungspfades wird in der Literatur teilweise unterschiedlich definiert. Begriffe wie Clinical oder Critical Pathways, Behandlungspfade, Leitlinien und Arbeitsanweisungen werden zum Teil synonym, zum Teil unterschiedlich verwendet. Für die vorliegende Arbeit wird auf eine Definition von Dr. Röder, Prof. Hindle et. al [RHLJ03 S.21f] zurückgegriffen. Diese definieren den klinischen Behandlungspfad wie folgt: „Ein klinischer Behandlungspfad ist der im Behandlungsteam selbst gefundene berufsgruppen- und institutionsübergreifende Konsens für die beste Durchführung der gesamten

---

<sup>115</sup> In der vorliegenden Arbeit auch als klinischer Kernprozess bezeichnet.

<sup>116</sup> Krankheitsbild aus der Kardiologie; siehe auch Kapitel 5.2.

<sup>117</sup> Dadurch erfolgt auch eine Abgrenzung zur Umwelt des betrachteten Systemausschnitts.

stationären Behandlung unter Wahrung festgelegter Behandlungsqualität sowie unter Berücksichtigung der notwendigen und verfügbaren Ressourcen, ebenso unter Festlegung der Aufgaben sowie Durchführungs- und Ergebnisverantwortlichkeiten. Der klinische Behandlungspfad steuert den Behandlungsprozess; gleichzeitig ist er das behandlungsbegleitende Dokumentationsinstrument und erlaubt die Kommentierung von Normabweichungen zum Zwecke fortgesetzter Evaluation und Verbesserung“ [RHLJ03 S.21f]. In Abgrenzung zum klinischen Behandlungspfad wird der Patientenpfad als die für den „Patienten verständliche aggregierte Form eines Behandlungspfades“ [RHLJ03 S.22] definiert.

Ein Behandlungspfad stellt keine starre Arbeitsanweisung dar, welche die Einschränkung der Handlungsfreiheit zur Folge hat, sondern vielmehr eine Art Handlungskorridor für Diagnose und Therapie eines Patienten [ROE02 S.462ff]. Das bedeutet, dass bei patientenindividuellen Umständen auch Abweichungen gestattet sind. Abweichungen vom Behandlungspfad sind zu dokumentieren, um Abweichungsanalysen durchführen zu können. So können Behandlungspfade gezielt optimiert werden.

Als wesentliche Zielsetzungen, die mit der Erstellung und Nutzung von Behandlungspfaden verfolgt werden, nennen Dr. Röder, Prof. Hindle et. al<sup>118</sup> die

- transparente Darstellung aller Abläufe inkl. der entscheidungs- und ereignisabhängigen Teilpfade (=Teilprozessschritte)
- Ausrichtung der Prozesse auf definierte Qualitäts- und Behandlungsziele
- Verbesserung der Prozess- und Ergebnisqualität; Verbesserung der Patientenzufriedenheit
- Sicherstellung der Ressourcenverfügbarkeit
- Schaffen von Prozesskostentransparenz
- Reduktion von Konflikten mit Kostenträgern hinsichtlich der Angemessenheit der erbrachten Leistung

---

<sup>118</sup> Vgl. [RHLJ03].

In den folgenden Kapiteln dieser Arbeit wird deutlich, wie die Modellierung und Simulation von Behandlungspfaden einen wesentlichen Beitrag leistet, den oben genannten Zielsetzungen gerecht zu werden.

## 5.2 Das akute Koronarsyndrom

Kardiovaskuläre<sup>119</sup> Erkrankungen gehören in den westlichen Industriestaaten zur häufigsten Todesursache. Akute koronare Syndrome sind Situationen, in denen plötzlich eine myokardiale Ischämie<sup>120</sup> auftritt. Dies führt i.d.R. zu Brustschmerzen, die im Gegensatz zur chronischen Angina Pectoris länger anhalten, in immer kürzeren Abständen und auch in Ruhesituationen auftreten können. Der Begriff des akuten Koronarsyndroms fasst die unmittelbar lebensbedrohlichen Phasen der koronaren Herzerkrankungen zusammen. Dazu gehören die instabile Angina, der akute Myokardinfarkt<sup>121</sup> sowie der plötzliche Herztod. Aufgrund der fließenden Übergänge dieser klinischen Krankheitsformen werden Patienten anhand des Elektrokardiogramms<sup>122</sup> nun in die Gruppen „mit ST-Streckenhebung“<sup>123</sup> und „ohne ST-Streckenhebung“<sup>124</sup> eingeteilt [HAB04 S.4f]. Charakteristisch für den ST-Streckenhebungsinfarkt sind Schmerzen in der Brust, die länger als 20 Minuten andauern sowie oftmals Schmerzattacken in den letzten Stunden oder Tagen vor dem Infarkt. Letztere können auch im Ruhezustand bzw. schon bei geringer Belastung auftreten.

---

<sup>119</sup> Herz und Gefäße betreffend.

<sup>120</sup> Minderdurchblutung des Herzmuskels.

<sup>121</sup> Bei einem Myokardinfarkt stirbt ein Teil des Herzmuskels durch den Verschluss eines Herzkranzgefäßes.

<sup>122</sup> Bei der Elektrokardiographie gewonnene Aufnahme.

<sup>123</sup> STEMI = ST-Streckenhebungsinfarkt.

<sup>124</sup> NSTEMI = Nicht-ST-Streckenhebungsinfarkt.

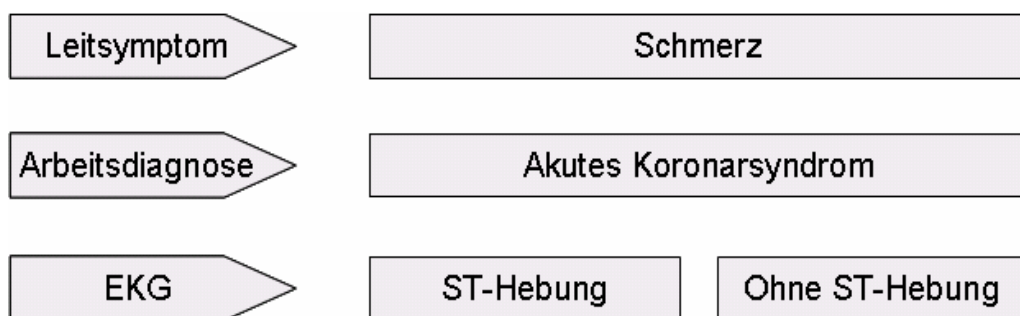


Abbildung 89: Terminologie zum akuten Koronarsyndrom

Ursachen des akuten Koronarsyndroms sind neben Gefäßspasmen<sup>125</sup> und Dissektionen<sup>126</sup> des Koronargefäßes überwiegend das Aufbrechen einer Plaque<sup>127</sup> in den Herzkranzgefäßen mit Gerinnselbildung und/oder Einblutung in die Gefäße. Die Folgen eines akuten Koronarsyndroms reichen von einer instabilen Angina Pectoris ohne bleibenden Schaden am Herzmuskel über den Herzinfarkt mit Myokardnekrose<sup>128</sup> bis hin zum plötzlichen Herztod.

### 5.2.1 Diagnose und Risikostratifizierung

Das Leitsymptom des ACS<sup>129</sup> ist der akute thorakale Schmerz (vgl. Abb. 89). Da dieser eine geringe Spezialität aufweist und sich dahinter ein Spektrum von funktionellen Beschwerden verbergen kann, ist eine Differentialdiagnose notwendig (vgl. Tabelle 4). Charakteristisch für das Vorliegen eines ACS sind Ruhebeschwerden, die länger als 20 Minuten andauern sowie Besserung auf antiischämische Medikamente<sup>130</sup> innerhalb von 5 Minuten.

<sup>125</sup> Zusammenziehen der Wandmuskulatur vom Herzen wegführender Blutgefäße (Arterien).

<sup>126</sup> Gefäßwandinriss.

<sup>127</sup> Gefäßablagerung /-verengung.

<sup>128</sup> Absterben von Herzmuskelzellen.

<sup>129</sup> ACS = Acute Coronary Syndrom (dt.: akutes Koronarsyndrom).

<sup>130</sup> Z.B. sublinguale Nitrate; Nitrate wirken gefäßerweiternd.



<b>Kardiovaskuläre Erkrankungen</b>
<ul style="list-style-type: none"> <li>- Rhythmusstörungen</li> <li>- Perikarditis<sup>131</sup></li> <li>- Myokarditis<sup>132</sup></li> <li>- Aortendissektion<sup>133</sup></li> </ul>
<b>Pulmonale Erkrankungen</b>
<ul style="list-style-type: none"> <li>- Lungenembolie<sup>134</sup></li> <li>- Pleuritis<sup>135</sup></li> <li>- Pneumothorax<sup>136</sup></li> </ul>
<b>Skeletterkrankungen</b>
<ul style="list-style-type: none"> <li>- Rippenfraktur / Prellungen</li> <li>- BWS-Erkrankungen</li> <li>- Tietze-Syndrom<sup>137</sup></li> </ul>
<b>Gastrointestinalerkrankungen</b>
<ul style="list-style-type: none"> <li>- Oesophagitis<sup>138</sup>/Ruptur</li> <li>- Ulcus<sup>139</sup> (Perforation)</li> <li>- Akute Pankreatitis<sup>140</sup></li> <li>- Gallenkolik</li> </ul>

Tabelle 4: Differentialdiagnosen zum akuten Koronarsyndrom

Die Krankengeschichte des Patienten mit früheren Ereignissen (z.B. Bypass-OP, Koronarintervention) sowie Hinweise auf Risikofaktoren (z.B. Diabetes, Kreislauf, Nikotinabusus, Lebensalter, Niereninsuffizienz) erhöhen die Wahrscheinlichkeit des Vorliegens einer koronaren Herzerkrankung und damit auch für das ACS. Bei der Diagnose nimmt die Elektrokardiographie eine zentrale Stellung ein. Sie kann

<sup>131</sup> Herzbeutelentzündung.

<sup>132</sup> Herzmuskelentzündung.

<sup>133</sup> Abtrennung der Hauptschlagader des menschlichen Körpers.

<sup>134</sup> Embolie = Verstopfung eines Blutgefäßes.

<sup>135</sup> Brustfell-/Rippenfellentzündung.

<sup>136</sup> Luftansammlung im Brustfellraum.

<sup>137</sup> Brust- und Brustbeinschmerzen, die bei Beschwerdezunahme auch in den linken Arm ausstrahlen und dann Herzbeschwerden vortäuschen können.

<sup>138</sup> Oesophagitis bedeutet die Entzündung der inneren Auskleidung des Oesophagus (=Speiseröhre).

<sup>139</sup> Ulcus = Geschwür.

<sup>140</sup> Entzündung der Bauchspeicheldrüse.

eine bestehende Minderdurchblutung des Herzmuskels aufdecken und Herzrhythmusstörungen erfassen. Bei Vorliegen eines „normalen“ Elektrokardiogramms kann dennoch ein ACS nicht vollkommen ausgeschlossen werden. Die Aussagekraft des EKG kann durch kontinuierliches Monitoring der ST-Strecke verbessert werden. [KFCH01 S.64ff; SAG01 S.707ff; HoDe99 S.667ff].

Biochemische Marker sind heute bei der Evaluierung von Patienten mit ACS ohne ST-Hebung unverzichtbar. Hinsichtlich Sensitivität kommt Troponin<sup>141</sup> als Marker die höchste Bedeutung zu. Die Troponinbestimmung liefert eine gute Aussage darüber, ob ein Herzmuskelschaden vorliegt oder nicht. Eine vernünftige Aussage ist allerdings erst ca. 3-4 Stunden nach Ischämieergebnis möglich, da i.d.R. erst dann erhöhte Troponinwerte auftreten und messbar sind. Bei Patienten mit ST-Streckenhebungsinfarkt im EKG darf aufgrund des fehlenden Anstiegs der Marker innerhalb der ersten 2-3 Stunden nach Auftreten des Infarktes der Nachweis des Anstiegs nicht abgewartet werden. Hier sind umgehend Reperfusionmaßnahmen<sup>142</sup> einzuleiten [HAB04a S.74f und HAB04b S.325].

Eine weitere diagnostische Methode zur Diagnose und Schweregradbeurteilung ist die Koronarangiographie<sup>143</sup>. Deren Befund bildet die Grundlage für die Indikation zur perkutanen<sup>144</sup> oder operativen Revaskularisation<sup>145</sup> und liefert Hinweise zur Risikobeurteilung. So haben beispielsweise Patienten mit Hauptstammstenosen<sup>146</sup> oder sichtbarem Thrombus<sup>147</sup> ein erhöhtes Risiko [HBHS99 S.1509ff; HAB04 S.10 und S.12f].

---

<sup>141</sup> Troponin ist eines der vier wesentlichen Proteine der Muskelzelle.

<sup>142</sup> Reperfusion = Wiederdurchströmung eines Blutgefäßes, das durch einen Thrombus (Klumpen, Blutpfropf) verschlossen war.

<sup>143</sup> Kontrastdarstellung der Koronargefäße.

<sup>144</sup> Perkutan = durch die Haut hindurch.

<sup>145</sup> Wiederherstellung der Durchblutung (operativ oder perkutan).

<sup>146</sup> Stenose = angeborene oder erworbene Verengung eines Blutgefäßes.

<sup>147</sup> Thrombus = Blutpfropf in einem Blutgefäß.



Abbildung 90: Röntgendarstellung einer Gefäßverengung mit Kontrastmittel<sup>148</sup>

Zur Durchführung einer Koronarangiographie wird dem Patienten nach örtlicher Betäubung eine sog. Schleuse<sup>149</sup> gesetzt, über die anschließend der Katheter eingeführt und positioniert wird. Dem Patienten wird Röntgenkontrastmittel in die Herzkammer, in die Hauptschlagader sowie die linke und rechte Koronararterie eingespritzt. Dadurch erkennt der Untersuchende direkt über Videodokumentation, wo genau Stenosen oder Verschlüsse in den Herzkranzgefäßen bestehen (vgl. Abb. 90).

### 5.2.2 Therapie des akuten Koronarsyndroms

Die deutsche Gesellschaft für Kardiologie nennt folgende prästationäre Therapiemaßnahmen bei Verdacht eines akuten Koronarsyndroms [HAB04 S.11]:

- O<sub>2</sub> über Nasensonde/Maske (4-8l/min)
- Glyceroltrinitrat 0,4-0,8 mg s.l.<sup>150</sup>, evtl. wiederholt
- Morphin 3-5 mg i.v., ggf. wiederholt bis Schmerzfreiheit
- bei vagaler<sup>151</sup> Reaktion Atopin 0,5 mg i.v.<sup>152</sup>
- bei Übelkeit / Erbrechen Antiemetika<sup>153</sup> (z.B. Metoclopramid)

<sup>148</sup> Abbildung nach [CARD05].

<sup>149</sup> Transfemoraler Zugang (femoral = den Oberschenkel betreffend).

<sup>150</sup> s.l. = sublingual.

<sup>151</sup> Den Vagusnerv (=Hirnnerv) betreffend.

<sup>152</sup> i.v. = intravenös.

<sup>153</sup> Arzneimittel mit Wirkung gegen Übelkeit und Erbrechen.

- bei Tachykardie<sup>154</sup> langwirksamer Beta-Blocker<sup>155</sup> (z.B. Metoprolol 5 mg i.v.)
- Acetylsalicylsäure (≥ 250 mg i.v.)
- Heparin<sup>156</sup> 70 U/kg i.v., max. 5.000 U oder Enoxaparin 30 mg i.v. + 1mg/kg s.c.

### 5.2.2.1 Therapie des akuten Koronarsyndroms ohne ST-Hebung

Die Akuttherapie im Krankenhaus zielt darauf ab, Beschwerdefreiheit zu erreichen und gleichzeitig Komplikationen<sup>157</sup> und Tod zu verhindern.

Die antiischämische Therapie soll eine deutliche Reduktion der Schmerzsymptomatik des Patienten erreichen. Hierfür werden Nitrate<sup>158</sup> und Beta-Blocker eingesetzt. Nitrate können als Kapsel<sup>159</sup> oder Spray<sup>160</sup> verabreicht werden. Wiederholte Dosen bis zum Eintreten des Therapieeffekts sind möglich. Die Wirkung hält ungefähr 20 Minuten an und kann mit intravenösen Verabreichungsformen nach stationärer Aufnahme fortgesetzt werden. Bei einem Blutdruck unter 90 mmHg systolisch<sup>161</sup> ist allerdings Vorsicht geboten [HAB04a S. 76f]. Beta-Rezeptorenblocker sind wirksame antiischämische Medikamente, besonders bei Patienten mit erhöhten Blutdruckwerten und Tachykardie. Durch Beta-Blocker wird eine überhöhte Aktivität in dem Teil des Nervensystems<sup>162</sup> verringert, der unter anderem für die Beschleunigung der Herzschläge und Steigerung des Blutdrucks (z.B. unter Stress oder bei gesteigerter körperlicher Leistung) verantwortlich ist. Das Herz würde sonst durch den erhöhten Sauerstoffbedarf, den der Herzmuskel benötigt, belastet. Beta-Blocker senken den Sauerstoffbedarf des Herzmuskels, indem sie die Herzfrequenzsteigerung,

---

<sup>154</sup> Erhöhung der Herzfrequenz auf über 100/min in Ruhe.

<sup>155</sup> Beta-Blocker blockieren Beta-Rezeptoren im Körper und hemmen so die Wirkung einiger Stresshormone (z.B. Adrenalin).

<sup>156</sup> Gerinnungshemmende Substanz.

<sup>157</sup> Z.B. den Myokardinfarkt.

<sup>158</sup> Wirken gefäßerweiternd.

<sup>159</sup> 1-2 Kapseln, sublingual.

<sup>160</sup> 2-3 Hub.

<sup>161</sup> Systolischer Blutdruck = maximaler Druck im Gefäß (Spitzendruck).

<sup>162</sup> Sympathikus.

den damit einhergehenden Blutdruckanstieg und die Pumpleistung<sup>163</sup> bremsen. Ziel ist es, eine Herzfrequenz von 50-60/min zu erreichen [Hab04a S.77].

Neben den antiischämischen Medikamenten gehört eine gerinnungshemmende Therapie mittels Antithrombinen<sup>164</sup> und Thrombozytenaggregationshemmer<sup>165</sup> zum Behandlungskonzept des ACS. Zu den Antithrombinen gehört Heparin<sup>166</sup>. Dieses<sup>167</sup> soll nach einer Anfangsdosierung von maximal 5000E mittels Infusion von 1000E/Stunde verabreicht werden. Nach 6 Stunden sollte eine erste Kontrolle erfolgen. [TOM88 S.1105ff; OWG96 S.811ff] Zu den Thrombozytenaggregationshemmern zählen Acetylsalicylsäure (ASS) und Clopidogrel. ASS gehört zur Standardtherapie und gilt als hochwirksam und kostengünstig [Risc90 S.827ff; TWM93 S.2045ff]. Die tägliche Dosierung liegt bei 100mg. Clopidogrel kann in Kombination mit ASS verabreicht werden. In der Dauerbehandlung liegt die Dosierung bei 75mg/Tag. Ein beschleunigter Wirkungseintritt kann durch die Gabe von 300mg nach ca. 6 Stunden bzw. 600mg nach ca. 3 Stunden erreicht werden [YZM01 S.494ff; CBT00 S.2823ff; HAB04a S.12].

Unabhängig vom primären Erfolg der Pharmakotherapie wird in der heutigen Zeit i.d.R. eine frühe Koronarangiographie<sup>168</sup> durchgeführt, um durch Bypassoperation oder Katheterintervention eine koronare Revaskularisation<sup>169</sup> zu erreichen [HGTH00 S.1159ff; KHH03 S.651ff; LHK02 S.1902ff]. Im Rahmen einer interventionellen Therapie senkt eine koronare Stentimplantation das Risiko einer Restenose<sup>170</sup> im Vergleich zur alleinigen PTCA<sup>171</sup>. Ballondilatationen ohne Stentimplantation haben in der klinischen Praxis daher an Bedeutung verloren.

---

<sup>163</sup> Herzminutenvolumen.

<sup>164</sup> Substanz, die Thrombin (=proteolytischer Faktor der Blutgerinnung) inaktiviert oder hemmt.

<sup>165</sup> Substanzen, die die Zusammenballung von Blutplättchen verhindern oder hemmen.

<sup>166</sup> Gerinnungshemmende Substanz.

<sup>167</sup> Unfraktioniertes Heparin.

<sup>168</sup> Angiographie = Kontrastmitteldarstellung von Gefäßen.

<sup>169</sup> Wiederherstellung der Durchblutung (operativ oder perkutan).

<sup>170</sup> Erneute Stenose (Einengung von Gefäßen) einer operativ aufgeweiteten Einengung.

<sup>171</sup> PTCA = Perkutane transluminale Koronarangioplastie (Ballondilatation).

### 5.2.2.2 Therapie des akuten Koronarsyndroms mit ST-Hebung

Im Rahmen einer prästationären Therapie findet die Fibrinolyse-Therapie Anwendung. Fibrin ist ein hochmolekulares, wasserundurchlässiges Protein und entsteht bei der Blutgerinnung aus Fibrinogen. Fibrinogen bildet die Vorstufe zu Fibrin und ist ein in der Leber gebildeter Blutgerinnungsfaktor. Die Fibrinolyse ist eine enzymatische Aufspaltung von Fibrin oder Fibringerinnsel. Die Wirksamkeit der Fibrinolyse hängt von der Zeit ab, die seit Schmerzbeginn verstrichen ist. Sie ist in den ersten 3 Stunden nach Schmerzbeginn am wirksamsten. In den ersten 2 bis 4 Stunden fällt die Wirksamkeit exponentiell ab, anschließend sinkt sie linear weiter ab. Bis 12 Stunden nach Schmerzbeginn konnte eine Wirksamkeit der Fibrinolyse belegt werden. [HAB04b S.326f; Pach02 S.225] Die Zeitdauer vom Erstkontakt bis zur prästationären Fibrinolyse nennt man „contact-to-needle“-Zeit.

Umgehend nach stationärer Aufnahme muss die Reperfusionstherapie eingeleitet bzw. fortgesetzt werden. Die maximale Zeitdauer bis zur Einleitung der medikamentösen Fibrinolyse darf 30 Minuten nicht überschreiten. Diese Zeitspanne nennt man „door-to-needle“-Zeit. Im Falle einer primären perkutanen Koronarintervention (=PCI) soll die „door-to-balloon“-Zeit von maximal 60 Minuten nicht überstiegen werden. Ziel einer perkutanen Koronarintervention ist im ersten Schritt eine Sicherung der Diagnose. Beim Nachweis eines verschlossenen Blutgefäßes wird anschließend eine Ballondilatation – i.d.R. mit Stentimplantation – durchgeführt. In den meisten Fällen gelingt so die Wiederherstellung der Durchblutung des verschlossenen Gefäßes [HAB04b S.328f; Pach02 S.225]. Beim ST-Hebungsinfarkt ist eine frühe PTCA<sup>172</sup> der Lysetherapie eindeutig überlegen, so dass die Bedeutung der Lysetherapie in den Hintergrund tritt, wenn die PTCA zeitnah durchgeführt werden kann.

Eine operative Therapie, in diesem Fall die Bypasschirurgie, ist bei Patienten mit erfolgreicher Fibrinolyse und/oder Ballondilatation sekundär. Begründet wird dies mit der Zeitspanne, die bis zum Beginn einer Operation vergeht, sowie hohe Komplikationsraten bei Akuteingriffen. Eine Herzoperation wird deshalb nur unter bestimmten Umständen beim ST-Streckenhebungsinfarkt in Betracht gezogen.

---

<sup>172</sup> Perkutane transluminale Koronarangioplastie.

Diese sind z.B. bei erfolgloser PCI mit persistierendem Verschluss eines Gefäßes oder Komplikationen bei der Ballondilatation<sup>173</sup> gegeben [HAB04b S.330].

### **5.3 Entwicklung der Prozesskomponenten für den klinischen Behandlungspfad des akuten Koronarsyndroms**

Auf Basis der Erkenntnisse des Kapitels 5.2 ergibt sich der in Abbildung 91 dargestellte klinische Behandlungspfad. Er zeigt den Ablauf der einzelnen Prozessschritte, beginnend mit dem Eintreffen des Patienten im Krankenhaus, der Diagnosestellung und der anschließenden Therapie.

---

<sup>173</sup> Z.B. Perforation eines Koronargefäßes.

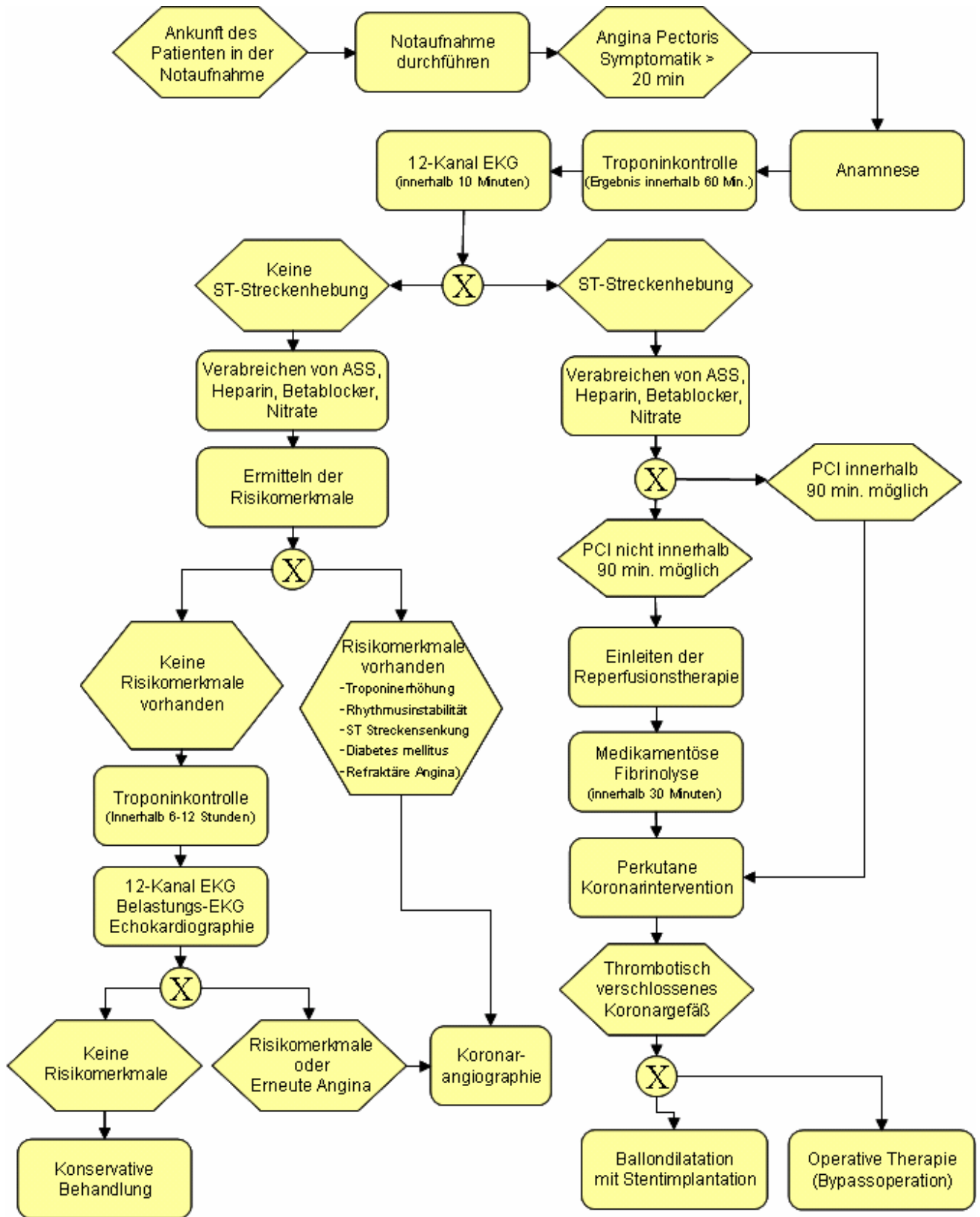


Abbildung 91: Klinischer Behandlungspfad für das akute Koronarsyndrom



Im Folgenden wird nun die standardisierte Beschreibung von Prozessschritten auf den unterschiedlichen Prozessebenen - wie in Kapitel 3.2 skizziert - fortgeführt und die „Toolbox“ mit den Prozesskomponenten für das Krankheitsbild des akuten Koronarsyndroms vervollständigt.

Die Subprozesse „Patienten aufnehmen“, „Patienten untersuchen“ und „Patienten behandeln“ sind Bestandteil des Patientenprozesses.<sup>174</sup> Der Prozess der Notaufnahme als Subprozess der Patientenaufnahme wurde bereits in Kapitel 3.2.2 beschrieben. Die Subprozesse „Patienten untersuchen (Untersuchung)“ und „Patienten behandeln (Behandlung)“ werden nun in analoger Art und Weise beschrieben. Abbildung 92 spiegelt die Gesamtsicht wider.

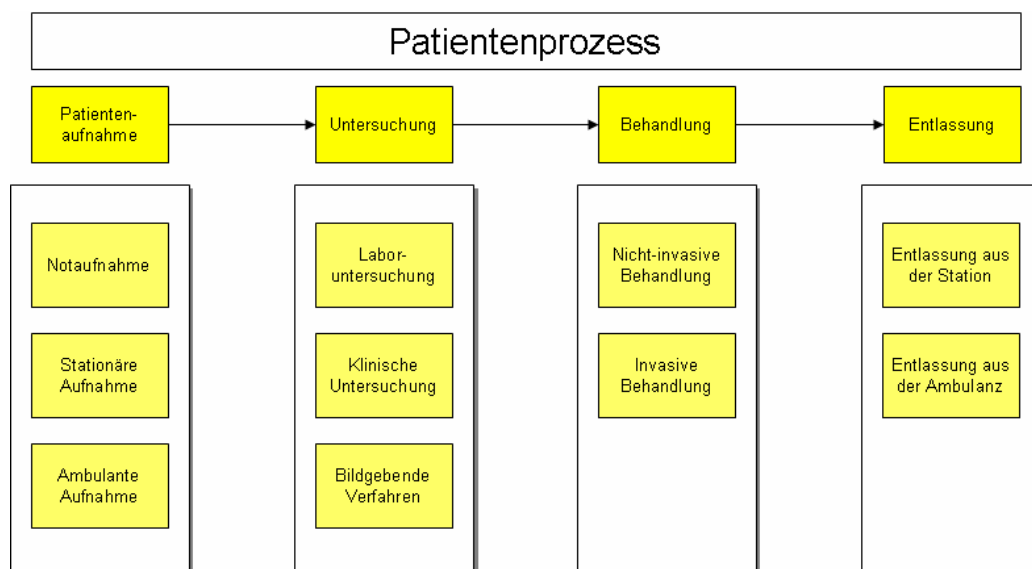


Abbildung 92: Gesamtübersicht mit Prozesskategorien

Der Prozess „Untersuchung“ wird in die Prozesskategorien Laboruntersuchung/-tests (z.B. histologische<sup>175</sup> oder mikrobiologische Untersuchungen), klinische Untersuchung (z.B. Biopsie<sup>176</sup>, Elektrokardiographie) und bildgebende Verfahren (z.B. Angiographie, Computertomographie, Ultraschall) eingeteilt. Abbildung 93 zeigt die grafische Darstellung der genannten Prozesse bis Prozesslevel 3. Am Beispiel der Elektrokardiographie und der Koronarangiographie werden die

<sup>174</sup> Siehe hierzu auch Kapitel 3.2.2.

<sup>175</sup> Histologisch = die Gewebe betreffend.

<sup>176</sup> Gewebeentnahme durch Punktion oder Exzision.

Prozesskategorien „klinische Untersuchung“ und „bildgebende Verfahren“ bis Prozesslevel 4 beschrieben (vgl. Abb. 94 und Abb. 95).

Der Prozess „Patienten behandeln (Behandlung)“ wird in die Prozesskategorien „nicht-invasiv“ und „invasiv“ eingeteilt. Zu den nicht-invasiven Prozeduren zählen z.B. das Verabreichen von Medikamenten und die Physiotherapie. Invasive Prozeduren sind z.B. die klassische Operation und die PTCA<sup>177</sup>. Am Beispiel der Medikation und der PTCA werden die Prozesskategorien „nicht-invasiv“ und „invasiv“ bis Prozesslevel 4 beschrieben (vgl. Abb. 96, Abb. 97 und Abb. 98).

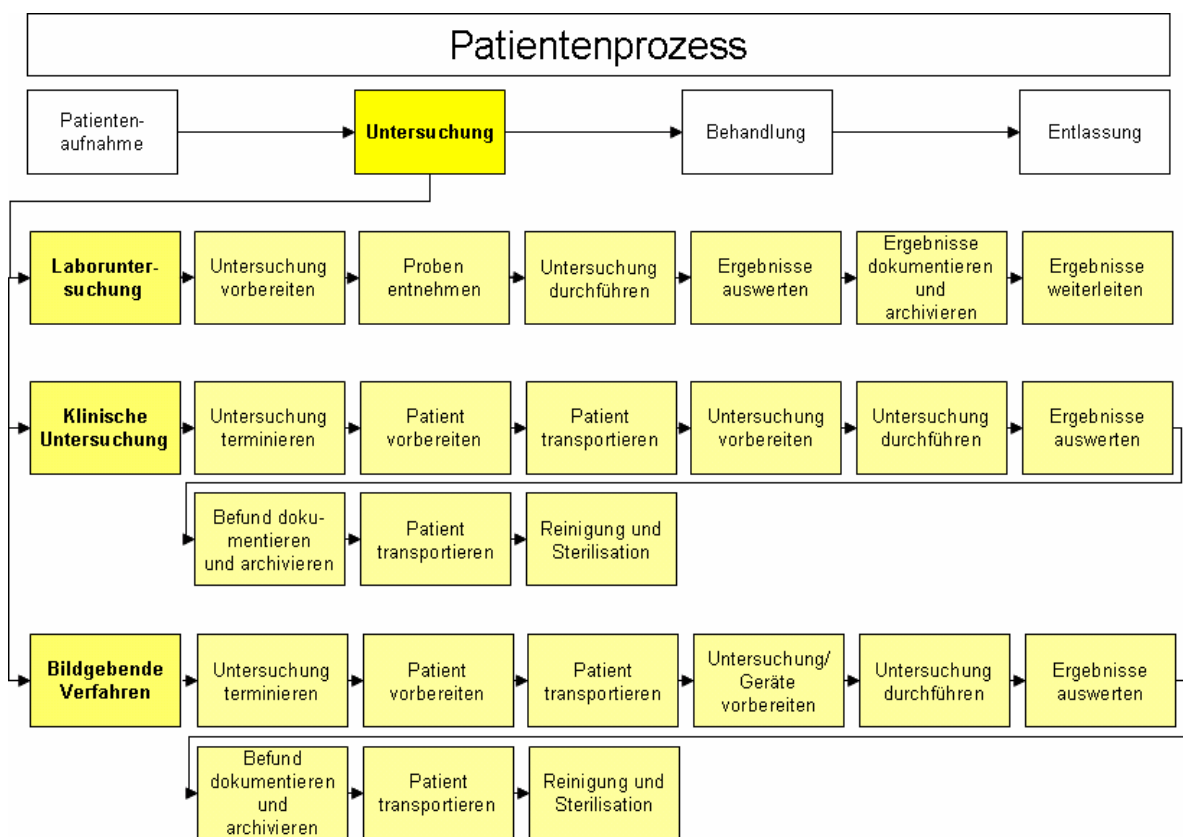


Abbildung 93: Darstellung des Teilprozesses „Patientenuntersuchung“ mit zugehörigen Subprozessen

<sup>177</sup> Perkutane transluminale Koronarangioplastie.

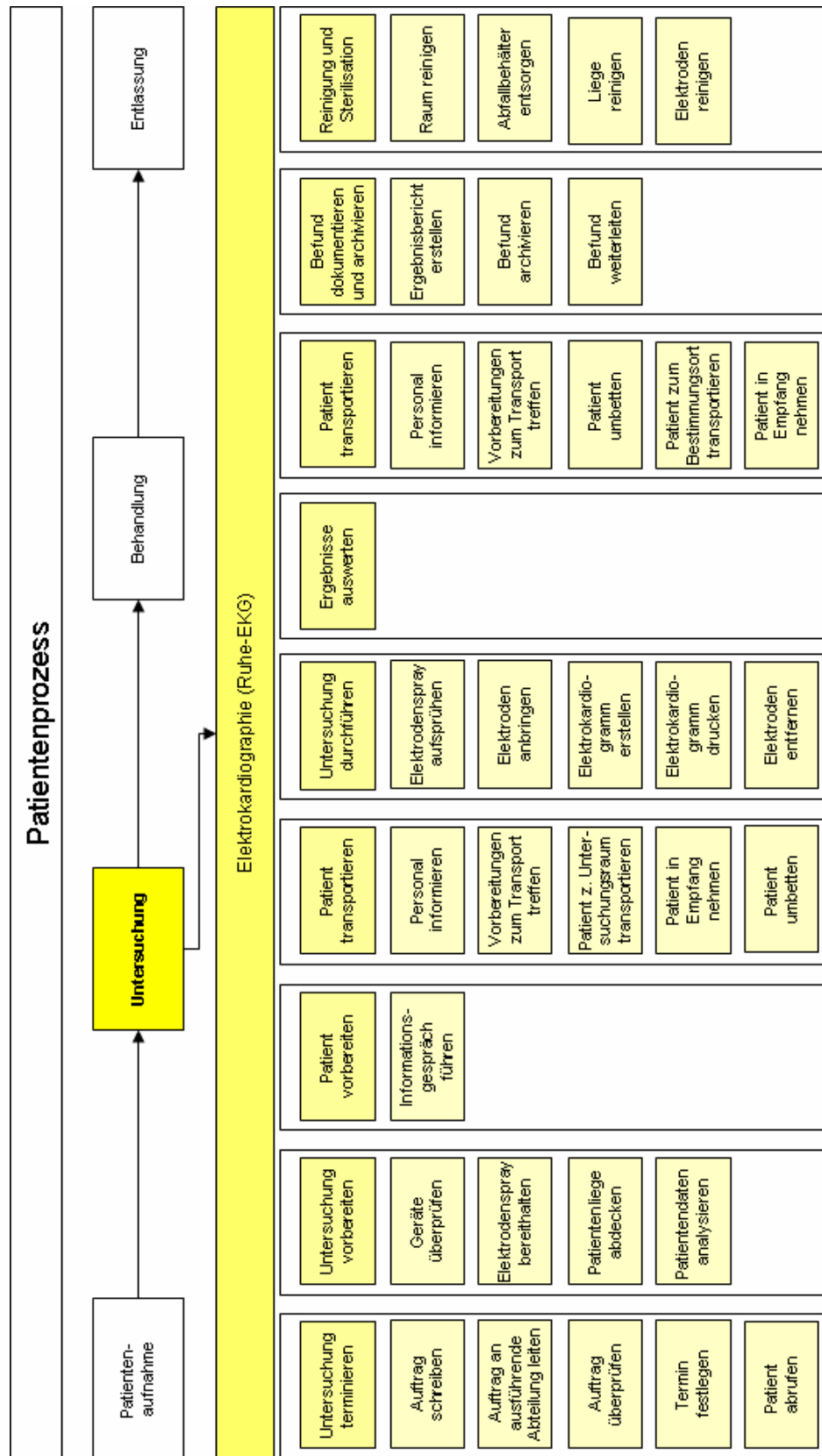


Abbildung 94: Darstellung des Teilprozesses „klinische Untersuchung“ am Beispiel der Elektrokardiographie bis Prozesslevel 4

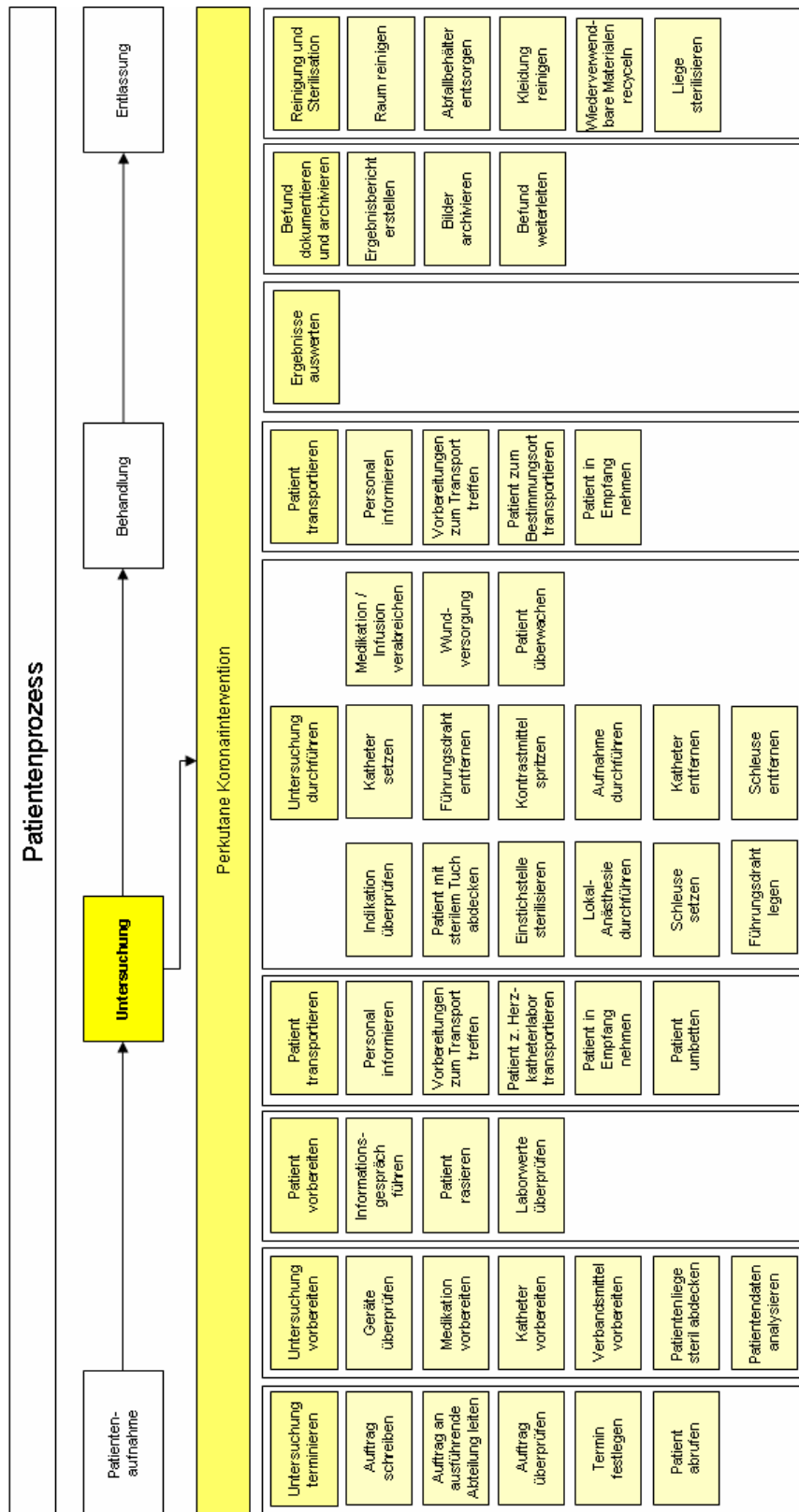


Abbildung 95: Darstellung des Teilprozesses „bildgebende Verfahren“ am Beispiel der perkutanen Koronarintervention mittels Angiographie bis Prozesslevel 4

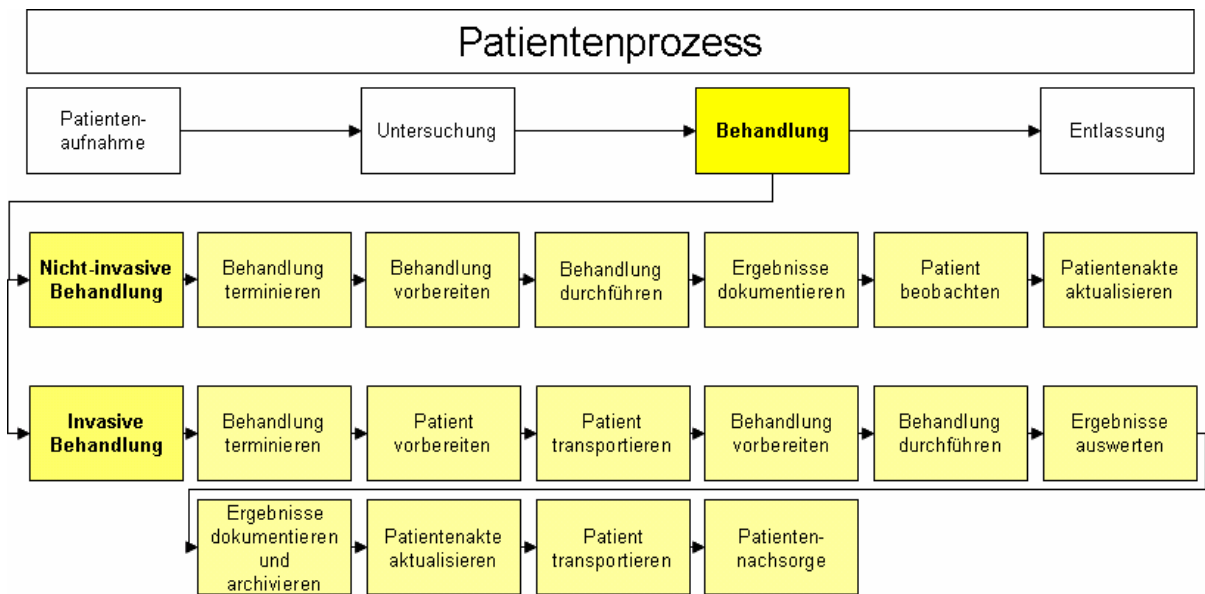


Abbildung 96: Darstellung des Teilprozesses „Patient behandeln“ mit zugehörigen Subprozessen bis Prozesslevel 3

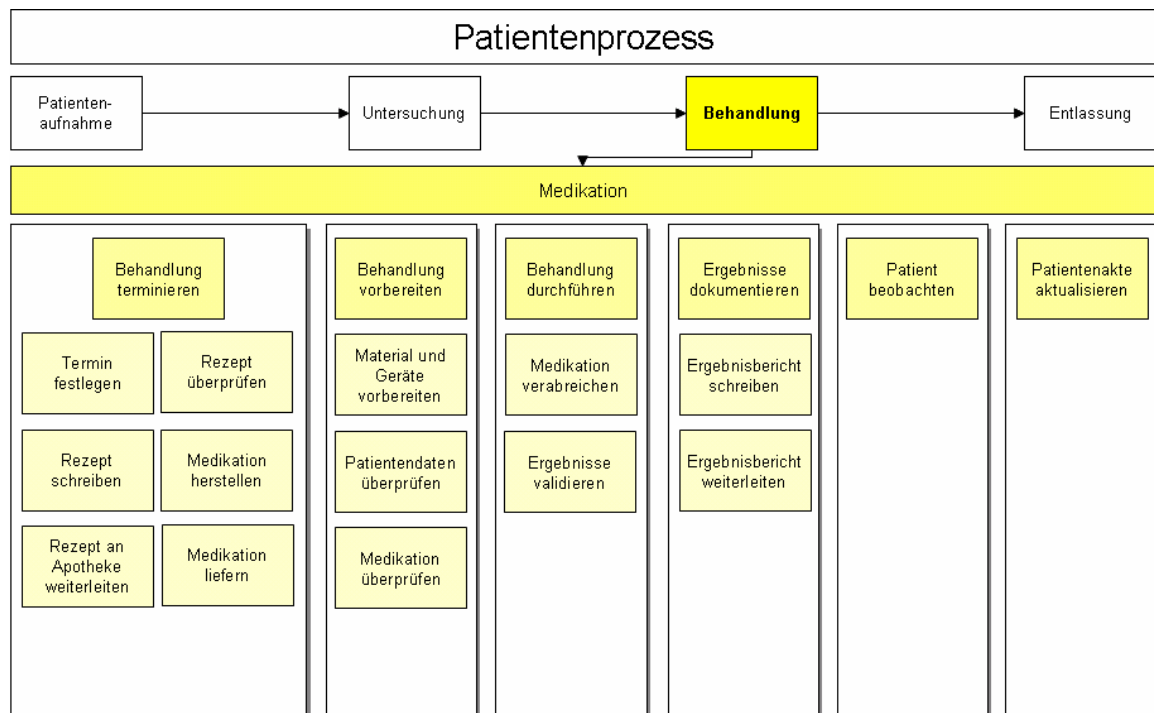


Abbildung 97: Darstellung des Teilprozesses „Nicht-invasive Behandlung“ am Beispiel der Medikation bis Prozesslevel 4



## 6. Entwicklung eines validierten Prozessreferenzmodells mittels standardisierter und wieder verwendbarer Modellkomponenten

In Kapitel 6 werden auf Basis der in Kapitel 5 beschriebenen klinischen Prozesse standardisierte, wieder verwendbare Modellkomponenten entwickelt. Die Modellkomponenten werden in geeigneten Bibliotheken strukturiert zusammengefasst und bilden in ihrer Gesamtheit ein validiertes, klinisches Prozessreferenzmodell. Als Simulationssystem, mit dem die Modellkomponenten entwickelt werden, wird der MLDesigner<sup>178</sup> eingesetzt. Der MLDesigner ist eine integrierte Plattform zur Modellierung und Analyse der Architektur, Funktion und Performance von Systemen. Er besteht u.a. aus einem grafischen Editor (GUI)<sup>179</sup>, einem System zur Analyse und Auswertung von Simulationsergebnissen sowie integrierten Simulatoren, darunter die „discrete event domain“. Diese wird bei der Entwicklung der Modellkomponenten verwendet, da in der späteren Simulation der Tagesklinik<sup>180</sup> diskrete Ereignisse abgebildet werden. Ferner enthält der MLDesigner eine Vielzahl von Bibliotheken und Beispielsystemen [MLD04 S.7f]. Als Modelltypen kommen in dieser Arbeit Systeme, Bibliotheken, Module und Primitive zum Einsatz [MLD04 S.40]. Ferner werden Ports<sup>181</sup>, Relationen und Argumente zur Beschreibung der Modellkomponenten eingesetzt. Module können eine hierarchische Struktur - bestehend aus Submodulen und Primitiven - aufweisen. Ferner können Sie Input-/Output-Ports sowie Argumente besitzen. Module sind im MLDesigner optisch durch ein grünes Dreieck, Primitive durch ein blaues Dreieck - jeweils rechts oben im Symbol - gekennzeichnet. Das „System“ ist die oberste Stufe und enthält unterschiedliche Instanzen von Modulen und Primitiven. Nur Systeme können im MLDesigner simuliert werden. Primitive stellen die unterste Stufe innerhalb der Modelltypen dar und enthalten in C++ Code erstellte Funktionalitäten. Sie können Input-/Output-Ports und Argumente enthalten.

---

<sup>178</sup> MLDesigner = Mission Level Designer.



<sup>179</sup> Siehe Anhang, Abbildung 161.

<sup>180</sup> Siehe hierzu Kapitel 7.

<sup>181</sup> Ports mit orangefarbenen Pfeilen geben Integer-Werte weiter; Ports mit hellblauen Pfeilen geben Datenstrukturen weiter; Ports mit blauen Pfeilen sind vom Typ „Real“; Ports mit roten Pfeilen sind vom Typ „AnyType“.

Bei der Entwicklung der Modellkomponenten wird der strukturierte, modulare Aufbau des Prozessreferenzmodells genutzt, welches in Kapitel 5 dieser Arbeit beschrieben wurde, sowie diverse Standardkomponenten<sup>182</sup> des MLDesigners. Alle Modellkomponenten werden so konzipiert, dass sie für unterschiedliche klinische Simulationsstudien eingesetzt, sowie jederzeit angepasst und ergänzt werden können. Dies ist ein entscheidender Faktor, wenn es um die Wiederverwendbarkeit geht.

## **6.1 Modellkomponenten zur Darstellung des Prozesses „Patientenaufnahme“**

Die Patientenaufnahme beginnt am Empfang einer Klinik. Deshalb wurde im MLDesigner eine Bibliothek mit dem Namen „Empfang“ angelegt. Unter dieser Bibliothek wurden Schritt für Schritt - analog zum Top-Down-Ansatz, der bereits bei der Beschreibung der klinischen Prozesse angewandt wurde - alle weiteren notwendigen Bibliotheken, Module und Primitive, die Teil der Bibliothek „Empfang“ sind, gruppiert. Das Hauptmodul in der o.g. Bibliothek wird mit „Empfang\_Hauptmodul“ (vgl. Abb. 99) bezeichnet. Es enthält die beiden untergeordneten Module „Empfangsbereich“ und „Patientenaufnahme“, einen Datenspeicher (=Memory) mit Namen „PatientenTabelle“ und einen Ressourcenspeicher mit Namen „Pflegekraefte“. Datenspeicher werden im MLDesigner mit dem Symbol , Ressourcenspeicher mit dem Symbol  dargestellt.

---

<sup>182</sup> Z.B. Warteschlangen, Delays, Belegungsstatistiken



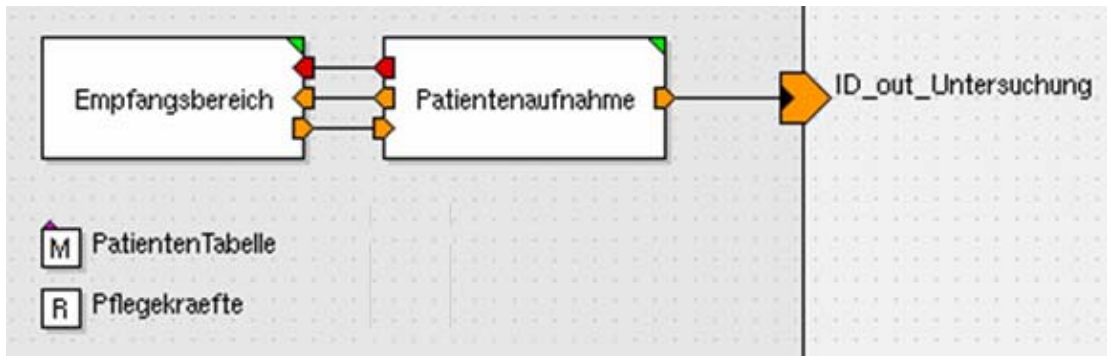


Abbildung 99: Modul „Empfang\_Hauptmodul“

Abbildung 100 zeigt den Aufbau des Moduls „Empfangsbereich“. Dieses Modul liest aus einer Text-Datei<sup>183</sup> alle notwendigen Patienteninformationen ein, darunter die eindeutige Patienten-ID und die jeweilige Ankunftszeit des Patienten. Für jeden Patienten wird ein Vektor erzeugt, dem die entsprechenden Datenfelder zugewiesen werden. Der Vektor wird im Memory abgelegt. Die Werte, die in die Datenfelder geschrieben werden, dienen der späteren Analyse des Systemverhaltens und sind Basis für die Systemoptimierung.<sup>184</sup> Die in Abbildung 100 gezeigte Warteschlange arbeitet nach dem FIFO-Prinzip. Sie dient als eine Art Zwischenpuffer für diejenigen Patienten, die nicht unmittelbar nach Ihrer Ankunft im Empfang aufgenommen werden können. Solche Fälle treten auf, wenn keine Pflegekraft am Empfang verfügbar ist. Warteschlangen sind Primitive, die der MLDesigner standardmäßig in seinen Bibliotheken zur Verfügung stellt.

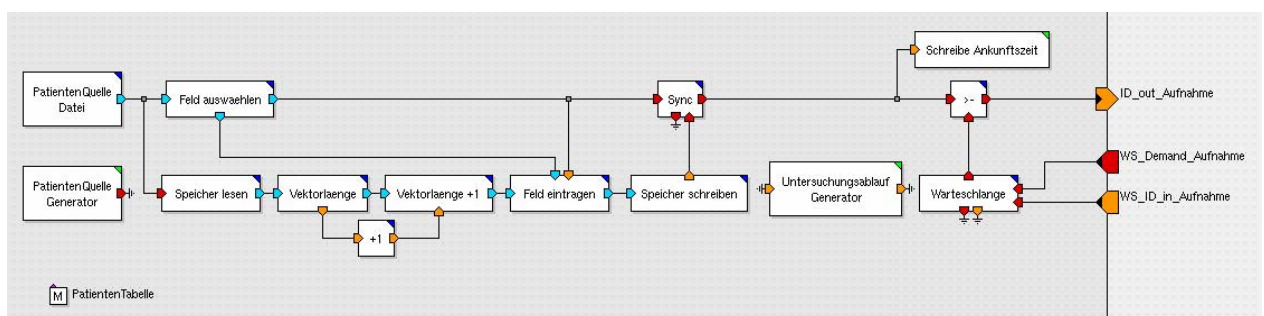


Abbildung 100: Aufbau des Moduls „Empfangsbereich“

Um das Laufzeitverhalten in der späteren Simulation zu verbessern, wird in der weiteren Verarbeitung die Patienten-ID des jeweiligen Vektors – anstatt des

<sup>183</sup> Siehe hierzu auch Kapitel 7.2.1, Abbildung 139.

<sup>184</sup> Siehe hierzu auch Kapitel 7.3 und 7.4.

gesamten Vektors – als Transaktionselement verarbeitet. Durch die Eindeutigkeit der Patienten-ID ist die Zuordnung der Transaktion zum jeweiligen Vektor jederzeit möglich.

Das Modul „Patientenaufnahme“ beinhaltet die Module „Notaufnahme“, „Stationäre Aufnahme“ und „Ambulante Aufnahme“ (vgl. Abb. 101).

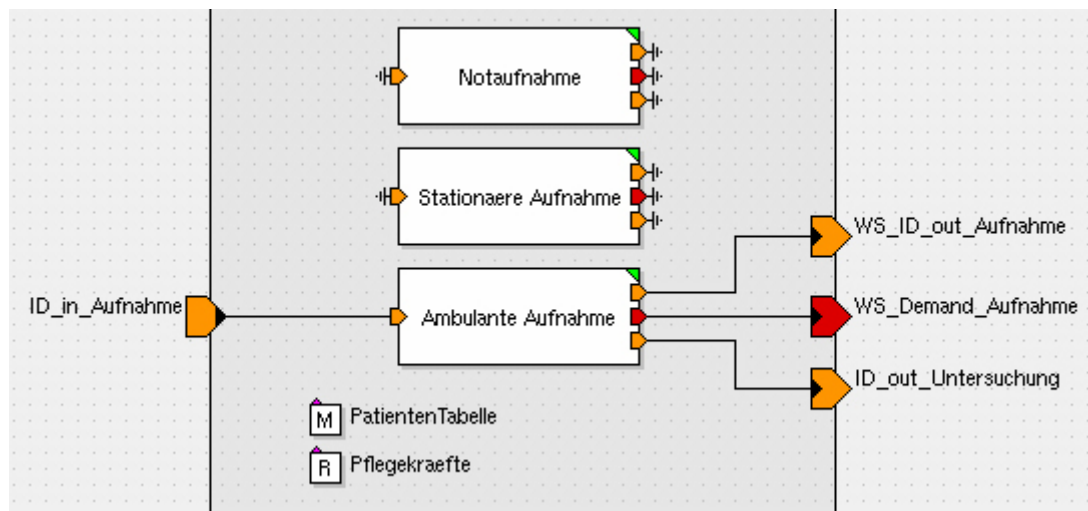


Abbildung 101: Modul „Patientenaufnahme“

Der Aufbau dieser Module entspricht den in Kapitel 3.2.2 dieser Arbeit entwickelten Prozessschritten des Prozessreferenzmodells. Abbildung 102 zeigt den Aufbau des Moduls zur Darstellung der ambulanten Patientenaufnahme.

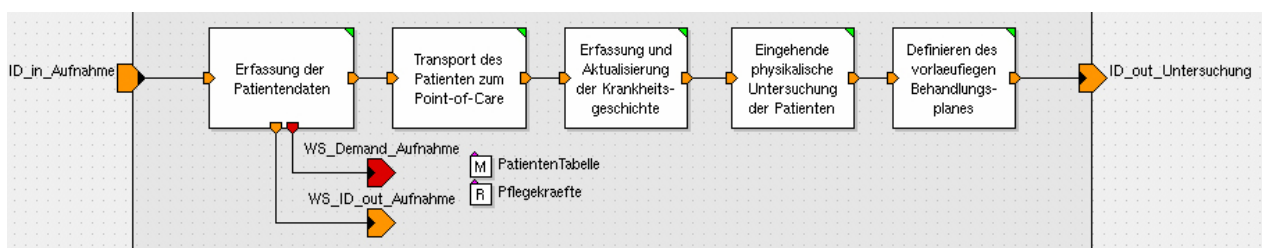


Abbildung 102: Substruktur des Moduls „Ambulante Aufnahme“

Das Modul „Erfassung der Patientendaten“ belegt die Ressource „Pflegerkrafte“ und berechnet die Dauer des Prozessschrittes mittels einer Poissonverteilung (vgl. Abb. 103).<sup>185</sup> Der Zeitverbrauch wird mit dem Primitive „Patientendaten

<sup>185</sup> Modul „Vorgangsdauer generieren“.

erfassen“ - einem Delay - abgebildet. Das Modul „Schreibe Beginn“ schreibt den Zeitpunkt des Beginns der Erfassung der Patientendaten, das Modul „Schreibe Ende“ schreibt den Ende-Zeitpunkt der Erfassung in das entsprechende Datenfeld des zugehörigen Vektors, der - wie oben beschrieben - über seine ID identifiziert wird.

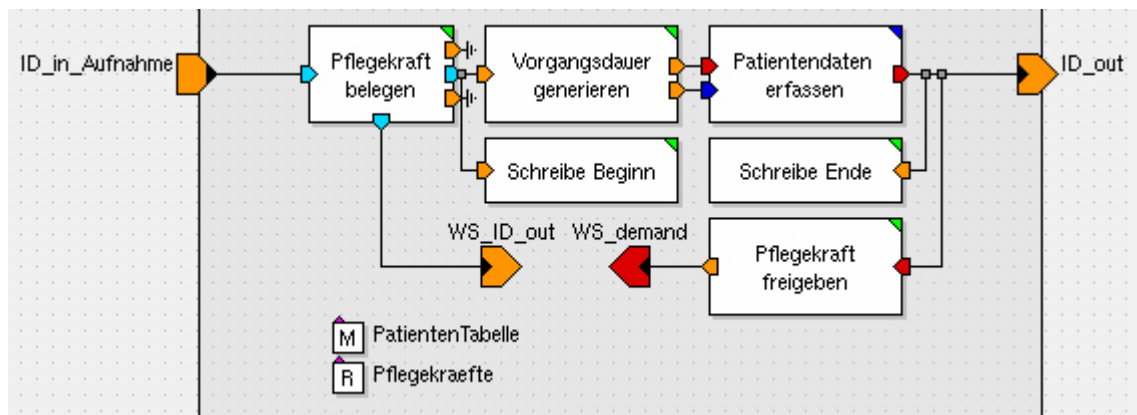


Abbildung 103: Substruktur des Moduls „Erfassung der Patientendaten“

Sind alle Arbeitsschritte erledigt, wird die Ressource „Pflegekraefte“ freigegeben. Alle anderen Module, die zur ambulanten Aufnahme gehören, enthalten die gleiche, eben beschriebene Grundfunktionalität.

Für den Fall, dass bei der Ankunft eines Patienten keine Pflegekraft verfügbar ist, wird die Transaktion über den Port „WS\_ID\_out“ an die Warteschlange<sup>186</sup>, die sich im Modul „Empfangsbereich“ befindet, weitergegeben. Sie verbleibt dort solange, bis eine Pflegekraft verfügbar ist. Dann wird die Transaktion über den Port „WS\_demand“ aus der Warteschlange abgerufen.

Abbildung 104 zeigt die Struktur der Bibliothek „Empfang“, wie sie sich im MLDesigner darstellt.

<sup>186</sup> FIFO-Warteschlange.

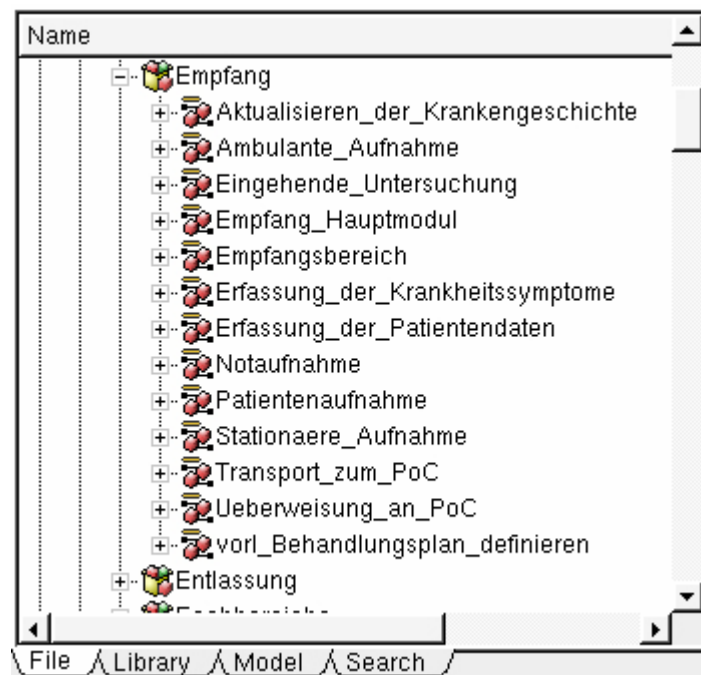


Abbildung 104: Baumstruktur der Bibliothek „Empfang“

## 6.2 Modellkomponenten zur Darstellung des Prozesses „Untersuchung“

Der zweite Hauptprozessschritt im Prozessreferenzmodell ist die Patientenuntersuchung. Für diese Modellkomponente wird im MLDesigner eine Bibliothek mit Namen „Untersuchung“ angelegt. Diese enthält die Module „Untersuchung\_Hauptmodul“, „Voruntersuchungen\_veranlassen“ und „Untersuchung\_aerztlich“ sowie die Bibliothek „Blutabnahme“ (vgl. Abb. 105).

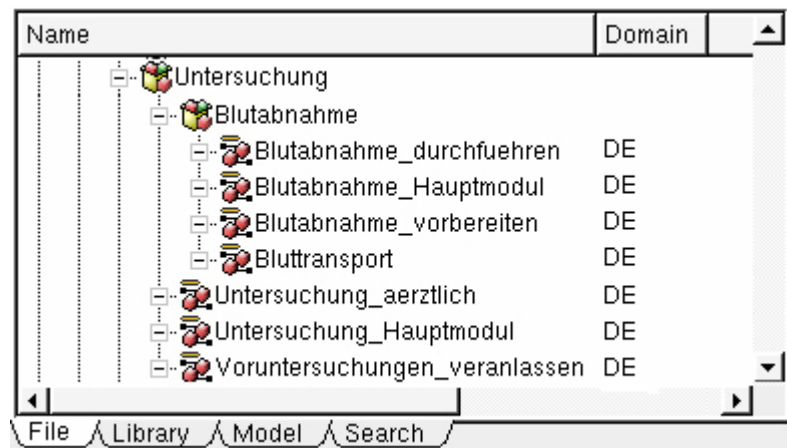


Abbildung 105: Baumstruktur der Bibliothek „Untersuchung“

Das Modul „Blutabnahme\_Hauptmodul“ als Teil der Bibliothek „Blutabnahme“ umfasst die folgenden Module (vgl. Abb. 106):

(1) Blutabnahme vorbereiten (vgl. Abb. 107)

Hier wird die Pflegekraft belegt, die Zeitdauer der Vorbereitung mittels einer Poissonverteilung berechnet<sup>187</sup> und der Zeitpunkt des Beginns und der Endezeitpunkt der Vorbereitung in den zugehörigen Vektor geschrieben. Der Zeitverbrauch wird mit dem Primitive „Blutabnahme vorbereiten“ dargestellt. Dies wurde mit einer Standardkomponente des MLDesigners, einem Delay, umgesetzt. Am Ende des Prozessschrittes wird die Ressource „Pflegekraefte“ freigegeben. Ist keine Pflegekraft verfügbar, wird die Transaktion in einer FIFO-Warteschlange zwischengepuffert.

<sup>187</sup> Modul „Vorgangsdauer generieren“.

(2) Blutabnahme durchfuehren (vgl. Abb. 108)

Der Zeitverbrauch für die Durchführung der Blutabnahme wird analog zu (1) mit dem Modul „Vorgangsdauer generieren“ und dem Primitive „Blutabnahme durchfuehren“ dargestellt. Sobald die Blutabnahme beendet ist, wird der entsprechende Zeitpunkt in den zugehörigen Vektor geschrieben. Ist die Nachbearbeitung - realisiert mit dem Primitive „Nachbereitung“ - abgeschlossen, wird die Ressource „Pflegekraefte“ freigegeben. Fehlende Ressourcen führen dazu, dass anstehende Transaktionen in einer FIFO-Warteschlange zwischengepuffert werden.<sup>188</sup>

(3) Bluttransport (vgl. Abb. 109)

Hat sich eine bestimmte Anzahl von Blutproben angesammelt oder ist ein bestimmtes Zeitintervall verstrichen, werden die Blutproben ins Labor transportiert. Diese Funktion übernimmt das Primitive „Sammelstelle“. Die dafür notwendigen Parameter („Anzahl Blutproben“ bzw. „Zeitintervall“) können vom Benutzer individuell vorgegeben werden. Das Modul „Vorgangsdauer generieren“ errechnet mittels einer Poissonverteilung die Zeit, die für das Versenden der Blutproben benötigt wird. Der Zeitverbrauch wird durch das Primitive „Blutproben transportieren“ dargestellt. Das Schreiben der Anfangs- und Endzeitpunkte übernehmen die Module „Schreibe Beginn“ und „Schreibe Ende“. Analog zu (1) und (2) werden Transaktionen bei Ressourcenengpässen in einer FIFO-Warteschlange zwischengepuffert.

---

<sup>188</sup> Primitive „Warteschlange FIFO“.

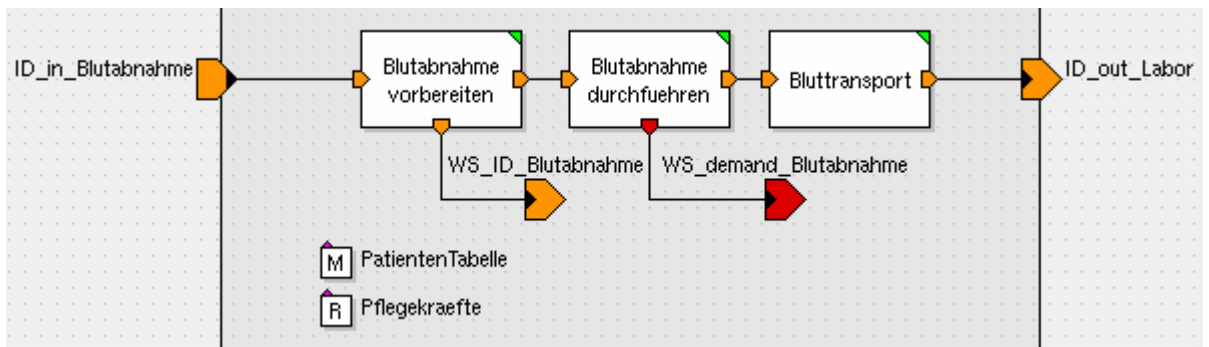


Abbildung 106: Modul „Blutabnahme\_Hauptmodul“

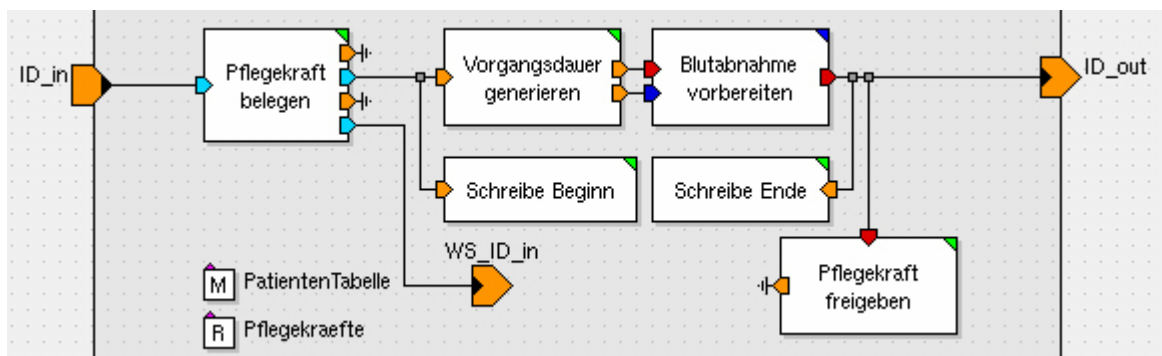


Abbildung 107: Modul „Blutabnahme\_vorbereiten“

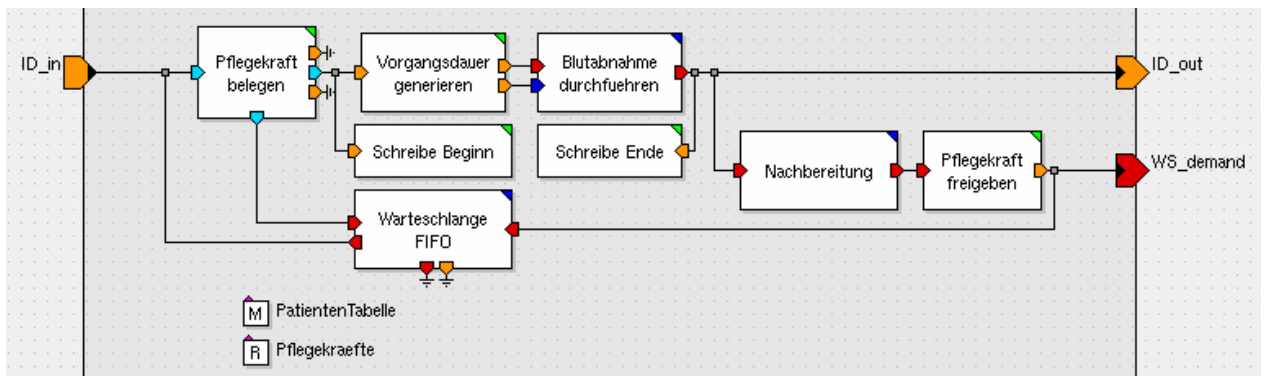


Abbildung 108: Modul „Blutabnahme\_durchfuehren“

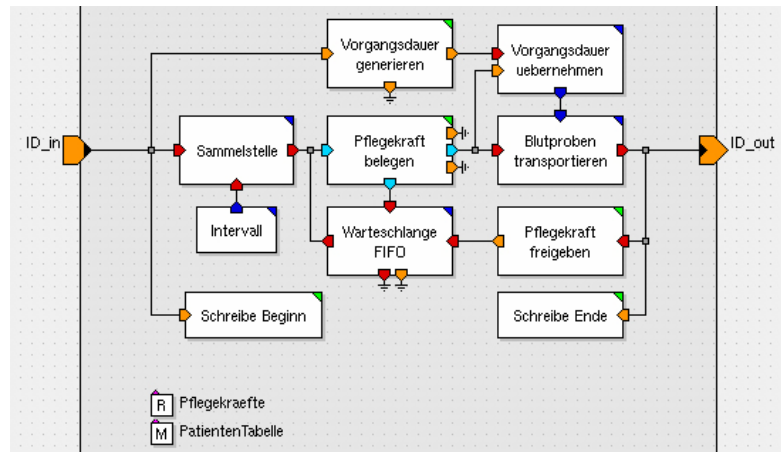


Abbildung 109: Modul „Bluttransport“

Das Modul „Untersuchung\_aerztlich“ dient der Darstellung der allgemeinen ärztlichen Untersuchung. Dieses Modul ist in Abbildung 110 dargestellt.

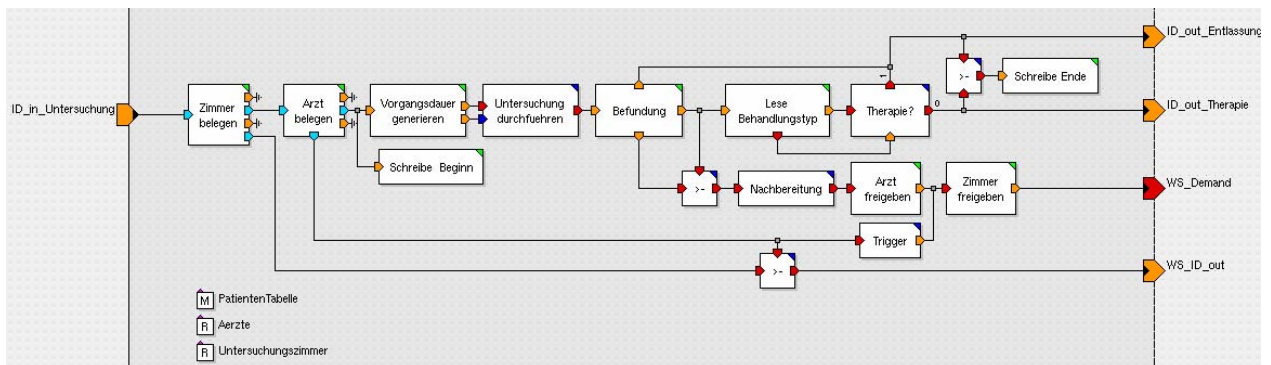


Abbildung 110: Modul „Untersuchung\_aerztlich“

Kommt ein Patient zur ärztlichen Untersuchung, wird zum einen ein Untersuchungszimmer (=Ressource „Untersuchungszimmer“), zum anderen die Ressource „Aerzte“ belegt. Anschließend wird der Beginn der ärztlichen Untersuchung in den zugehörigen Vektor geschrieben, die Zeitdauer der Untersuchung mittels einer Poissonverteilung berechnet<sup>189</sup> und der Zeitverbrauch der Untersuchung durch das Primitive „Untersuchung durchfuehren“ realisiert. Das Modul „Befundung“ legt mittels einer Binomialverteilung fest, ob der Patient in ausreichender physischer wie auch psychischer Verfassung für eine weitere Behandlung ist. Patienten mit unzureichender Verfassung werden direkt zur

<sup>189</sup> Modul „Vorgangsdauer generieren“.



Entlassung weitergeleitet, da sie am betreffenden Tag nicht mehr weiterbehandelt werden. Das Modul „Lese Behandlungstyp“ liest aus dem Vektor der betreffenden Transaktion aus, ob ein Patient eine Therapie durchläuft (0=Therapiepatient) oder nicht (1=Mitbehandler). Das Primitive „Therapie?“ entscheidet dann, an welchen Port die Transaktion weitergegeben wird.<sup>190</sup> Ist die Untersuchung und Nachbereitung beendet, wird sowohl die Ressource „Aerzte“ als auch die Ressource „Untersuchungszimmer“ freigegeben.

Im klinischen Alltag sind i.d.R. mehrere klinische Disziplinen bzw. Abteilungen bei der Behandlung von Patienten involviert. Um diesen Sachverhalt abzubilden, wird zum einen ein Modul angelegt, welches die notwendigen Untersuchungen des jeweils betrachteten Patienten ermittelt (=Modul „Voruntersuchungen\_veranlassen“) (vgl. Abb. 116). Zum anderen wird eine Bibliothek mit Namen „Fachbereiche“ angelegt. Innerhalb dieser Bibliothek werden wiederum die Bibliotheken Computertomographie, Elektrokardiographie, Sonographie, Apotheke und Labor angelegt. Zur Abbildung von Komponenten, die innerhalb der Bibliothek „Fachbereiche“ in mehreren Modulen zum Tragen kommen, wird die Bibliothek „Bausteine\_FB“<sup>191</sup> angelegt. Dies führt zu folgender Struktur (vgl. Abb. 111):

---

<sup>190</sup> Im Falle eines Mitbehandlers „Port 1“, im Falle eines Therapiepatienten „Port 0“.

<sup>191</sup> FB=Fachbereich.

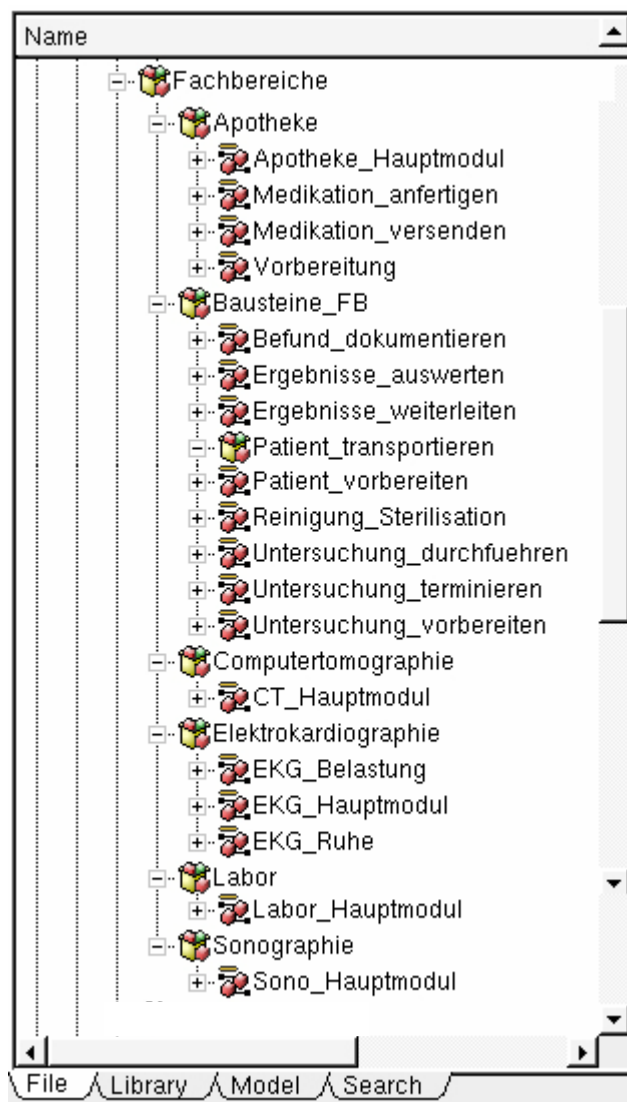


Abbildung 111: Baumstruktur der Bibliothek „Fachbereiche“

Nun werden am Beispiel der Laboruntersuchung und der Computertomographie die jeweiligen Elemente, die zu diesen beiden Untersuchungen gehören, beschrieben. Die Elemente der übrigen Modellkomponenten der Bibliothek „Fachbereiche“ sind in ihrem grundlegenden Aufbau weitgehend identisch, so dass auf deren Beschreibung an dieser Stelle nicht näher eingegangen wird. Die detaillierte Beschreibung dieser Modellkomponenten findet sich im Anhang dieser Arbeit.

Innerhalb der Bibliothek „Labor“ befindet sich das Modul „Labor\_Hauptmodul“ (vgl. Abb. 112).

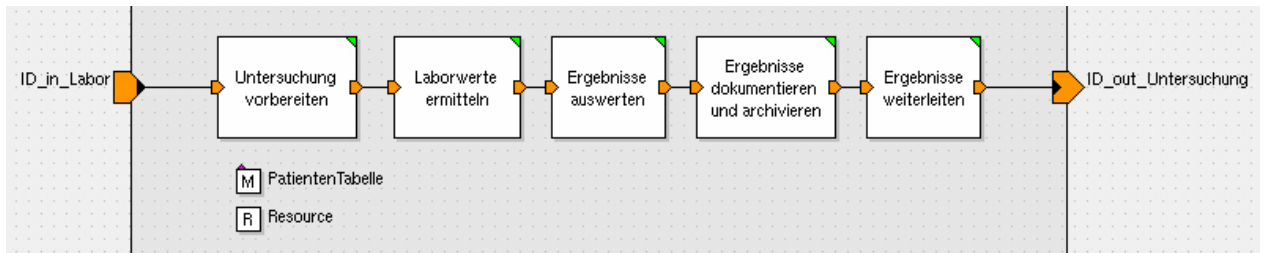


Abbildung 112: Modul „Labor\_Hauptmodul“

Der Aufbau der Module, die der Beschreibung der Komponente „Labor\_Hauptmodul“ dienen, wird am Beispiel des Moduls „Untersuchung\_vorbereiten“ (vgl. Abb. 113)<sup>192</sup> erläutert: Transaktionen werden nach dem FIFO-Prinzip abgearbeitet. Die notwendige Ressource wird belegt, der Beginn der Laboruntersuchung wird in den zugehörigen Vektor geschrieben. Mittels einer Poissonverteilung wird die Zeitdauer für die Vorbereitung der Laboruntersuchung ermittelt. Ist der Ende-Zeitpunkt erreicht, wird dieser in den zugehörigen Vektor geschrieben und die Ressource freigegeben. Das dedizierte protokollieren der Anfangs- und Endzeitpunkte in jedem Modul ermöglicht bei Bedarf eine detaillierte Analyse und Auswertung für jeden einzelnen Prozessschritt.

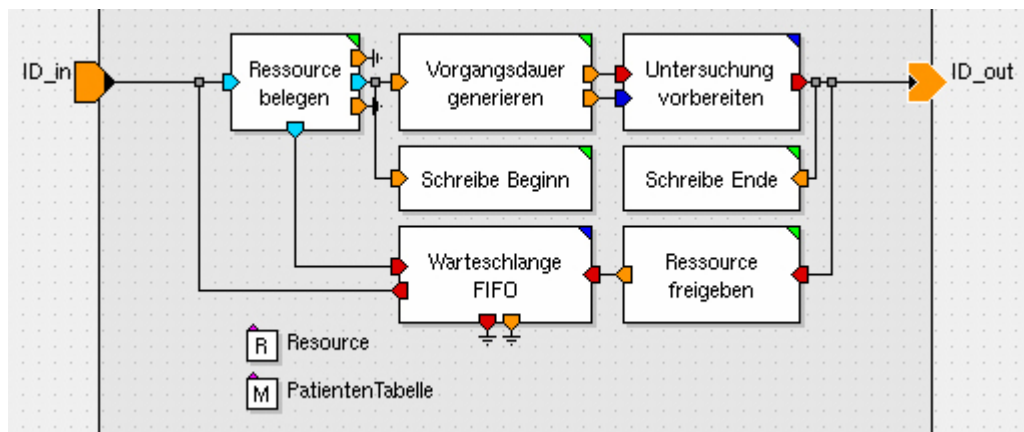


Abbildung 113: Modul „Untersuchung\_vorbereiten“

Die Bibliothek „Computertomographie“ besitzt analog zur Bibliothek „Labor“ ein Hauptmodul, welches mit „Computertomographie\_Hauptmodul“ bezeichnet wurde

<sup>192</sup> Die anderen Module sind analog aufgebaut und im Anhang dieser Arbeit erläutert.

(vgl. Abb. 114). Dieses Modul besteht aus den Modulen „Untersuchung terminieren“, „Untersuchung vorbereiten“, Patient vorbereiten“, Patient transportieren“, „Untersuchung durchfuehren“, Ergebnisse auswerten“, „Befund dokumentieren“ und „Reinigung und Sterilisation“.

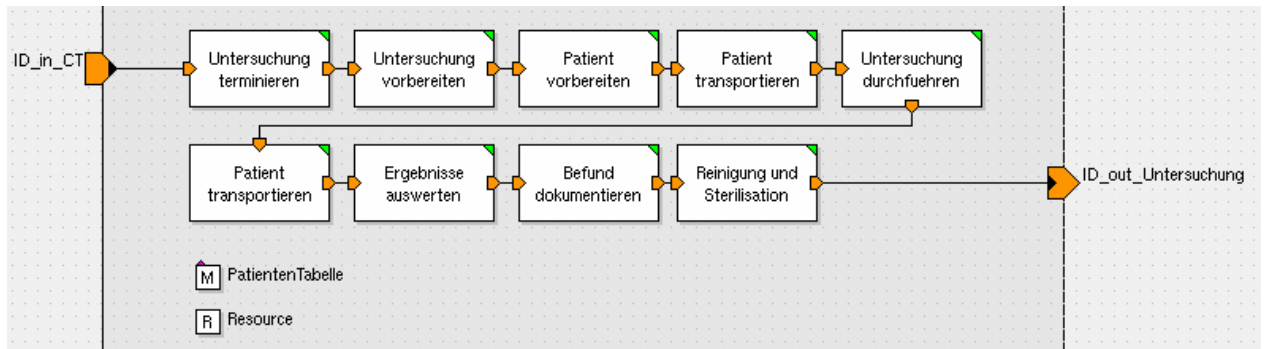


Abbildung 114: Modul „Computertomographie\_Hauptmodul“

Der Aufbau dieser Module ist identisch zu dem in Abbildung 113 gezeigten und erläuterten Modul „Untersuchung vorbereiten“. Auf obige Ausführungen wird verwiesen.

Wie eingangs erwähnt, enthält die Bibliothek „Bausteine\_FB“ diejenigen Modellkomponenten, die innerhalb der einzelnen Fachbereiche mehrfache Verwendung finden – quasi fachbereichsübergreifend sind. Abbildung 115 zeigt die Struktur dieser Bibliothek.

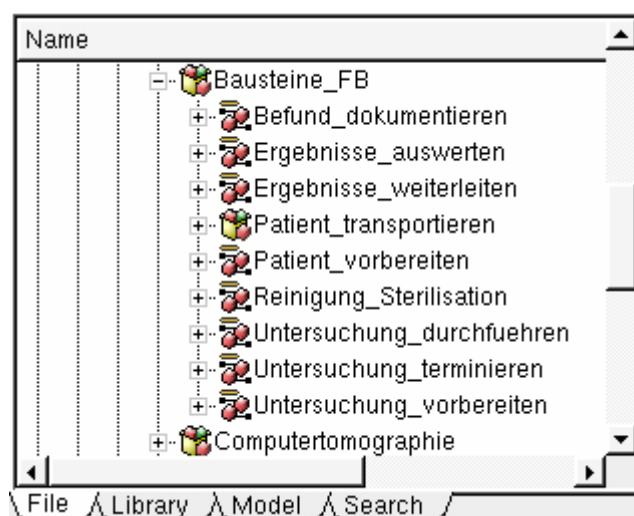


Abbildung 115: Baumstruktur der Bibliothek „Bausteine\_FB“

Die in Abbildung 115 gezeigten Module fanden bei der Entwicklung der Modellkomponenten der medizinischen Fachbereiche mehrfache Verwendung. So wurden sie für die Entwicklung der Module „Computertomographie“, „Sonographie“, „Elektrokardiographie“ und für das Modul „Labor“ eingesetzt.

Wie weiter oben in Abbildung 105 dargestellt, enthält die Bibliothek „Untersuchung“ ein Modul namens „Voruntersuchungen\_veranlassen“. Dieses Modul liest aus der Patiententabelle die Datenfelder aus, die angeben, welche klinischen Untersuchungen ein Patient durchlaufen muss (vgl. Abb. 116).<sup>193</sup> Nur wenn diese Datenfelder mit dem Wert 1 belegt sind, durchläuft der Patient die entsprechende Untersuchung. Als mögliche Untersuchungen kommen in der vorliegenden Arbeit die Computertomographie, die Elektrokardiographie und die Sonographie in Frage. Eine Ergänzung um weitere klinische Bereiche<sup>194</sup> ist durch den modularen Aufbau der einzelnen Modellkomponenten jederzeit möglich.

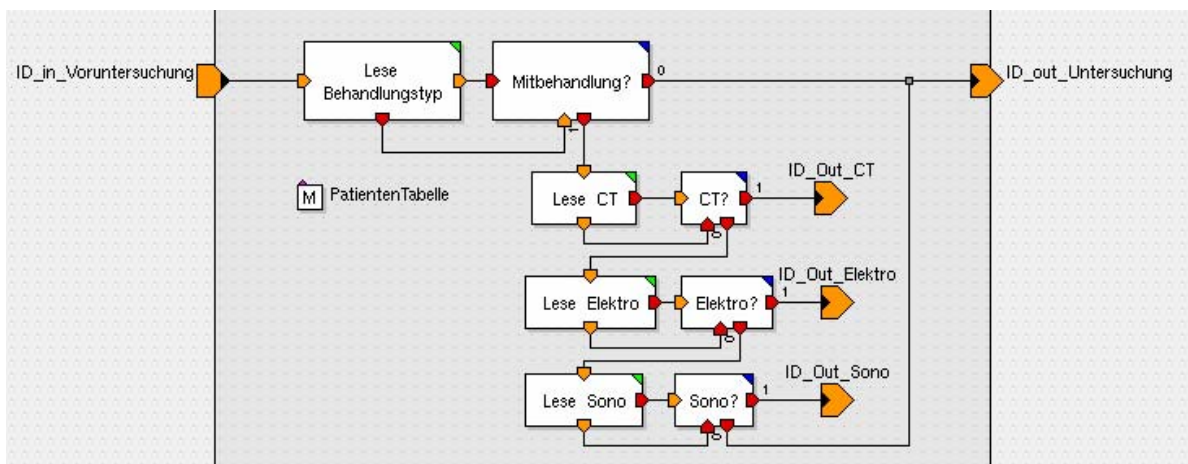


Abbildung 116: Modul „Voruntersuchungen\_veranlassen“

<sup>193</sup> Datenfelder „CTStart“, „ElektroStart“ und „EchoStart“.

<sup>194</sup> Z.B. Chirurgie.

### 6.3 Modellkomponenten zur Darstellung des Prozesses „Therapie“

Zur Abbildung des Hauptprozessschrittes „Patientenbehandlung“ wird die Bibliothek „Therapie“ angelegt. Diese beinhaltet die in Abbildung 117 gezeigten Module.

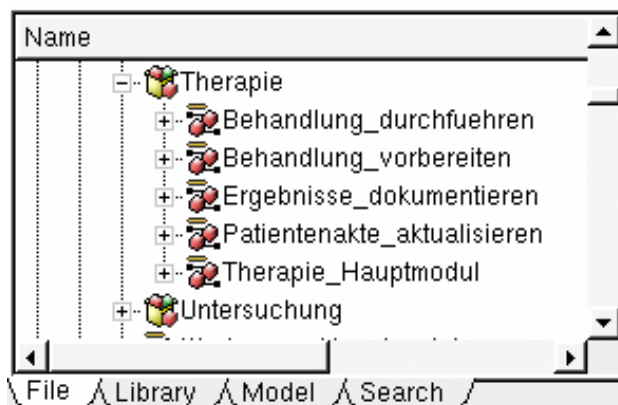


Abbildung 117: Baumstruktur der Bibliothek „Therapie“

Das Modul „Therapie\_Hauptmodul“ umfasst die folgenden Module (vgl. Abb. 118):

#### (1) Behandlung vorbereiten (vgl. Abb. 119)

Das Modul „Behandlung vorbereiten“ übernimmt die Belegung und Freigabe der Ressource „Therapieplaetze“, die Belegung und Freigabe der Ressource „Pflegekraefte“, die Berechnung der Vorbereitungsdauer mittels einer Poissonverteilung, Darstellung des Zeitverbrauchs durch ein Delay (Primitive „Behandlung vorbereiten“) und das Schreiben des Start- und Endzeitpunktes in den zugehörigen Vektor. Ist keine Pflegekraft oder kein Therapieplatz verfügbar, wandert die Transaktion in eine FIFO-Warteschlange.

Der Zeitpunkt des Eintreffens der von der Apotheke gelieferten Medikation wird durch das Modul „Eintreffen der Medikation“ protokolliert.

(2) Behandlung durchführen (vgl. Abb. 120)

Das Modul „Behandlung durchführen“ ist weitgehend analog zum Modul „Behandlung vorbereiten“ aufgebaut. Neu ist, dass der Zeitverbrauch der Behandlungsdurchführung durch zwei Primitive - „Behandlung durchführen“ und „Pflegebetreuung“ - dargestellt wird. Letzteres übernimmt folgende Funktion: Die Belegung einer Pflegekraft muss nicht über den gesamten Zeitraum der Behandlung erfolgen, da z.B. beim Verabreichen einer Infusion die Pflegekraft nicht permanent anwesend ist. Die zeitliche Belegung der Pflegekraft wird deshalb durch das Primitive „Pflegebetreuung“ gesteuert.

Ist für den Prozessschritt keine Ressource verfügbar, wird die Transaktion in der o.g. FIFO-Warteschlange zwischengepuffert.

Nach Ende der Therapie wird der Therapieplatz freigegeben (Modul „Therapieplatz freigegeben“).

(3) Ergebnisse dokumentieren (vgl. Abb. 121)

Das Modul „Ergebnisse dokumentieren“ belegt eine Pflegekraft und gibt diese nach erfolgter Abarbeitung des Prozessschrittes wieder frei. Die Zeitdauer wird mit dem Modul „Vorgangsdauer generieren“ ermittelt und anschließend als Input-Parameter an das Primitive „Ergebnisse dokumentieren“ weitergegeben. Start- und Endzeitpunkt werden durch die Module „Schreibe Beginn“ und „Schreibe Ende“ protokolliert. Analog zu (2) werden Transaktionen in einer FIFO-Warteschlange zwischengepuffert, wenn Ressourcenengpässe bestehen.

(4) Patientenakte aktualisieren (vgl. Abb. 122)

Das Modul „Patientenakte aktualisieren“ ist in seinem Aufbau identisch zu dem unter (3) beschriebenen Modul „Ergebnisse dokumentieren“. Auf die obigen Ausführungen wird verwiesen.

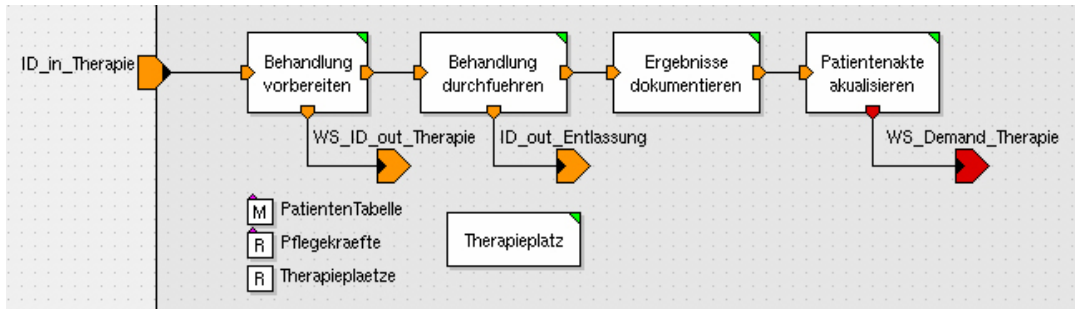


Abbildung 118: Modul „Therapie\_Hauptmodul“

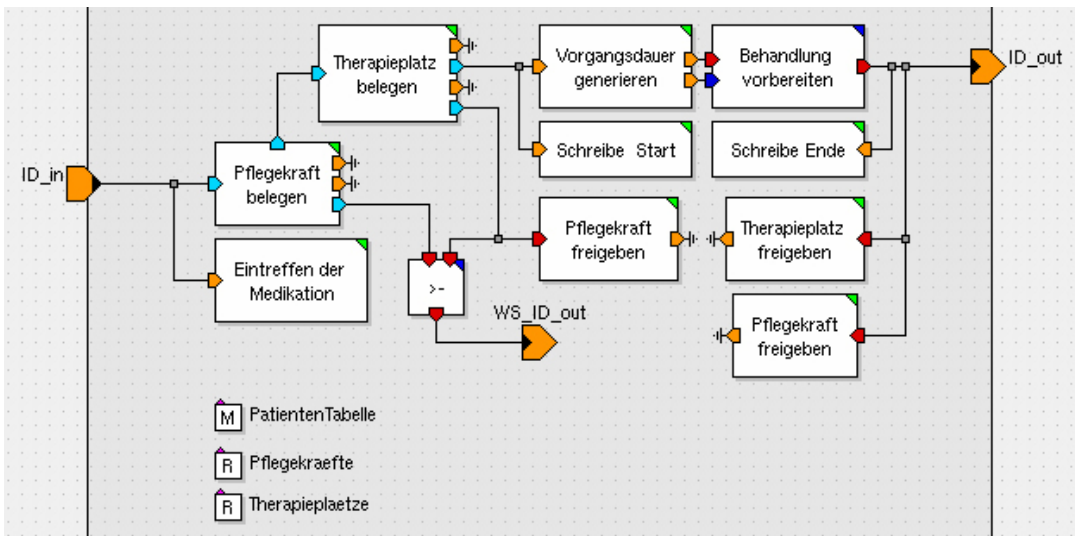


Abbildung 119: Modul „Behandlung\_vorbereiten“

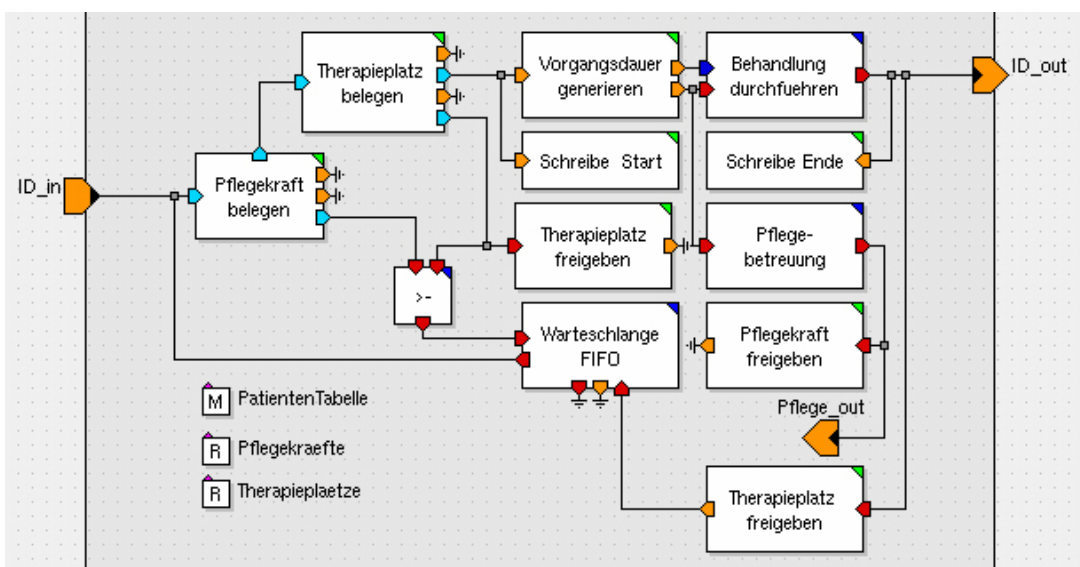


Abbildung 120: Modul „Behandlung\_durchfuehren“



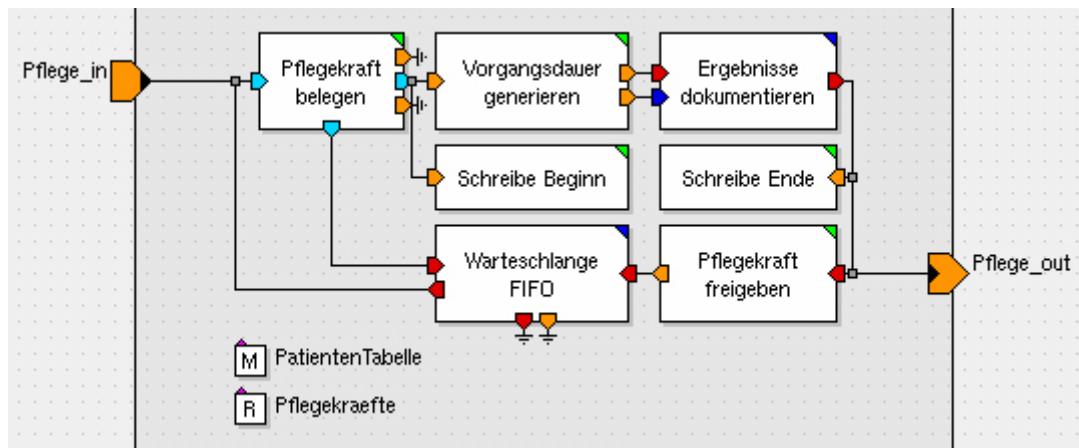


Abbildung 121: Modul „Ergebnisse\_dokumentieren“

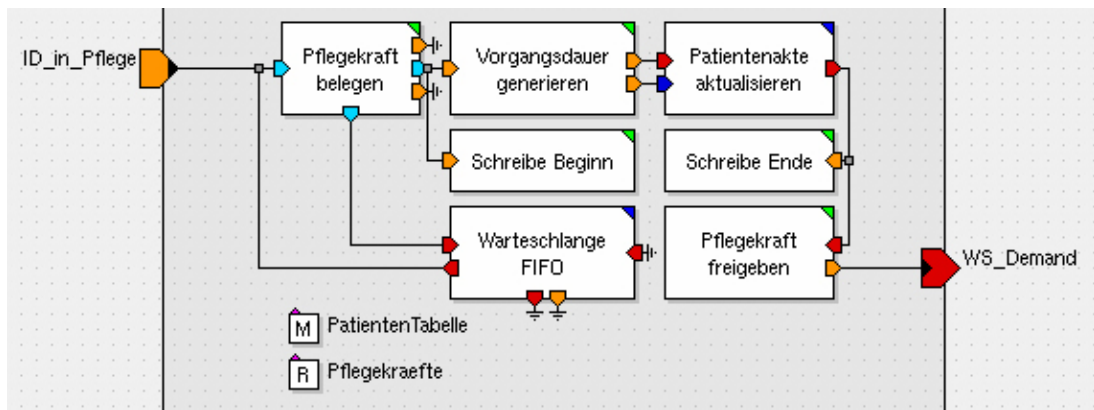


Abbildung 122: Modul „Patientenakte\_aktualisieren“

## (5) Therapieplatz

Zur Darstellung der Belegung der Therapieplätze im Zeitablauf wird eine statistische Funktion genutzt, die als Standard im MLDesigner verfügbar ist.<sup>195</sup> Letzteres ist der Grund, warum dieses Modul nicht in der Bibliothek „Therapie“ erscheint, dennoch zum Modul „Therapie\_Hauptmodul“ gehört. Das Modul „Therapieplatz“ erzeugt eine Grafik, wie sie in der nachfolgenden Abbildung 123 beispielhaft dargestellt ist.

<sup>195</sup> Modul „QrGraph“.

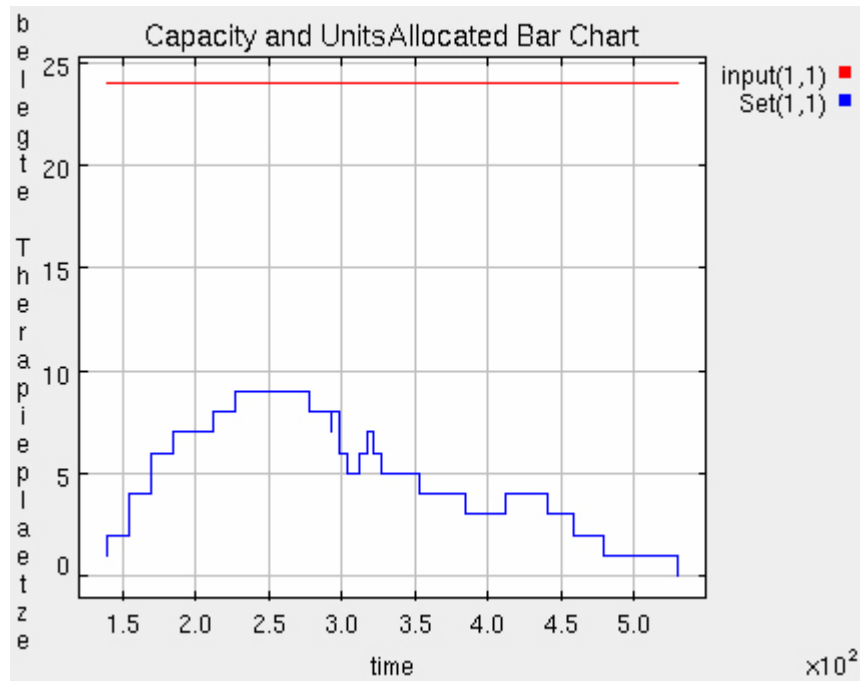


Abbildung 123: Durch das Modul „Therapieplatz“ erzeugte Grafik

An der x-Achse ist die Zeit, an der y-Achse die Anzahl belegter Therapieplätze/-zimmer abgetragen. Diese Auswertung ist entscheidend, um den Grad der Nutzung der Therapieplätze analysieren und gegebenenfalls zielgerichtet optimieren zu können.

#### 6.4 Modellkomponenten zur Darstellung des Prozesses „Patientenentlassung“

Das Modul „Entlassung\_Hauptmodul“ (vgl. Abb. 124) besteht im Wesentlichen aus zwei Delays, welche die Dauer des Schreibens eines Arztbriefes und die Wartezeit bis zum Aushändigen des Arztbriefes abbilden, sowie aus einem Funktionsblock, welcher den jeweiligen Datensatz mit allen Werten, die während des Simulationslaufs für die betreffende Transaktion protokolliert wurden, in eine Textdatei schreibt. Die Dauer des Schreibens des Arztbriefes sowie die Zeit, die bis zur Aushändigung des Arztbriefes vergeht, werden mittels einer Poissonverteilung errechnet. Um festzuhalten, welche Transaktionen (=Patienten) das Modell durchlaufen haben, werden die Patienten-IDs sowie die

Endzeitpunkte der Transaktionen in einem Ausgabefenster des MLDesigners aufgelistet (Primitive „Text“).<sup>196</sup>

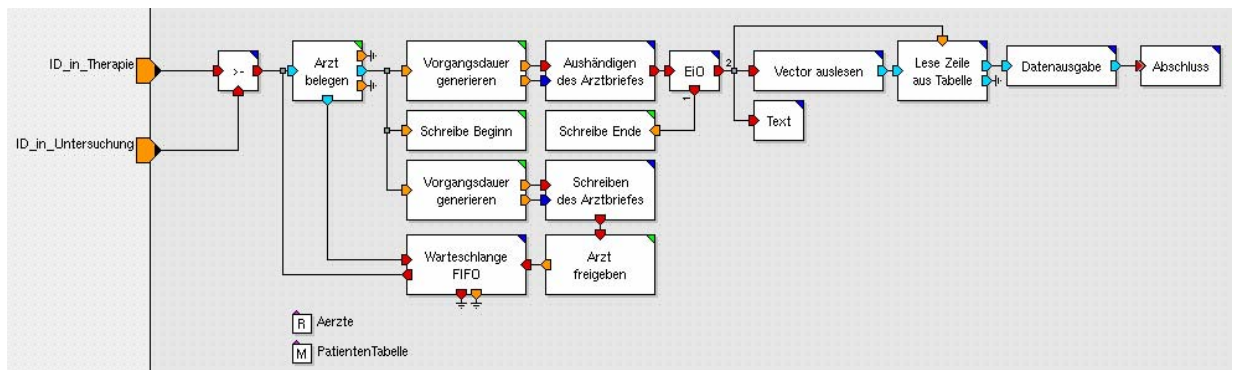


Abbildung 124: Komponenten des Moduls „Entlassung\_Hauptmodul“

<sup>196</sup> Siehe Anhang 1, Abbildung 175.

## **7 Entwicklung des Simulationsmodells und Durchführen der Simulation**

In diesem Kapitel wird das Modell einer medizinisch-onkologischen Tagesklinik unter Verwendung der entwickelten und in Kapitel 6 beschriebenen Modellkomponenten erstellt.

Im ersten Schritt werden die Ist-Prozesse der Tagesklinik aufgenommen, analysiert und in Abstimmung mit dem Personal der Tagesklinik im Modell nachgebildet. Die korrekte Funktionsweise des Modells wird durch Vergleich der Ergebnisse der Simulationsläufe mit den im Realsystem ermittelten Werten überprüft. Ausgehend von der konkreten Modellzielsetzung werden die optimierten Soll-Prozesse modelliert, die Ergebnisse plausibilisiert und der Grad der Zielerreichung überprüft. Als Modellierungs- und Simulationssoftware kommt der MLDesigner, mit dem bereits die Modellkomponenten entwickelt wurden<sup>197</sup>, zum Einsatz.

### **7.1 Ausgangssituation und Zielsetzung des Modells**

Die medizinisch-onkologische Tagesklinik dient der ambulanten chemotherapeutischen Behandlung von Patienten mit Krebserkrankungen. Die betrachtete Tagesklinik hat drei Untersuchungszimmer, ein Zimmer für die Blutentnahme und einen Behandlungsraum für die Verabreichung der Chemotherapien. Dieser ist mit 24 Liegen ausgestattet. Täglich werden durchschnittlich 34 Patienten untersucht und teilweise ambulant chemotherapiert. Patienten, die am Tag der Einbestellung keine Chemotherapie bekommen, werden als Mitbehandler bezeichnet. Das Personal besteht aus vier Pflegefachkräften sowie drei Ärzten.

Alle Patienten werden i.d.R. zwischen 7:30 Uhr und 8:30 Uhr des jeweiligen Arbeitstages einbestellt. Die Patienten melden sich bei der Pflegefachkraft am Empfang. Diese weist den Therapiepatienten einen Behandlungsplatz zu. Sowohl

---

<sup>197</sup> Siehe hierzu auch Kapitel 6.

für Therapiepatienten als auch für Mitbehandler wird die Uhrzeit des Besuches sowie der Name des Patienten und seine Patientenummer<sup>198</sup> in das System SAP IS-H MED<sup>199</sup> eingetragen. Die Krankenakte des jeweiligen Patienten wird am Empfang bereitgestellt. Bevor der Patient vom zuständigen Arzt körperlich untersucht wird, erfolgt die Blutabnahme. Diese wird von den Pflegefachkräften, teils auch von den Ärzten, durchgeführt. Die entnommenen Blutproben werden am Tresen des Empfangsbereichs zur Abholung bereitgestellt und in unregelmäßigen Abständen ins Labor gebracht. Nach Erhalt der Laborergebnisse ruft der zuständige Arzt den Patienten zur Untersuchung auf, dokumentiert und archiviert den Befund.

Nach erfolgter Untersuchung sowie Auswertung der Laborergebnisse entscheidet der Arzt, ob die Therapie durchgeführt werden kann oder nicht. Wird die Therapie durchgeführt, ruft der Arzt telefonisch die bereits am Vortag avisierten Therapien in der Apotheke ab, informiert die Pflegefachkraft mündlich über die Freigabe der Therapie und dokumentiert die Freigabe mit seiner Unterschrift auf dem Therapieplan. Anschließend wird die Prämedikation und Wässerung vorbereitet und dem Patienten angehängt.

Pro Patient werden ein Tablett mit der Chemotherapie sowie der Therapieplan vorbereitet. Auf dem Therapieplan wird die Kontrolle der Therapie dokumentiert. Eine Pflegefachkraft bereitet die Therapien vor und hängt sie dem Patienten an. Der Therapiebeginn wird mit entsprechender Uhrzeit in einem Dokumentationsbogen eingetragen. Während der Therapie wird der Patient von den Pflegefachkräften überwacht. Der Therapieverlauf wird dokumentiert, das Therapieende wird mit Uhrzeit auf dem Dokumentationsbogen eingetragen. Die Therapiedauer, also das Verabreichen der Chemotherapien, liegt etwa zwischen 5 Minuten und 5 Stunden – je nach Art der Gabe (z.B. Bolus<sup>200</sup> oder Infusion; Zytostatika oder Antikörper<sup>201</sup>) und Menge der Gabe. Nach erfolgter Therapie wird die Patientenakte aktualisiert.

Die ersten Chemotherapien beginnen aufgrund der aktuellen Prozessabläufe i.d.R. nicht vor 10:00 Uhr. Bis dahin sind die Patientenliegen im

---

<sup>198</sup> Die Patientenummer ist ein eindeutiges Identifikationsmerkmal.

<sup>199</sup> Ein Produkt der Firma SAP AG Walldorf.

<sup>200</sup> Bolus-Gabe: Bolus bezeichnet die impulsartige Gabe eines Medikamentes.

<sup>201</sup> I.d.R. dauert die Gabe von Antikörpern im Vergleich deutlich länger als das Verabreichen von Zytostatika.

Behandlungsraum der Tagesklinik nicht belegt, obwohl die ersten Therapiepatienten bereits um 7:30 Uhr eintreffen. Die letzten Patienten verlassen die Tagesklinik etwa zwischen 17:00 Uhr und 17:30 Uhr. Offiziell schließt die Tagesklinik um 17:00 Uhr.

Die beiden nachfolgenden Auswertungen, die auf Basis einer umfassenden Datenerhebung in der Tagesklinik entstanden sind, zeigen den repräsentativen Verlauf der Liegenbelegung (vgl. Abb. 125) sowie die durchschnittliche, maximale und minimale Wartezeit der Patienten am Beispiel ausgewählter Behandlungstage (vgl. Abb. 126).

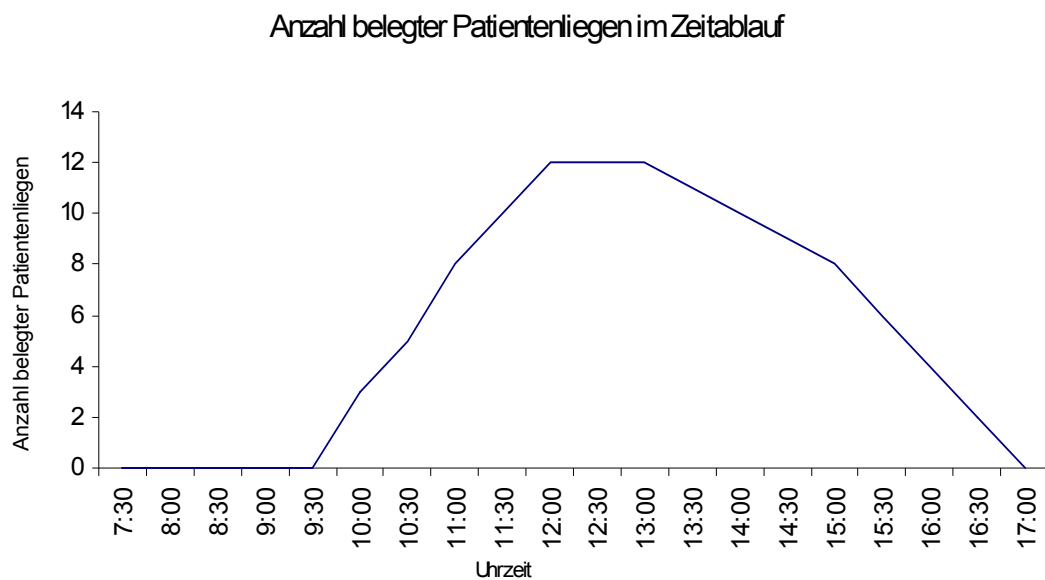


Abbildung 125: Repräsentative Belegungsstatistik der Patientenliegen im Behandlungsraum

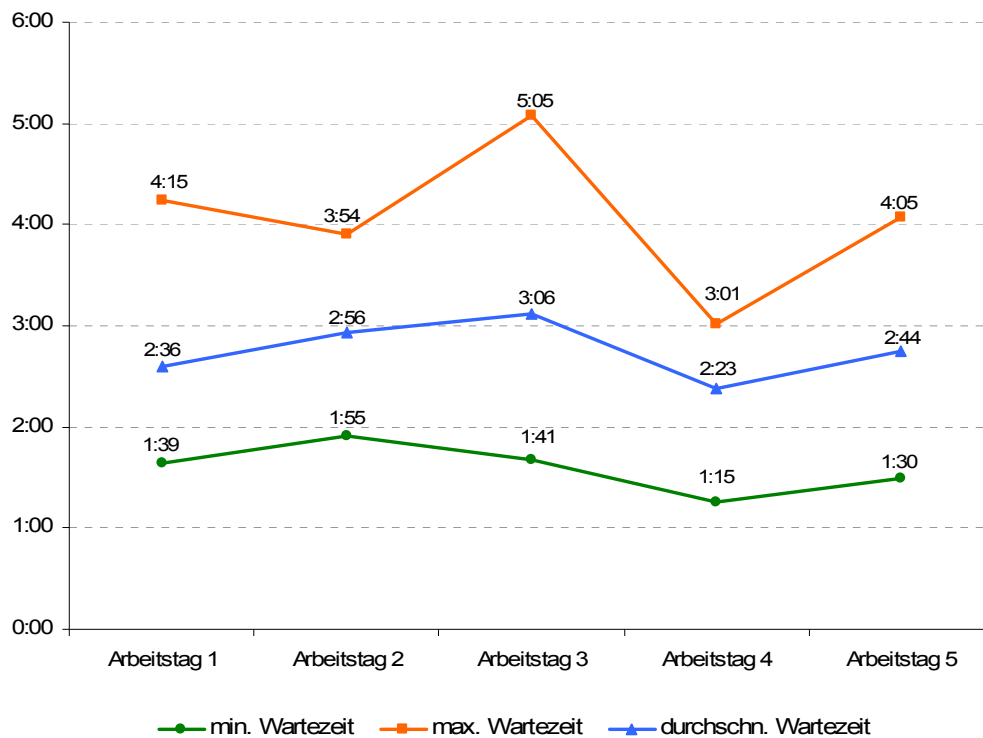


Abbildung 126: Verteilung der Wartezeiten (in Std.) am Beispiel ausgewählter Behandlungstage

Im Zeitraum der Datenerfassung<sup>202</sup> wurde als längste Wartezeit 5 Stunden 5 Minuten gemessen, die kürzeste Wartezeit betrug 55 Minuten, die durchschnittliche Wartezeit lag bei 2 Stunden 45 Minuten.

Abbildung 127 zeigt über einen Zeitraum von einem Quartal die Verteilung der Anzahl der Patienten, die pro Tag einbestellt wurden. Die Anzahl der einbestellten Patienten je Arbeitstag schwankte im Erhebungszeitraum zwischen 18 und 49.

<sup>202</sup> Datenerhebung erfolgte über 55 Arbeitstage; Muster des Datenerhebungsbogens im Anhang, Tabelle 8.

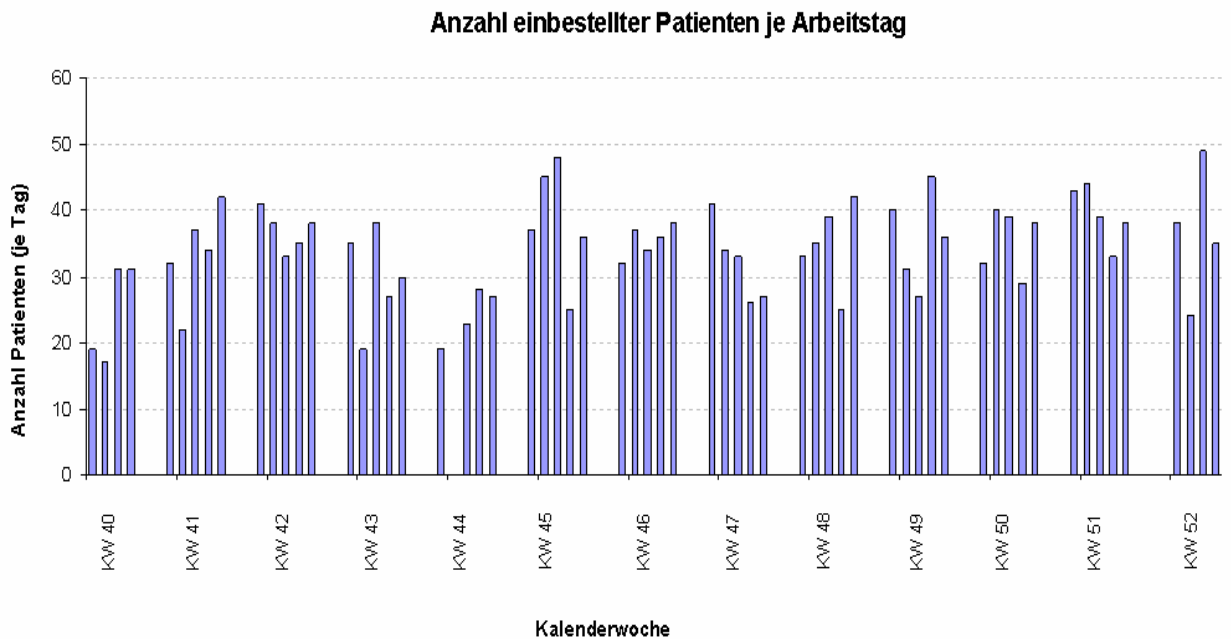


Abbildung 127: Verteilung der Patientenanzahl am Beispiel ausgewählter Behandlungstage<sup>203</sup>

Der detaillierte Ablauf der Ist-Prozesse ist im Anhang in den Abbildungen 162 bis 169 in Form von Ablaufdiagrammen dargestellt.

Die Zielsetzung der Leitung der Tagesklinik ist es...

- (1) ...den Beginn der Therapien zeitlich nach vorne zu legen (als Vorbereitung auf einen Mehrschichtbetrieb).
- (2) ...die durchschnittliche Wartezeit je Patient zu reduzieren und damit die durchschnittliche Gesamtbehandlungsdauer zu verkürzen.
- (3) ...die Öffnungszeit der Klinik von 7:30 Uhr bis 17:00 Uhr auf 07:30 Uhr bis 16:00 Uhr zu verkürzen.

Diese Ziele gelten damit für die in der vorliegenden Arbeit durchzuführende Simulationsstudie.

<sup>203</sup> Fehlende Balken repräsentieren Feiertage.



## 7.2 Aufbau des Simulationsmodells

Bei der Entwicklung des Modells wurde schrittweise und in enger Abstimmung mit dem Personal der Tagesklinik vorgegangen. Dies hat den Vorteil, dass in jedem Entwicklungsschritt die korrekte Funktionsweise der Modellkomponenten sichergestellt werden kann und am Ende des Modellentwicklungsprozesses keine ungewollten Überraschungen<sup>204</sup> auftreten. Das schrittweise Vorgehen hilft ferner, den Komplexitätsgrad bei der Entwicklung des Modells besser zu beherrschen. Das modular aufgebaute Modell wurde sukzessive verfeinert, um eine ausreichende Abbildungsgenauigkeit des Realsystems zu erhalten.

Die Abläufe in der Tagesklinik werden in die Hauptprozessschritte „Patienten aufnehmen“, „Patienten untersuchen“, „Patienten behandeln“ und „Patienten entlassen“ eingeteilt. Dies entspricht der Einteilung des Prozessreferenzmodells in Kapitel 3.2.2 der vorliegenden Arbeit. Bei der Patientenaufnahme handelt es sich im Fall der Tagesklinik um die Kategorie der ambulanten Aufnahme. Zur Abbildung des Hauptprozessschritts „Patientenuntersuchung“ werden die entwickelten Modellkomponenten Laboruntersuchung, Sonographie, Elektrokardiographie und Computertomographie sowie die Komponente zur Abbildung des Prozessschritts der körperlichen Untersuchung des Patienten<sup>205</sup> benötigt. Die Behandlung der Patienten wird mit der entwickelten Modellkomponente „Therapie“ abgebildet. Ferner wird die Modellkomponente „Apotheke“ benötigt, um die Prozessschritte der Herstellung und Auslieferung der Medikation zu modellieren. Zur Darstellung der Patientenentlassung wird der in Kapitel 6.4 beschriebene Baustein eingesetzt.

Bezüglich der zum Modell gehörigen Subprozesse wird - soweit nicht im vorliegenden Kapitel erläutert - auf die Kapitel 3.2.2, 5.3 und 6 sowie auf die Abbildungen 162 bis 169 im Anhang dieser Arbeit verwiesen.

Im Folgenden wird Schritt für Schritt das Simulationsmodell entwickelt. Auf Basis der weiter oben dargestellten Prozessbausteine sowie der beschriebenen

---

<sup>204</sup> Z.B. mangelnde Akzeptanz des Modells seitens des Klinikpersonals oder nicht korrekte Abbildung der betrachteten Prozesse.

<sup>205</sup> Z.B. durch Auskultation (Abhören der im Körper entstehenden Geräusche mit dem Stethoskop), Untersuchen nach allergischen Reaktionen oder Infekten, Messen des Blutdrucks.

Ausstattung der Tagesklinik<sup>206</sup> ergibt sich ein erstes Modelllayout, welches in Abbildung 128 dargestellt ist.

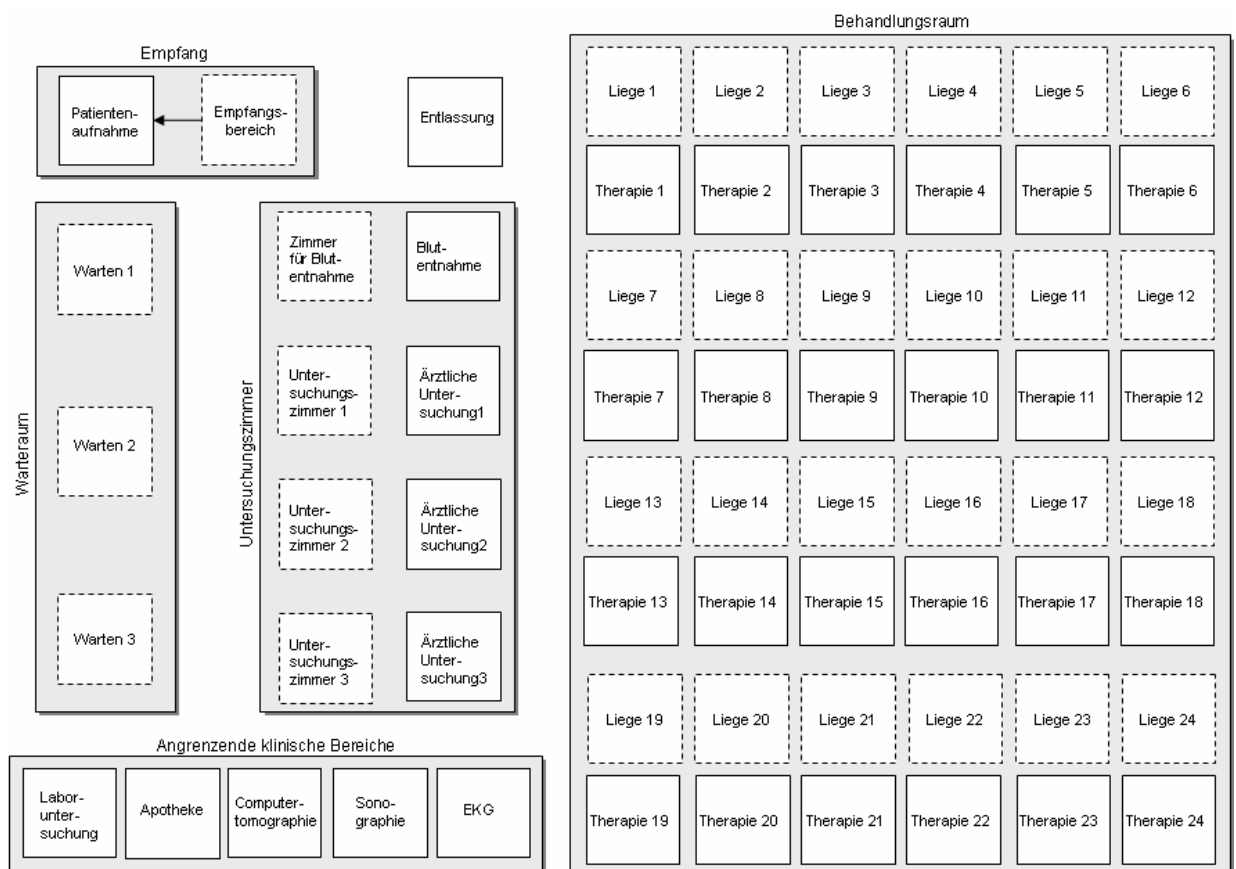


Abbildung 128: Grundmodell (Layout) der onkologischen Tagesklinik<sup>207</sup>

Die Unterteilung des Wartebereiches hat dabei folgenden Hintergrund: Patienten müssen i.d.R. vor jedem Behandlungsschritt Wartezeiten in Kauf nehmen. Um diese Wartezeiten dediziert messen zu können, wurde der Warteraum aufgeteilt. In der betrachteten Tagesklinik warten die Patienten ...

- (1) ... bis sie zur Blutentnahme kommen („Warten 1“),
- (2) ... bis sie die einzelnen Untersuchungen durchlaufen („Warten 2“) und
- (3) ... bis die eigentliche Behandlung/Therapie beginnt („Warten 3“).

<sup>206</sup> Siehe hierzu Kapitel 7.1.

<sup>207</sup> Weiße Quadrate mit gestrichelten Linien stellen Buffer dar, weiße Quadrate mit durchgezogenen Linien repräsentieren Automaten.

Abbildung 129 zeigt das Modell der Tagesklinik auf der obersten Abstraktionsebene. Es wurde mit dem MLDesigner unter Einsatz der in Kapitel 6 dargestellten, wieder verwendbaren Modellbausteine entwickelt.

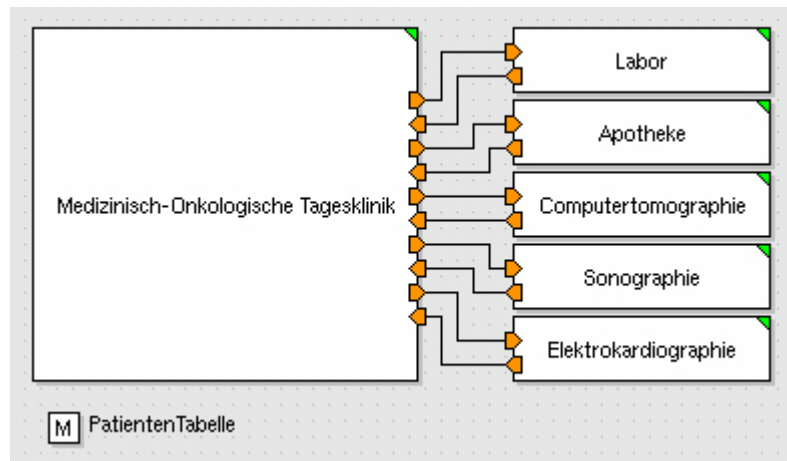


Abbildung 129: Modell der onkologischen Tagesklinik und angrenzender Bereiche (Abstraktionsebene 1)

Die Relationen zwischen den Elementen werden durch Ein- und Ausgangsporte dargestellt.

Abbildung 130 zeigt die Modellstruktur der Tagesklinik mit den zugehörigen Teilbereichen. Die Verbindung zu den angrenzenden Bereichen (Labor, Apotheke, CT, Sonographie und EKG) erfolgt über die entsprechenden Ein- und Ausgangsporte<sup>208</sup>.

<sup>208</sup> Z.B. „ID\_out\_CT“; „ID\_in\_CT“.

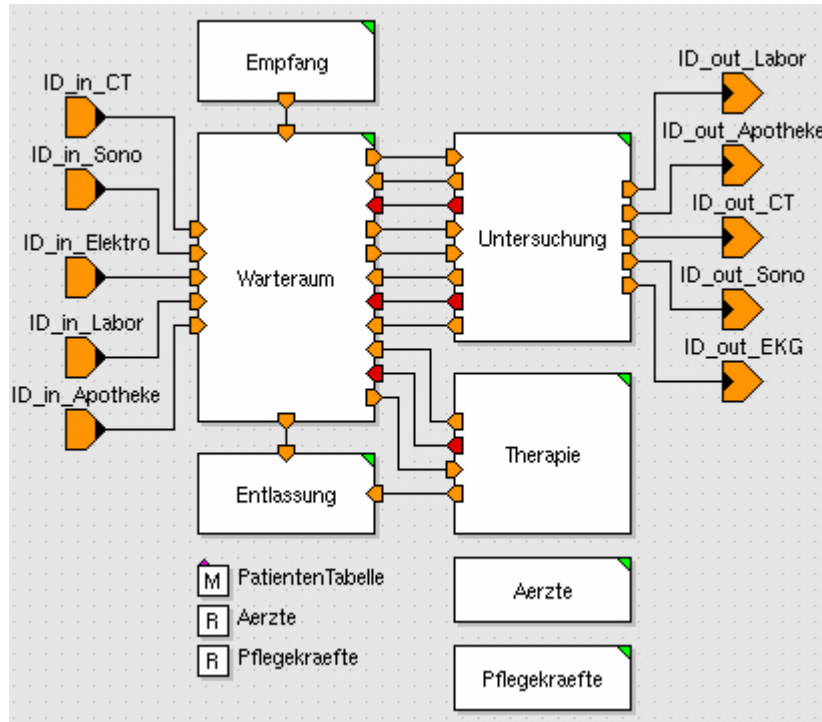


Abbildung 130: Modell der Tagesklinik (Abstraktionsebene 2)

Die Module „Ärzte“ und „Pflegekräfte“ erzeugen eine grafische Auswertung der Ressourcenbelegung, wie sie in Abbildung 131 am Beispiel der Ärzte dargestellt ist.

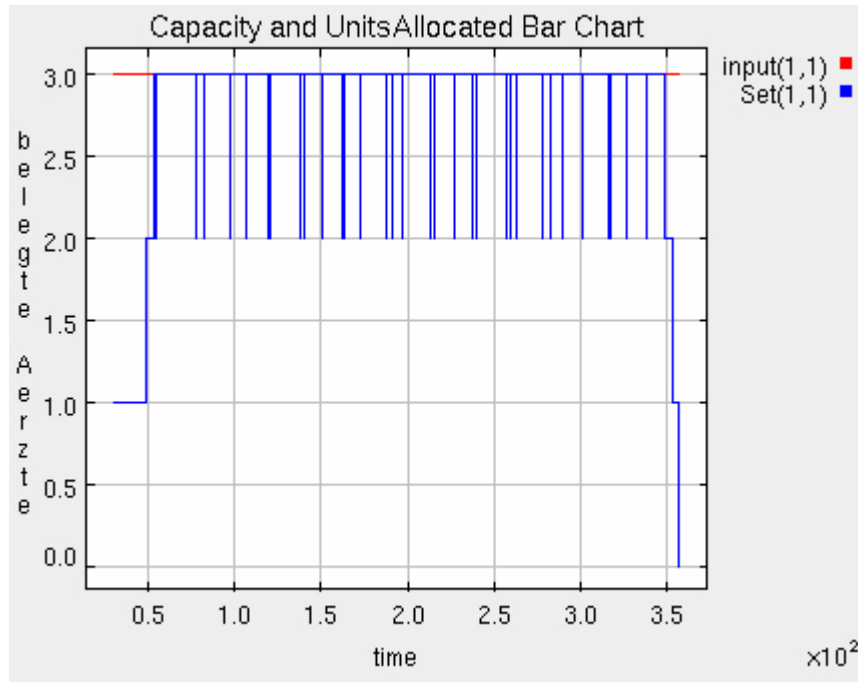


Abbildung 131: Belegungsstatistik am Beispiel der Ressource „Aerzte“

In der betrachteten Tagesklinik werden Patienten ausschließlich ambulant aufgenommen. Deshalb wird hier nur die Modellkomponente zur Abbildung der ambulanten Patientenaufnahme benötigt. Abbildung 132 zeigt die Modellkomponente „Patientenaufnahme“ als Teil des Empfangs. Wie hier zu sehen ist, sind die Ports der Module „Notaufnahme“ und „Stationaere Aufnahme“ nicht mit anderen Komponenten verbunden und damit inaktiv. Lediglich die Eingangs- und Ausgangsports des Moduls „Ambulante Aufnahme“ haben eine Verbindung zu anderen Komponenten des Modells, so z.B. zum „Empfangsbereich“ über den Port „ID\_in\_Aufnahme“ und zur „Untersuchung“ über den Port „ID\_out\_Untersuchung“.

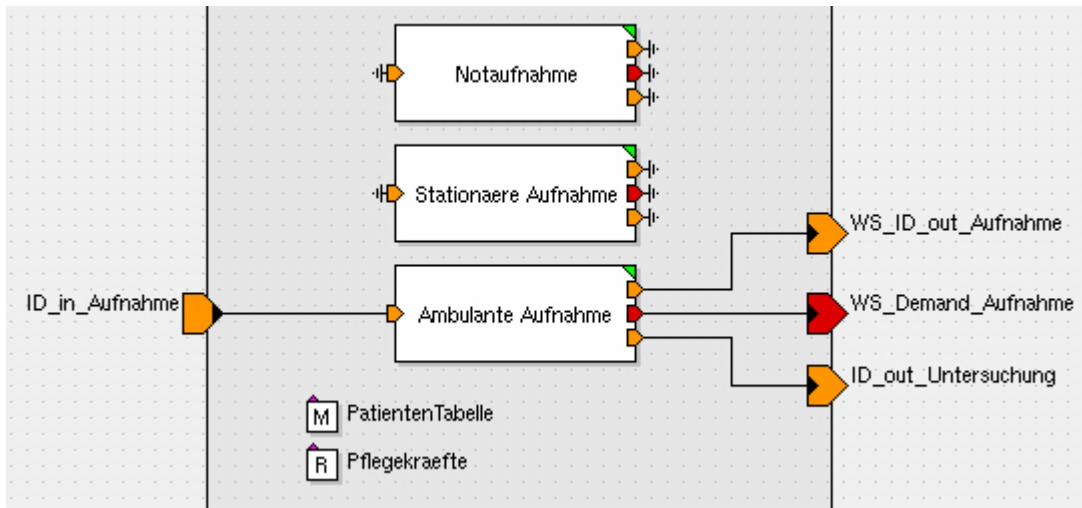


Abbildung 132: Modellkomponente „Patientenaufnahme“

Abbildung 133 zeigt die Prozessschritte der ambulanten Patientenaufnahme, beginnend mit der Erfassung der Patientendaten bis zur Definition des vorläufigen Behandlungsplans<sup>209</sup>

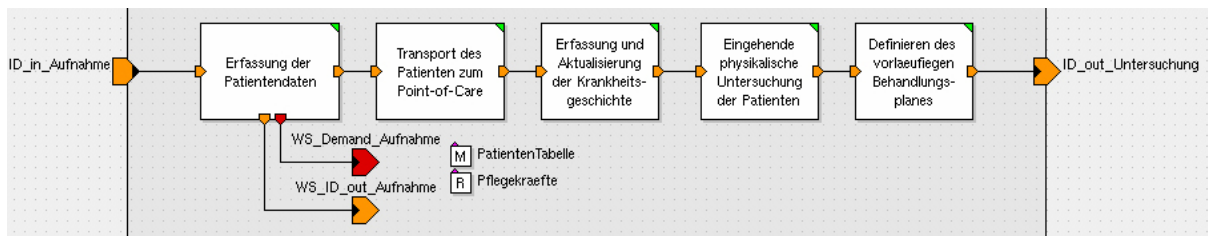


Abbildung 133: Ambulante Patientenaufnahme als Subprozess der Patientenaufnahme

Abbildung 134 zeigt die Struktur des Warteraums. Dieser wurde individuell für die vorliegende Aufgabenstellung entwickelt [Kühn06 S.53]. Seine grundlegende Funktionalität kann aber auch auf andere Modelle übertragen werden.

Um die Wartezeiten zwischen den einzelnen Behandlungsschritten zu messen, unterteilt sich der Warteraum im Modell - wie bereits weiter oben beschrieben - in die Bereiche „Warten 1“ (Wartezeit bis zur Blutentnahme), „Warten 2“ (Wartezeit nach Ende der Blutentnahme bis zum Beginn der ärztlichen Untersuchung) und

<sup>209</sup> Siehe auch Kapitel 6.1.

„Warten 3“ (Wartezeit nach Ende der Untersuchungen bis zum Beginn der Therapie).

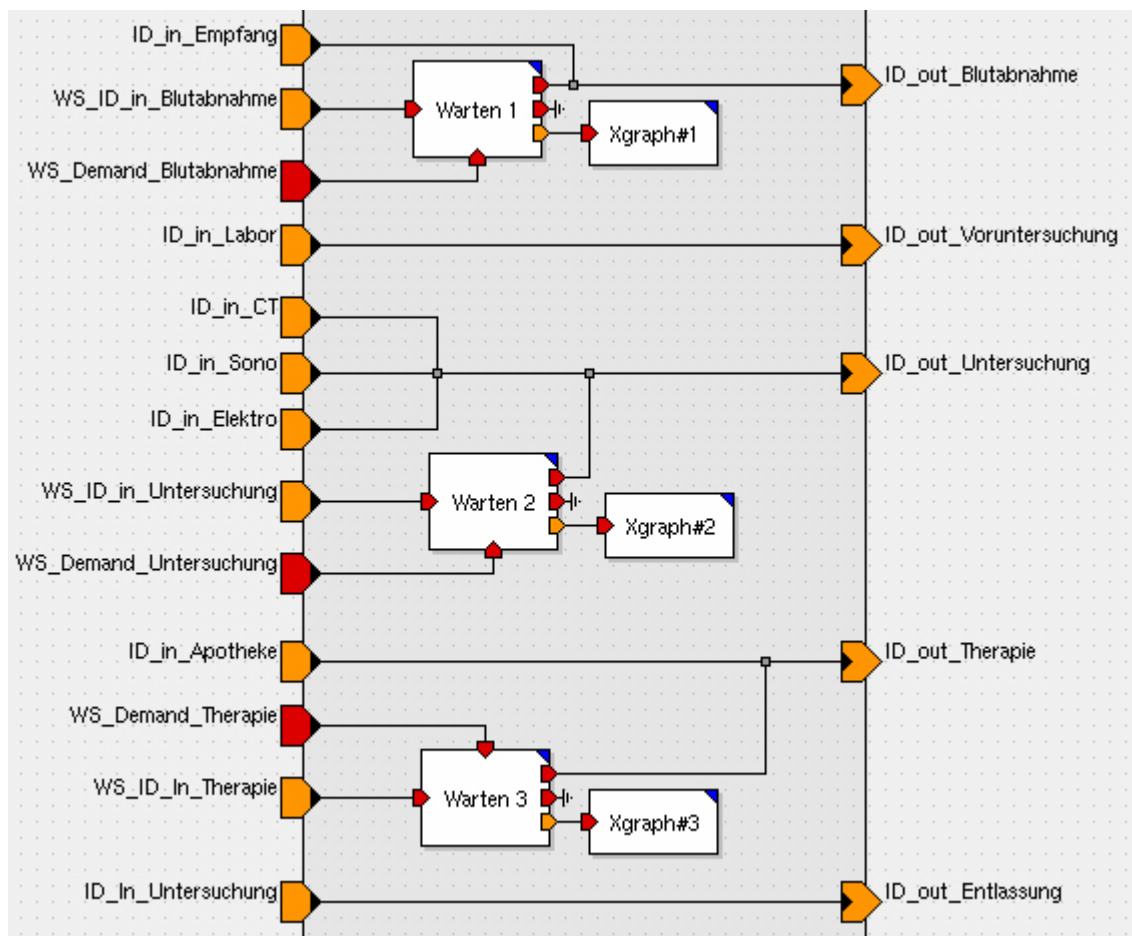


Abbildung 134: Struktur des Warteraums [Kühn06 S.53]

Die Primitive „Xgraph#1“ bis „Xgraph#3“ dienen zur grafischen Auswertung der Warteschlangenbelegung.

Abbildung 135 zeigt die Patientenuntersuchung. Teil dieses Funktionsblocks sind die Blutabnahme, das Veranlassen der Voruntersuchungen (CT, Sonographie und EKG), die ärztliche Untersuchung und das Abrufen der Medikation.

Patienten kommen zuerst zur Blutabnahme. Anschließend werden - wenn notwendig - weitere Untersuchungen wie z.B. eine CT veranlasst.<sup>210</sup> Ist dies nicht notwendig, geht der betreffende Patient direkt zur ärztlichen Untersuchung. Beschließt der Arzt, eine Therapie zu verabreichen, wird diese in der Apotheke

<sup>210</sup> Modul „Voruntersuchungen veranlassen“.

abgerufen. Dies stellt das Modul „Medikation abrufen“ dar. Die Verbindung zur Modellkomponente „Apotheke“ wird über den Ausgangsport „ID\_out\_Apotheke“ hergestellt.

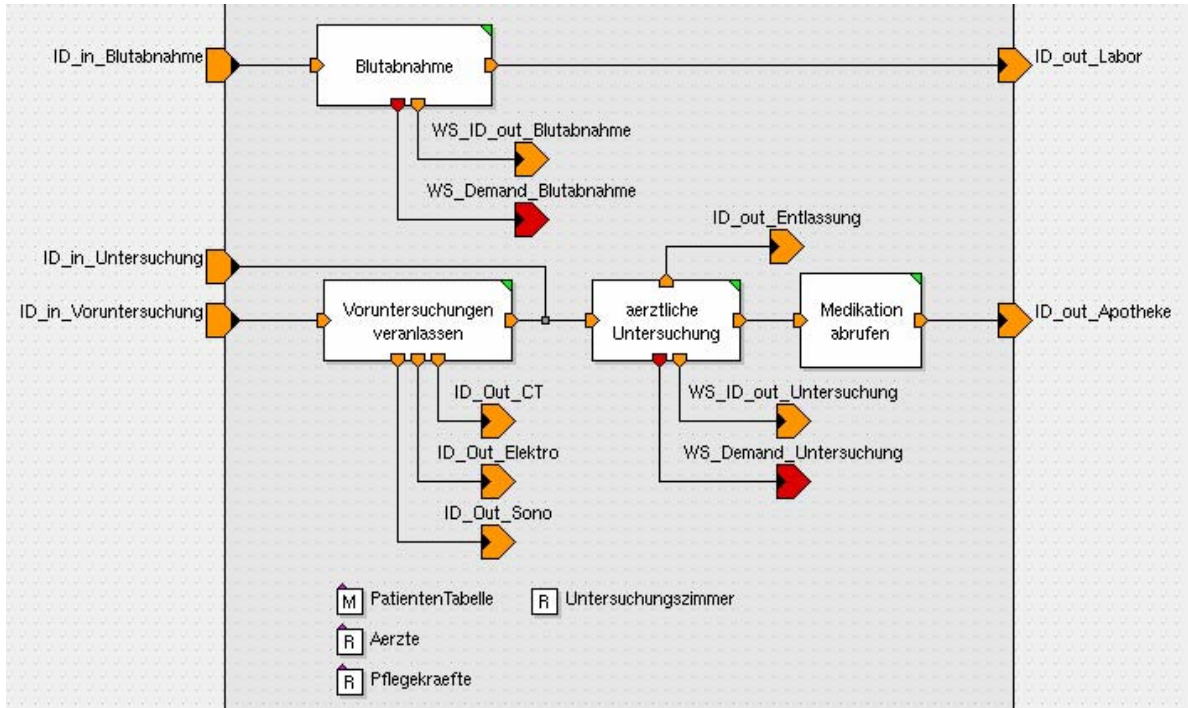


Abbildung 135: Aufbau der Modellkomponente „Untersuchung“

Der Aufbau der Modellkomponente „Apotheke“ ist in Abbildung 136 dargestellt.

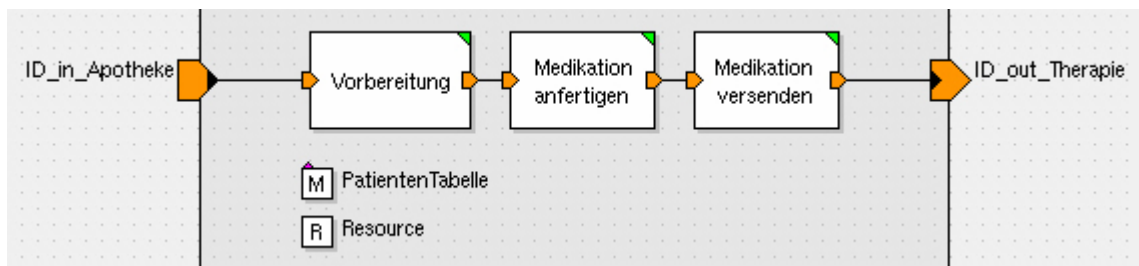


Abbildung 136: Aufbau der Modellkomponente „Apotheke“

Die eintreffenden Medikamentenbestellungen werden in der Apotheke erfasst und überprüft (Modul „Vorbereitung“). Anschließend erfolgen die Herstellung und das Versenden der Medikation an die Tagesklinik mit internem Kurier. Letzteres ist in Abbildung 137 dargestellt.



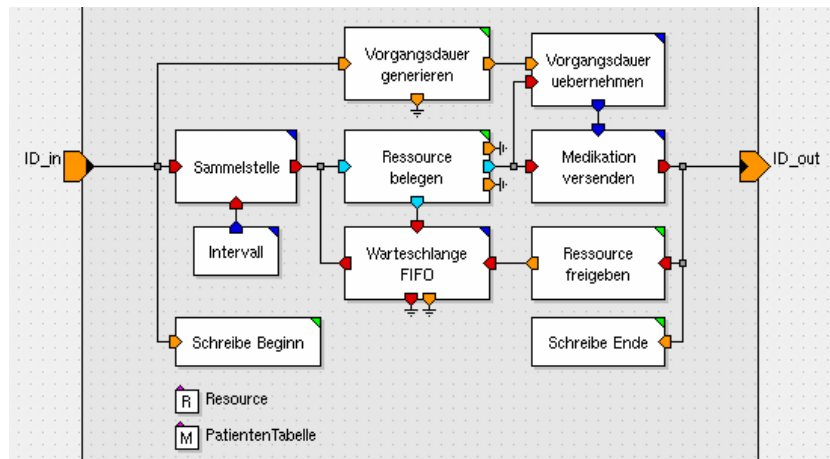


Abbildung 137: Aufbau des Moduls „Medikation versenden“

Das Prinzip der Sammelstelle funktioniert analog zur Sammelstelle, die in der Modellkomponente „Bluttransport“<sup>211</sup> eingesetzt wird: Erst nach einer gewissen Menge an zum Transport bereitstehenden Medikamenten bzw. nach einem bestimmten Warteintervall (Primitive „Intervall“) werden die Medikamente zur Tagesklinik transportiert.

Die beiden Modellkomponenten „Therapie“ und „Entlassung“ wurden bereits in Kapitel 6.3 bzw. 6.4 erläutert. Auf die entsprechenden Kapitel wird verwiesen.

Abbildung 138 zeigt den Gesamtüberblick der Bibliotheken und Module, die zur Modellentwicklung verwendet wurden.

<sup>211</sup> Siehe auch Kapitel 6.2.

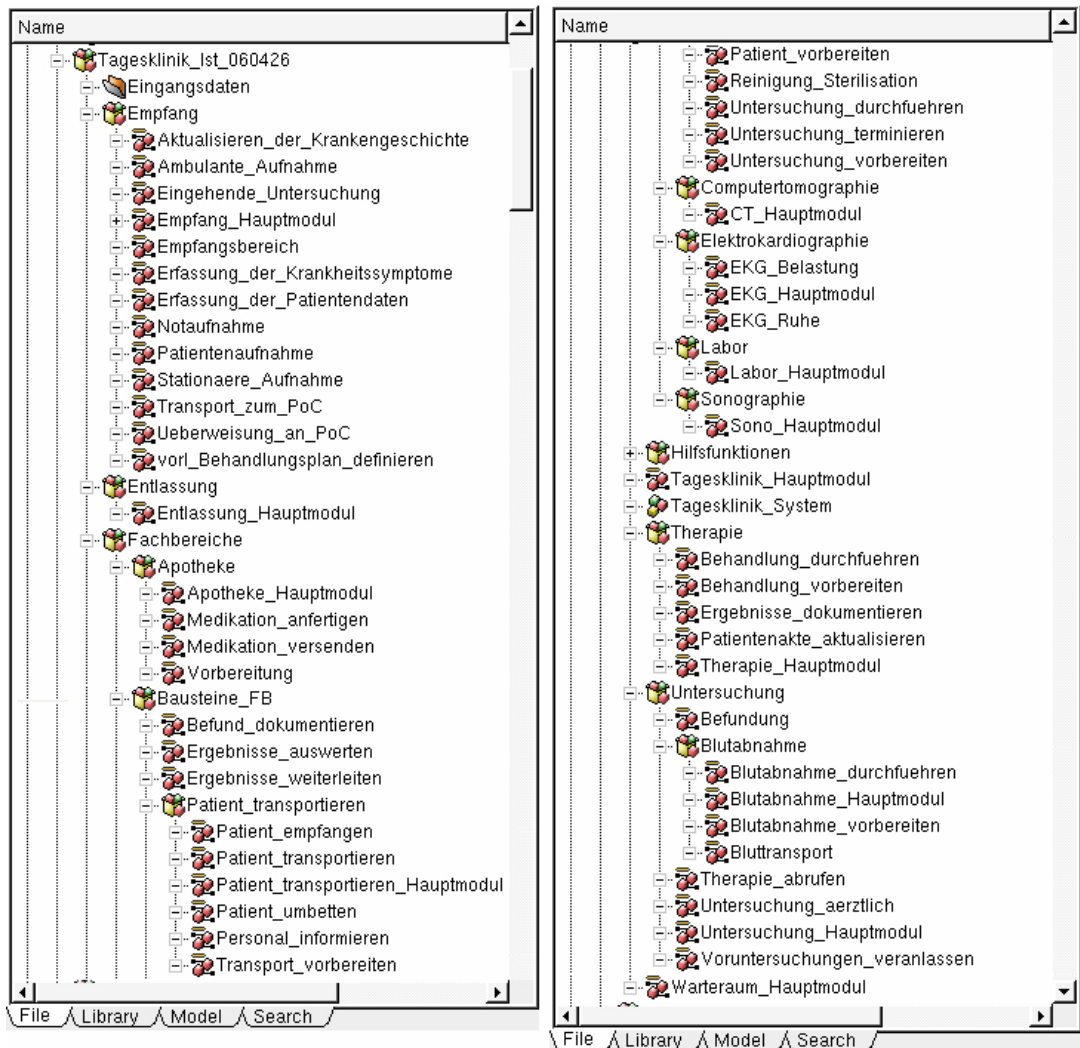


Abbildung 138: Baumstruktur des Modells der Tagesklinik im MLDesigner

Die für das Modell notwendigen Transaktionsdaten werden von einer Text-Datei eingelesen (vgl. Abb. 139). Diese Datei enthält u.a. die Patienten-ID<sup>212</sup>, die Patientenankunftszeit<sup>213</sup> sowie die Kennzeichnung der Patienten als Mitbehandler bzw. Therapiepatient<sup>214</sup>. Ferner werden während der Simulation bestimmte Zeiten protokolliert und anschließend in die Text-Datei geschrieben. Diese Zeiten dienen der späteren Analyse.

<sup>212</sup> Feld „ID“.

<sup>213</sup> Feld „Ankunft“.

<sup>214</sup> Feld „Mitbehandlung“.

Nr.	Bezeichnung	Nr.	Bezeichnung
0	ID	16	ElektroDauer
1	Ankunft	17	SonoStart
2	AufnahmeStart	18	SonoDauer
4	AufnahmeDauer	19	UntersuchungStart
5	AufnahmeEnde	20	UntersuchungDauer
6	Mitbehandlung	21	UntersuchungEnde
7	BlutabnahmeStart	22	BlutwertCheck
8	BlutabnahmeDauer	23	ZytostatikaAbruf
9	BlutabnahmeEnde	24	ZytostatikaDauer
10	LaboruntersuchungStart	25	ZytostatikaEintreffen
11	LaboruntersuchungDauer	26	TherapieStart
12	LaboruntersuchungEnde	27	TherapieDauer
13	CTStart	28	TherapieEnde
14	CTDauer	29	Entlassung
15	ElektroStart	30	Hilfswerte

Abbildung 139: Datenstruktur der Text-Datei und Vektoren

Bei der Modellierung der Tagesklinik wurde der Materialverbrauch nicht berücksichtigt, da dieser im vorliegenden Fall keinen Engpass im Sinne einer „Nicht-Verfügbarkeit“ darstellt.

Für die Ärzte und Pflegekräfte wurde aus Gründen der Reduktion der Modellkomplexität jeweils ein identischer Ausbildungsstand<sup>215</sup> angenommen und eine Ressource „Aerzte“ bzw. „Pflegekraefte“ angelegt. Über den Property Editor des MLDesigners wurde die initiale Anzahl<sup>216</sup> der jeweiligen Ressource festgelegt. Abbildung 140 zeigt den Property Editor am Beispiel der Ressource „Aerzte“.

<sup>215</sup> Siehe hierzu auch Kapitel 4.1.4.1.

<sup>216</sup> Engl.: Initial Capacity.

Name	Value
<b>Aerzte</b>	
Name	Aerzte
Type	Quantity
Scope	Internal
Description	
Number of Dimensions	1
Initial Capacity	3
Maximum Queue Occupancy	0

Module Properties | Resource Properties

Abbildung 140: Property Editor am Beispiel der Ressource „Aerzte“

Wie in Kapitel 6 beschrieben, werden die Vorgangsdauern der einzelnen Komponenten mittels Poissonverteilung errechnet. Die hierfür benötigten Werte wurden für jede Modellkomponente im Property Editor eingetragen. Abbildung 141 zeigt dies am Beispiel des Prozessschrittes „Empfang“.

Name	Value
<i>Name</i>	<i>Empfang#1</i>
Label	Empfang
<b>P</b> minDauer	3
<b>P</b> maxDauer	6
<b>P</b> Schnitt	5

Instance Properties

Abbildung 141: Parametrisierung des Prozessschrittes „Empfang“

Abbildung 142 zeigt die Gesamtübersicht der eingestellten Durchlaufzeiten für die einzelnen Modellkomponenten.

<b>Beschreibung</b>	<b>Minimum</b>	<b>Maximum</b>	<b>Durchschnitt</b>
Empfang	3 min	6 min	5 min
Blutabnahme	4 min	32 min	12 min
Labor	10 min	15 min	12 min
CT	30 min	60 min	50 min
Sonographie	15 min	30 min	25 min
EKG	10 min	30 min	20 min
Untersuchung	3 min	60 min	18 min
Apotheke	20 min	60 min	50 min
Therapie	5 min	305 min	145 min
Entlassung	5 min	225 min	34 min

Abbildung 142: Übersicht der Durchlaufzeiten der einzelnen Modellkomponenten

Der Einsatz der wieder verwendbaren Modellkomponenten, wie sie in Kapitel 6 entwickelt und beschrieben wurden, hat die Entwicklungszeit des vorliegenden Modells signifikant beschleunigt. Durch die komfortable Möglichkeit, Modellbausteine im MLDesigner mittels des „Property Editors“ zu parametrisieren, konnte in nur kurzer Zeit ein lauffähiges Modell der Tagesklinik erstellt werden.

### 7.3 Modellbestimmung und Validierung des Modells

Zur Modellbestimmung sowie zur Überprüfung der korrekten Funktionsweise des Modells wurden Ist-Daten an insgesamt 55 Arbeitstagen mittels eines strukturierten Erfassungsbogens<sup>217</sup> in der Tagesklinik erhoben. Die erfassten Daten wurden anschließend gemeinsam mit den Ärzten und dem Pflegepersonal auf Vollständigkeit überprüft und plausibilisiert. Die daraus errechneten minimalen, maximalen und durchschnittlichen Warte- und Durchlaufzeiten je Prozessschritt dienten zum einen zur Modellbestimmung<sup>218</sup>, zum anderen zur Validierung<sup>219</sup> des Modells. Zur Modellbestimmung wurde ein Datenbestand von 15 Arbeitstagen herangezogen. Für die Validierung wurden die Daten der verbleibenden 40 Arbeitstage herangezogen.

Im Rahmen der Datenerhebung wurden für jeden Prozessschritt die jeweilige Durchlaufzeit und die vor den einzelnen Prozessschritten entstandenen Wartezeiten gemessen. Aus der Summe dieser Zeiten errechnet sich die Gesamtdurchlaufzeit des jeweiligen Patienten. Zudem wurde für jeden betrachteten Arbeitstag die entsprechende Verteilungsfunktion für die Liegenbelegung erstellt. Die minimalen, maximalen und durchschnittlichen Durchlaufzeiten wurden den jeweiligen Modellkomponenten als Werte zugewiesen. Abbildung 143 zeigt dies am Beispiel der Patientenaufnahme (Modul „Empfangsbereich“).

---

<sup>217</sup> Siehe Anhang 1, Tabelle 8.

<sup>218</sup> 1. Teilmenge des Datenbestands.

<sup>219</sup> 2. Teilmenge des Datenbestands.

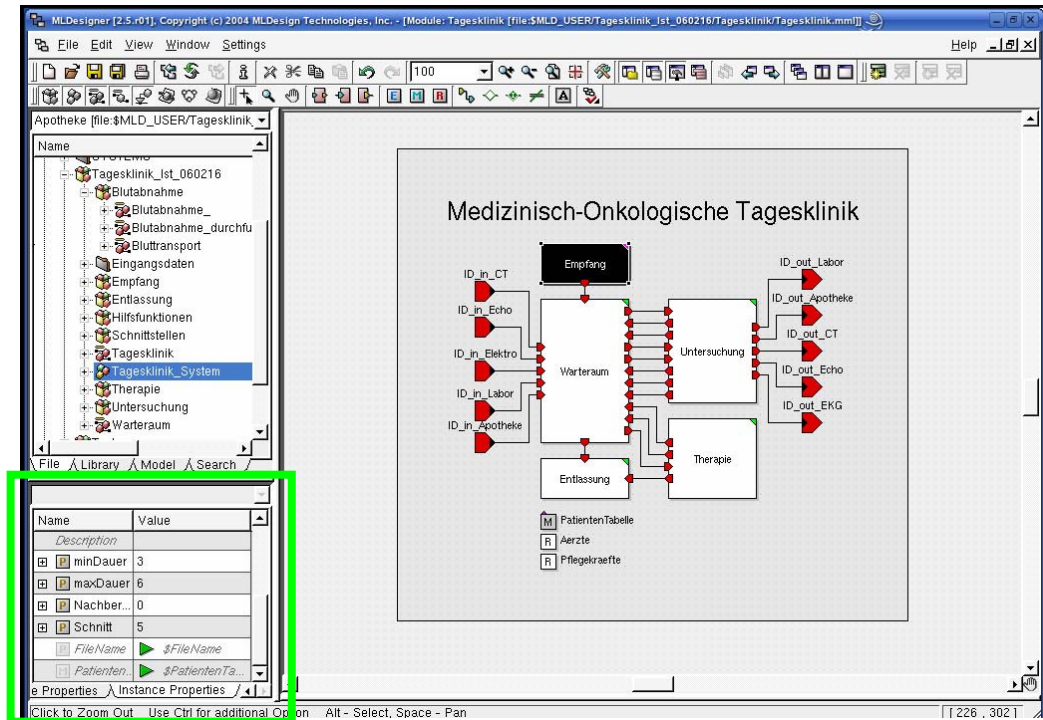


Abbildung 143: Minimale, maximal und durchschnittliche Dauer der Patientenaufnahme

Die Ist-Ankunftszeiten der Patienten<sup>220</sup> werden vom Modell aus einer Text-Datei eingelesen und lösen zum Zeitpunkt  $x_t =$  „Ist-Ankunftszeit des Patienten“ das Starterereignis der Transaktion aus. Als Ergebnis für jeden Simulationstag ergibt sich u.a. eine Verteilungsfunktion für die Liegenbelegung sowie eine Tabelle, welche die im Simulationslauf entstandenen Durchlauf- und Wartezeiten je Tag enthält. Die Ergebnisse der einzelnen Simulationsläufe wurden anschließend mit den jeweiligen Ergebnissen, die sich auf Basis der Ist-Datenerhebung ergeben haben, verglichen<sup>221</sup>. Das Modell wurde schrittweise verfeinert, bis sich das Laufzeitverhalten des Modells dem Verhalten des Realsystems ausreichend annäherte. Am Ende betrug die Abweichung der im Ist-Modell gemessenen mittleren Gesamtdurchlaufzeit gegenüber der mittleren Gesamtdurchlaufzeit im Realsystem -2,3%. Die Abweichung der im Ist-Modell gemessenen mittleren Gesamtwartezeit betrug gegenüber der mittleren Gesamtwartezeit im Realsystem -5,1%.

Zur Validierung wurden – wie oben bereits beschrieben - die verbleibenden 40 Arbeitstage herangezogen. Ein Datenvergleich, d.h. Vergleich der im „Ist“

<sup>220</sup> Ermittelt während der Datenerhebung in der Tagesklinik.

<sup>221</sup> Siehe Anhang 1, Tabelle 9.

ermittelten Daten mit den Modellergebnissen, wurde wie bereits bei der Modellbestimmung für die ermittelten maximalen, minimalen und durchschnittlichen Durchlauf- und Wartezeiten durchgeführt. Bei der Validierung beträgt die Abweichung der im Modell gemessenen mittleren Gesamtdurchlaufzeit gegenüber der mittleren Gesamtdurchlaufzeit im Realsystem -0,7%. Die Abweichung der im Modell gemessenen mittleren Gesamtwartezeit beträgt gegenüber der mittleren Gesamtwartezeit im Realsystem -4%.

Neben dem Vergleich der Ist-Daten mit den Daten aus den Simulationsläufen wurde auch das grundlegende Verhalten des Modells plausibilisiert. Hier lag der Fokus auf der Entwicklung der Wartezeiten bei steigender bzw. fallender Patientenzahl je Arbeitstag, differenziert nach dem Zeitpunkt der Einbestellung der Patienten. Es ist an dieser Stelle nicht überraschend, dass bei gleichzeitiger Einbestellung vieler Patienten die durchschnittliche Wartezeit im Modell ansteigt. Bei über den Arbeitstag verteilter Einbestellung der Patienten fällt die durchschnittliche Wartezeit im Vergleich insgesamt niedriger aus. Diese Tendenzen entsprechen auch dem Verhalten des Realsystems.

Weiterhin wurde das Verhalten des Modells bei Veränderung der Behandlungsdauer je Prozessschritt untersucht. Das Ergebnis entsprach auch hier dem gewünschten Modellverhalten: Je länger die durchschnittliche Behandlungsdauer der einzelnen Prozessschritte gewählt wurde, desto länger war folglich die durchschnittliche Gesamtdurchlaufzeit der Patienten.

#### **7.4 Lösungen zur Optimierung der Prozesse**

Zur Optimierung der in der Tagesklinik ablaufenden Prozesse kommen unterschiedliche Gebiete der Optimierung in Betracht. Hierzu gehören die Optimierung des Patientenflusses inkl. der Einbestellungsreihenfolge, des Materialflusses, des Informations- bzw. Datenflusses, die Optimierung der Abfolge (=Reihenfolge) der einzelnen Behandlungsschritte sowie die Optimierung der Verfügbarkeit und Allokation benötigter Ressourcen (vgl. Abb. 144).



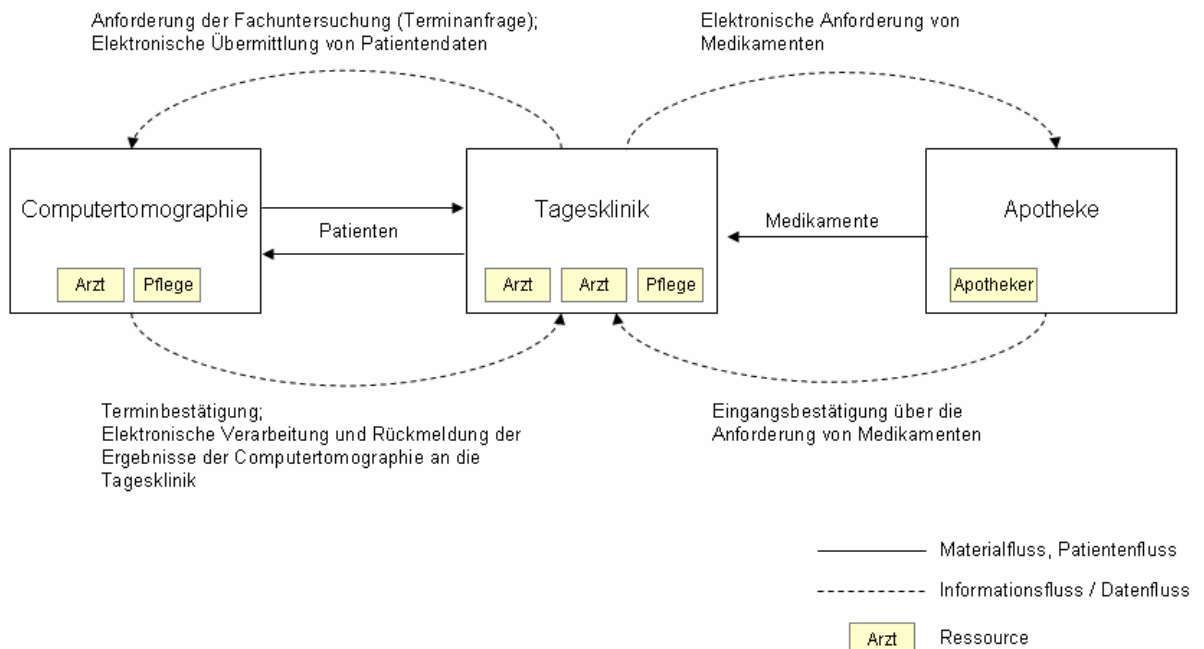


Abbildung 144: Prinzipdarstellung von Patienten-, Material- und Informationsflüssen sowie Personalressourcen

Der Patientenfluss umfasst die Bewegung der Patienten innerhalb der Tagesklinik sowie zwischen Tagesklinik und Fachuntersuchungen. Zum Materialfluss gehört z.B. das Versenden der Medikamente von der Apotheke zur Tagesklinik und die Weitergabe der entnommenen Blutproben von der Tagesklinik an das Labor. Der Informationsfluss bezieht sich im Fall der betrachteten Tagesklinik u.a. auf die Weitergabe von Patientendaten (→ Patientenakte) an die behandelnden Ärzte der einzelnen Fachbereiche sowie das Anfordern von Medikamenten in der Apotheke mittels eines Bestellformulars. Letzteres kann elektronisch übermittelt oder in Papierform weitergegeben werden. Die Anforderung eines Medikamentes löst - sofern das Medikament verfügbar ist - den entsprechenden Materialfluss von der Apotheke zur Tagesklinik aus. Zur Durchführung der Behandlung sind u.a. Ärzte und Pflegefachkräfte (= Ressourcen) notwendig. Nur wenn diese in ausreichender Kapazität und mit entsprechender Qualifikation zur Verfügung stehen, können die betreffenden Behandlungsschritte - ohne Zeitverzug - durchgeführt werden.

In den folgenden Kapiteln erstreckt sich die Analyse und Optimierung der Prozessabläufe in der betrachteten Tagesklinik auf den Patientenfluss, auf die Planung der Einbestellungsreihenfolge der Patienten, auf den Materialfluss sowie auf die Ressourcenverfügbarkeit. Informations- und Datenflüsse werden aus zwei Gründen im folgenden Verlauf der Arbeit ausgegrenzt: Einerseits wird dadurch ein weiterer Anstieg der Modellkomplexität vermieden. Andererseits ergaben die in der Tagesklinik durchgeführten Analysen der Ist-Prozesse keinerlei Hinweise darauf, dass benötigte Informationen bzw. Daten nicht bzw. vermehrt nicht rechtzeitig zur Verfügung stehen. In Bezug auf die vorliegende Aufgabenstellung wird aus den oben genannten Gründen deshalb kein entscheidendes Optimierungspotential gesehen, welches den damit verbundenen Anstieg der Modellkomplexität rechtfertigen würde.

#### **7.4.1 Performance-Indikatoren zur Analyse von Schwachstellen**

Performance-Indikatoren werden im Kontext der vorliegenden Arbeit als steuerungsrelevante Kennzahlen zur Analyse von Unternehmensprozessen definiert. Um die für den vorliegenden Modellzweck sinnvollen Performance-Indikatoren herzuleiten, wird an dieser Stelle nochmals auf die Ausgangssituation<sup>222</sup> der Tagesklinik Bezug genommen. Es ist auffallend, dass die Patienten überwiegend zwischen 7:30 Uhr und 8:30 Uhr in die Klinik einbestellt werden, die ersten Therapien jedoch i.d.R. nicht vor 10:00 Uhr beginnen.

Die Durchlaufzeit der Patienten spielt als Performance-Indikator eine entscheidende Rolle. Sie setzt sich aus den jeweiligen Wartezeiten vor jedem Behandlungsschritt und den Behandlungszeiten je Behandlungsschritt zusammen. Obwohl Behandlungszeiten grundsätzlich auch Gegenstand einer Optimierung sein können, werden diese in der vorliegenden Simulationsstudie nicht als mögliches Optimierungspotential berücksichtigt. Dies hat mehrere Gründe: Einerseits wäre fundiertes medizinisches Fachwissen notwendig, um

---

<sup>222</sup> Siehe hierzu auch Kapitel 7.1.

Behandlungszeiten zu analysieren und zu optimieren, andererseits sind die einzelnen Behandlungsdauern von verschiedensten, oftmals nicht beeinflussbaren Faktoren abhängig. Dies gilt z.B. für die Computertomographie und das EKG. Hier ist der Zeitaufwand weitgehend durch die Untersuchungsmethode selbst und nur bedingt durch die Erfahrung des Personals determiniert. Es ist ferner zu erwarten, dass die Behandlungsdauer nicht die Hauptursache für die insgesamt langen Durchlaufzeiten der Patienten ist, sondern dies durch die Wartezeiten bedingt wird. Letztere betragen immerhin bis zu 45% der Gesamtdurchlaufzeit. Wartezeiten sind nichts anderes als „unproduktive“ Zeiten und sollten daher in jedem Fall minimiert werden.

Wartezeiten werden durch den Grad der Verfügbarkeit notwendiger Ressourcen sowie durch den Grad der Prozessintegration und –qualität beeinflusst. Sind Ressourcen nicht zur „richtigen Zeit am richtigen Ort“, kommt es zu Wartezeiten bei den Patienten. Fehlt es an ausreichender Kommunikation und Abstimmung der an der Behandlung beteiligten klinischen Bereiche, könnten auch trotz ausreichender Ressourcenverfügbarkeit unnötige Wartezeiten für Patienten auftreten. Dies ist zum Beispiel dann der Fall, wenn eine Fachuntersuchung nicht rechtzeitig angemeldet wird oder der Fachabteilung die Patientenakte nicht vorliegt.

Bei der Analyse des Realsystems stellte sich heraus, dass sowohl Material als auch medizinische Geräte in ausreichender Qualität und Quantität vorhanden und rechtzeitig verfügbar sind. Da deshalb weder Material noch medizinische Geräte im vorliegenden Fall einen Engpass darstellen, werden diese beiden Ressourcenarten bei der Optimierung nicht berücksichtigt. Ebenfalls aus und vor bleiben die räumlichen Kapazitäten, da diese in der betrachteten Tagesklinik zum einen fest vorgegeben sind, zum anderen ergab die in der Tagesklinik durchgeführte Analyse keine Hinweise auf einen Engpass bei den Räumlichkeiten. Jedem Arzt steht sein eigenes Untersuchungszimmer zur Verfügung. Ein Raum für die Blutabnahme mit ausreichender Kapazität ist in der Tagesklinik ebenfalls vorhanden. Zudem erfolgt die Blutabnahme teils auch direkt auf den Patientenplätzen, so dass auch hier kein Engpass entstehen kann, da insgesamt mehr Blutentnahmeplätze als Personal vorhanden sind.

Engpässe - und die damit entstehenden Wartezeiten bei Patienten - könnten jedoch durch mangelnde Verfügbarkeit des Personals entstehen. Deshalb wird vom entwickelten Simulationsmodell eine Belegungsstatistik für die Ärzte und das Pflegepersonal erzeugt und ausgewertet – wie weiter oben in Abbildung 131 exemplarisch gezeigt.

Ein weiterer Performance-Indikator ist die Auslastung der Therapieplätze (=Patientenliegen). Unnötige Wartezeiten für Patienten könnten dann entstehen, wenn die Liegenkapazität zu bestimmten Zeitpunkten ausgeschöpft ist, sich jedoch noch Patienten in der Tagesklinik befinden und auf ihre Therapie warten.

Aus den bisherigen Erkenntnissen ergeben sich daher für die vorliegende Aufgabenstellung folgende relevante Performance-Indikatoren:

- (1) Wartezeit zwischen dem Ende der Patientenaufnahme und dem Beginn der Blutabnahme
- (2) Wartezeit zwischen dem Ende der Blutabnahme und dem Beginn der Untersuchung
- (3) Wartezeit zwischen dem Ende der Untersuchung und dem Beginn der Therapie
- (4) Wartezeit zwischen dem Ende der Therapie und der Entlassung
- (5) Gesamtdurchlaufzeit der Patienten
- (6) Auslastung der Ärzte und des Pflegepersonals
- (7) Auslastung der Patientenliegen

Bei der späteren Optimierung rücken die obigen Kennzahlen ins Zentrum der Betrachtung.

### 7.4.2 Eliminierung redundanter Arbeitsschritte

Als ersten Ansatz zur Prozessoptimierung wurde die Möglichkeit der Eliminierung redundanter Arbeitsschritte untersucht. Eine Eliminierung redundanter Arbeitsschritte wirkt sich positiv auf die Durchlaufzeiten der Patienten aus, da sich einerseits beim Wegfall von Arbeitsschritten die Durchlaufzeiten verkürzen, andererseits schafft der Wegfall von Arbeitsschritten auch freie Kapazitäten beim Personal.

Im Rahmen der durchgeführten Analysen wurde festgestellt, dass Patienten, die zur Computertomographie müssen, sowohl in der radiologischen Fachabteilung als auch in der Tagesklinik „gestochen“ werden (Ist-Prozess). Bei der Computertomographie ist dies notwendig, um Kontrastmittel zu spritzen, in der Tagesklinik, um Blut abzunehmen und gegebenenfalls die Chemotherapie anzuhängen.

In dem anschließend optimierten Soll-Prozess bekommt diese Patientengruppe nunmehr direkt und ausschließlich in der Radiologie eine Kanüle gelegt.<sup>223</sup> Die gelegten Kanülen können sowohl für das Spritzen des Kontrastmittels als auch zur Blutabnahme in der Tagesklinik genutzt werden. Im Anschluss an die Computertomographie kommt diese Gruppe von Patienten zur weiteren Untersuchung und Blutabnahme in die Tagesklinik. Dieses Vorgehen entlastet aufgrund des Wegfalls eines Arbeitsschrittes die Pflegefachkräfte der Tagesklinik bei den anfallenden Blutentnahmen und führt letztendlich zur Verringerung der Patientendurchlaufzeiten. Zudem steigt die Qualität der Behandlung insofern an, als dass Patienten nicht mehr zweifach zum Legen der Kanüle gestochen werden, sondern nur noch einmal.

Abbildung 145 zeigt das Layout des geänderten Soll-Modells für das eben beschriebene Vorgehen. Der Output-Port vom „Empfang“ zum „CT“ („ID\_out\_CT“) repräsentiert den optimierten Prozessablauf.

---

<sup>223</sup> Man spricht hier auch vom Legen eines venösen Zugangs.

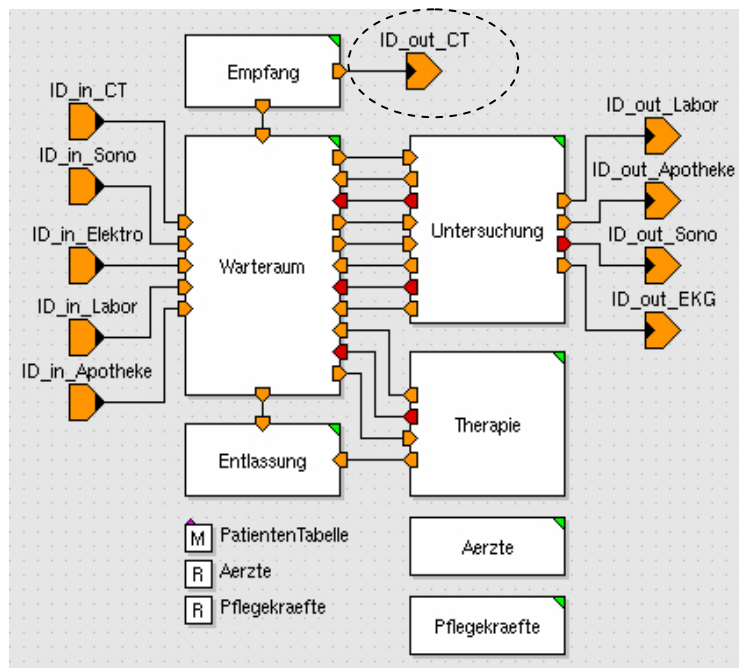


Abbildung 145: Soll-Modell der Tagesklinik [Kühn06 S.78]

Die erwartete Zeitersparnis lässt sich auf einfache Art und Weise wie folgt berechnen: Durchschnittlich 15% der Patienten bekommen eine Computertomographie. Dies entspricht täglich ca. 5 Patienten. Die durchschnittliche Zeitdauer, um einen venösen Zugang zu legen, wurde mit 10 Minuten ermittelt. Somit ergibt sich für das Personal der Tagesklinik eine durchschnittliche Zeitersparnis von insgesamt ca. 50 Minuten täglich.

Dieses Ergebnis stellte sich auch bei den späteren Simulationsdurchläufen mit dem Soll-Modell ein.

### 7.4.3 Überprüfen der Möglichkeit zur Wegstrecken-Optimierung

Ein weiterer Ansatz zur Reduktion der Durchlaufzeiten von Patienten könnte in der Optimierung der von den Patienten zurückzulegenden Wegstrecken innerhalb der Klinik liegen. Um dies näher zu untersuchen, wurde die nachstehende Distanzmatrix ermittelt (vgl. Abb. 146). Diese zeigt die Kombination aller

theoretisch möglichen Wegstrecken, die Patienten der Tagesklinik gegebenenfalls zurücklegen können.

	Empfang	Warteraum	Blutentnahme	Untersuchung	Therapiebereich	Sonographie	Computertomographie	EKG
Empfang	0	5	5	5	7	14	28	17
Warteraum	5	0	3	3	12	19	33	22
Blutentnahme	5	3	0	4	12	19	33	22
Untersuchung	5	3	4	0	12	19	33	22
Therapiebereich	7	12	12	12	0	21	25	14
Sonographie	14	19	19	19	21	0	30	8
Computertomographie	28	33	33	33	25	30	0	24
EKG	17	22	22	22	14	8	24	0

Abbildung 146: Distanzmatrix<sup>224</sup>

Die ermittelten Distanzen innerhalb der Tagesklinik, von der Tagesklinik zu den Fachuntersuchungen, sowie die Wegstrecken zwischen den einzelnen Fachuntersuchungen sind in Bezug auf die vorliegende Aufgabenstellung<sup>225</sup> eher als zu gering einzustufen, als dass sich hier ein nennenswertes Optimierungspotential ergibt. Es könnte sich jedoch trotz der geringen Einzeldistanzen ein Ansatzpunkt zur Optimierung ergeben, wenn ein hoher Anteil an Patienten mehrere Fachuntersuchungen hintereinander durchlaufen muss und sich damit die einzelnen Wegstrecken kumulieren. Hier ergab die durchgeführte Analyse, dass durchschnittlich etwa 15% der Patienten zur Computertomographie, 20% zur Sonographie und ca. 3% zum EKG müssen. Ca. 12% aller Patienten müssen an einem Behandlungstag zu mehr als einer Fachuntersuchung. Bei durchschnittlich 34 Patienten täglich ergibt sich für die vorliegende Aufgabenstellung damit dennoch kein nennenswertes Optimierungspotential.

<sup>224</sup> Angaben in Meter.

<sup>225</sup> Siehe hierzu auch Kapitel 7.1.

Ein weiterer Grund, weshalb sich in Bezug auf die Wegstrecken innerhalb der Tagesklinik im Sinne der vorgegebenen Modell-Zielsetzungen<sup>226</sup> kein ausreichendes Optimierungspotential ergibt, liegt darin, dass die Reihenfolge der Wegstrecken, welche die Patienten innerhalb der Klinik zurücklegen müssen, durch die Behandlungsabläufe weitgehend vorbestimmt ist. Um überhaupt Optimierungen für die Tagesklinik vornehmen zu können, müssten (gravierende) bauliche Veränderungen durchgeführt werden, um z.B. einzelne Fachabteilungen näher an die Tagesklinik zu setzen. Dieser Aufwand – der zudem wiederum Einfluss auf andere Fachbereiche der Klinik hätte – ist gemessen am zu erwartenden Nutzen in keinsten Weise zu rechtfertigen.

Im Rahmen der Modellabstraktion wurde aus den oben genannten Gründen deshalb auf die Berücksichtigung der Wegstrecken und deren Optimierung im Soll-Modell verzichtet.

#### **7.4.4 Reduktion von Wartezeiten**

Weitere Ansätze zur Prozessoptimierung lassen sich in Bezug auf die Wartezeiten vermuten, da der Anteil der Wartezeit an der gesamten Durchlaufzeit der Patienten mit durchschnittlich 45% auffallend hoch ist. Mit anderen Worten verbringen die Patienten in der ambulanten Tagesklinik knapp die Hälfte ihrer Zeit mit Warten – also „unproduktiv“. Die Wartezeit ist daher – wie bereits in Kapitel 7.4.1 dargelegt – einer der Hauptansatzpunkte im Rahmen der vorliegenden Prozessoptimierung. Es gilt nun zu analysieren, wodurch die signifikanten Wartezeiten entstehen und wie die Ursachen hierfür beseitigt werden können.

---

<sup>226</sup> Vgl. Kapitel 7.1.



### 7.4.4.1 Analyse der Verfügbarkeit und Auslastung eingesetzter Ressourcen

Wartezeiten der Patienten könnten aus der unzureichenden Verfügbarkeit von Ärzten, Pflegepersonal und Behandlungsplätzen resultieren. Deshalb wurde die Auslastung dieser Ressourcen analysiert. Die Abbildungen 147 bis 149 zeigen repräsentative Beispiele für die im Ist-Modell ermittelten Belegungsstatistiken der o.g. Ressourcen.

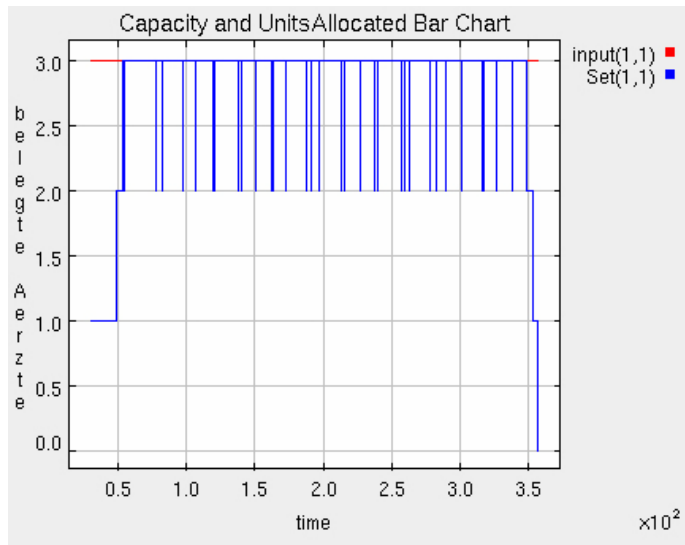


Abbildung 147: Belegungsstatistik der Ärzte

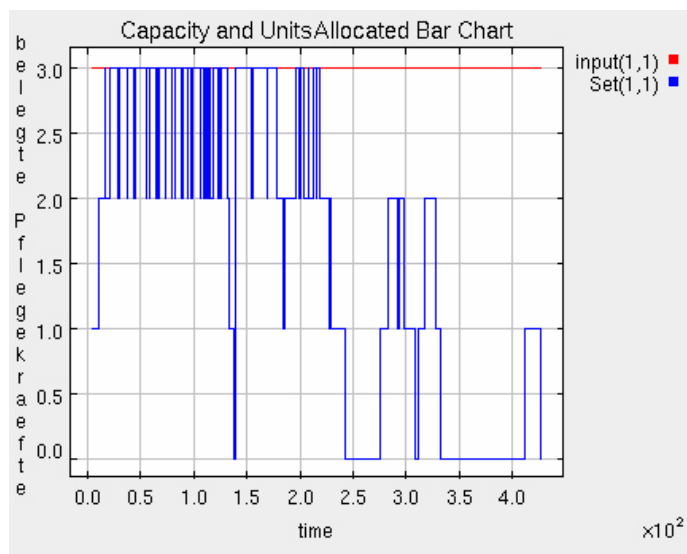


Abbildung 148: Belegungsstatistik der Pflegekräfte

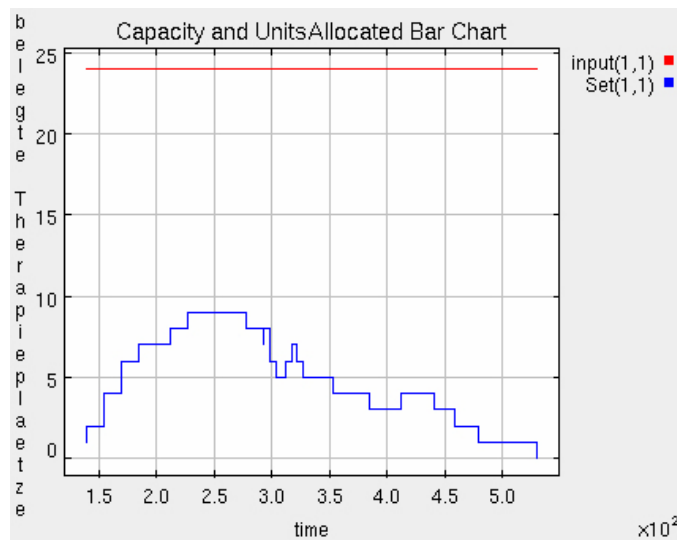


Abbildung 149: Belegungsstatistik der Behandlungsplätze (Therapieplätze)

Während das Pflegepersonal in den ersten Stunden gut ausgelastet ist, zeigen sich am Nachmittag freie Kapazitäten. Ärzte sind etwa während der ersten Stunde nach Öffnung der Tagesklinik sowie auch in der letzten Stunde der Öffnungszeit nur teilweise ausgelastet. Die ermittelten Belegungsstatistiken geben den klaren Hinweis, dass in der Entzerrung der Arbeitsbelastung - insbesondere beim Pflegepersonal - das Verbesserungspotential zu suchen ist. Dies kann durch einen ausgewogenen „Patientenmix“<sup>227</sup> sowie durch eine optimierte Terminvergabe (= optimierte Einbestellungsreihenfolge der Patienten) erreicht werden. Auf diesen Punkt wird in Kapitel 7.4.4.2 dediziert eingegangen. Eine bloße Aufstockung des Personals könnte zwar theoretisch auch zu einer Verkürzung der Patientenwartezeiten führen, dies wäre aber weder aus wirtschaftlichen Aspekten sinnvoll, da dies zu höheren Kosten für die Klinik führen würde, noch würde dies per se zu einem optimierten Prozessablauf beitragen. Ferner ist aufgrund der fest vorgegebenen Anzahl an Arztzimmern<sup>228</sup> sowie der vorgegebenen räumlichen Beschränkung in Bezug auf das Blutentnahmezimmer eine Aufstockung der Ärzte und des Pflegepersonals im vorliegenden Fall keine sinnvolle Option.

<sup>227</sup> Unter Patientenmix wird hier die Anzahl der unterschiedlichen Krankheitsbilder und Schwere der Erkrankung je Patient und Behandlungstag verstanden.

<sup>228</sup> Je einem Arzt steht je ein Arztzimmer/Behandlungsraum zur Verfügung.

Die Belegungsstatistik der Behandlungsplätze zeigt, dass gemessen an der bisherigen Anzahl durchschnittlich behandelter Patienten die 24 Patientenliegen ausreichende Kapazität zur Verfügung stellen (vgl. Abb. 149). Die durchgeführten Simulationsläufe ergaben, dass von den 24 Plätzen nur teilweise mehr als die Hälfte gleichzeitig belegt sind. Auch hier ist im Rahmen der vorliegenden Modellzielsetzung somit kein Optimierungspotential in Bezug auf die Anzahl der Patientenliegen vorhanden.

#### **7.4.4.2 Optimierung der Einbestellungsreihenfolge der Patienten mittels eines genetischen Algorithmus**

Die Analyse der Einbestellungspraxis von Patienten im Ist-Prozess ergab, dass hierfür keine definierten und standardisierten Prozesse existierten. Es könnte daher sein, dass diese „chaotische“ Vergabe von Patiententerminen (→ ineffiziente Reihenfolgeplanung) zu unnötigen Warte- und damit auch Durchlaufzeiten führt. Um diese Vermutung zu belegen und konkrete Aussagen zu einer optimierten Einbestellungsreihenfolge zu erhalten, wurde im Folgenden ein genetischer Algorithmus entwickelt und in das Simulationsmodell implementiert. Dieser soll Aufschluss darüber geben, ob es bei der Reihenfolgeplanung von Patienten Optimierungspotential gibt und wie dieses konkret ausgeschöpft werden kann.

##### **7.4.4.2.1 Kurzdarstellung von genetischen Algorithmen**

Genetische Algorithmen sind heuristische Optimierungsverfahren und gehören zur Gruppe der evolutionären Algorithmen [Stew87 S.247-251; Niss97 S.1]. Der Vorteil dieses naturanalogen Verfahrens liegt darin, dass es - im Gegensatz zu anderen Optimierungsverfahren wie z.B. Greedy-Strategien - die Optimierung nicht bei einem lokalen Optimum abbricht [Gold89 S.2-6; SiDe08 S.11-12 u. S.24-27; SHF94 S.102-103; Kinn94 S.43-45 u. S.65]. Dies wäre im Sinne des

Findens einer (möglichst) optimalen Lösung problematisch, da lokale Optima i.d.R. hinter dem globalen Optimum liegen, also deutlich schlechter sein können, als das globale Optimum.

Bei genetischen Algorithmen werden die evolutionären Prozesse auf Ebene der Genetik eingesetzt. Die dahinterliegende Philosophie: Aus besserem Erbgut / Erbinformationen entstehen bessere Nachfahren (= bessere Lösungen) [Wag04 S.9-26; Niss97 S.2-8; Gold89 S.1-2].

Genetische Algorithmen haben folgenden Aufbau bzw. Ablauf: Nachdem in einem ersten Schritt per Zufallsauswahl eine Anfangspopulation bestimmt und deren Fitness<sup>229</sup> berechnet wurde, kommt es zu einer iterativen Anwendung der folgenden Schritte [HesJ92 S.5-8; Wag04 S.28-30; Niss97 S.12-16 u. S.197-200; Dav91 S. 1-5 u. S.13-20; Aff05 S.16-22; SiDe08 S.1-11 u. S.29-33 u. S.39-42; Kinn94 S.66-72]:

- Selektion von Individuen in Abhängigkeit ihrer Fitness.
- Erzeugen einer neuen Population durch Anwendung genetischer Operatoren (Reproduktion, Crossover und Mutation).
- Bestimmung der Fitness der neuen Individuen.

Die Abbildung 150 stellt grafisch den Ablauf bei genetischen Algorithmen dar:

---

<sup>229</sup> Über eine geeignete, zu definierende Fitnessfunktion.

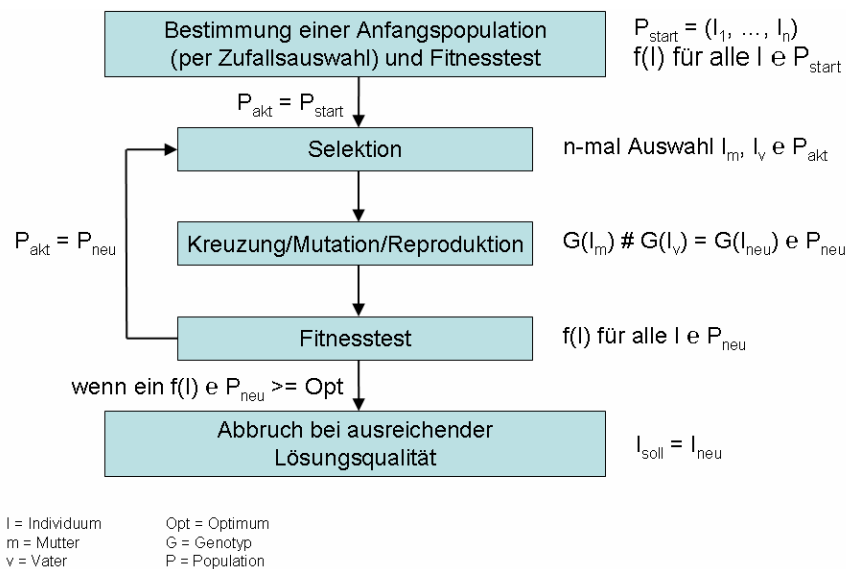


Abbildung 150: Prinzipdarstellung des Ablaufs genetischer Algorithmen [BauT 07]

#### 7.4.4.2 Entwicklung eines genetischen Algorithmus für die vorliegende Problemstellung

Der im Folgenden verwendete genetische Algorithmus soll Aufschluss darüber geben, wie die Einbestellungsreihenfolge von Patienten - bezogen auf die beiden Hauptpatientengruppen - gewählt werden muss, um die durchschnittlichen Wartezeiten der Patienten möglichst minimal zu halten. Die Hauptpatientengruppe wird eingeteilt in „Patienten mit Therapie“ und „Patienten ohne Therapie“<sup>230</sup>.

Gelingt es, eine optimierte Einbestellungsreihenfolge der Patienten auf Basis der beiden Hauptpatientengruppen herauszuarbeiten, können die entsprechenden Ergebnisse dann in das Soll-Modell und anschließend in den operativen Krankenhausalltag übertragen werden. Dieses Vorgehen wurde gewählt, da in der betrachteten Tagesklinik die Terminvergabe nicht direkt mittels genetischer Algorithmen geplant werden bzw. erfolgen kann. Zum einen besitzt die Klinik derzeit keinerlei eigene Tools (d.h. Simulationssysteme), um genetische

<sup>230</sup> Patienten ohne Therapie werden auch als Mitbehandler bezeichnet.

Algorithmen einzusetzen. Ebenso wenig ist der Einsatz entsprechender Lösungen nach heutigem Stand in der Tagesklinik geplant. Ein weiterer Grund liegt darin, dass Patienten ihre Termine (Behandlungstag und Uhrzeit) sequentiell und teils mit unterschiedlichem zeitlichen Vorlauf erhalten. Ein bestimmter Behandlungstag „füllt“ sich also erst nach und nach. I.d.R. bekommen Patienten ihren nächsten Behandlungstermin bereits mehrere Tage bzw. Wochen im Voraus genannt. Würde man im operativen Tagesgeschäft der Tagesklinik die letztendliche Einbestellungsreihenfolge je Behandlungstag mit Hilfe eines genetischen Algorithmus ermitteln, so könnte den Patienten bei Terminvergabe zuerst nur der jeweilige Behandlungstag, nicht jedoch zugleich die Uhrzeit ihrer Einbestellung mitgeteilt werden. Die Vergabe der Uhrzeit und damit die finale Festlegung der konkreten Einbestellungsreihenfolge könnte in diesen Fällen erst dann mittels eines genetischen Algorithmus erfolgen, wenn für den betreffenden Behandlungstag bereits die maximale Anzahl an Patienten ungeachtet ihrer letztendlichen Reihenfolge terminiert wurde.<sup>231</sup> Ein solches Vorgehen wäre im vorliegenden Fall für Patienten mit weitem Anreiseweg zur Tagesklinik problematisch, wenn die Uhrzeiten der Einbestellung zu kurzfristig bekannt gegeben werden. Der Koordinationsaufwand würde signifikant steigen. Zudem schließt sich dann ein weiterer Problemkreis an: Eine allzu kurzfristige Festlegung der Einbestellzeiten hätte ebenfalls Auswirkungen auf die Terminplanung der angrenzenden Fachbereiche und den dort durchzuführenden Untersuchungen. Die abteilungsübergreifende Terminabstimmung würde in solchen Fällen damit ebenfalls erheblich erschwert.

Zur Vollständigkeit sei erwähnt, dass sich genetische Algorithmen auch zur dynamischen Optimierung während des klinischen Tagesablaufs eignen. Dies bietet sich insbesondere dann an, wenn in klinischen Bereichen Störfaktoren<sup>232</sup> z.B. in Form von Notfallpatienten, welche ungeplant über den Tag verteilt auftreten können, existieren. Ohne eine dynamische Optimierung würde in solchen Fällen ein im Sinne einer Prozessoptimierung ursprünglich optimal geplanter Tagesablauf verändert und nachteilig beeinflusst werden. Im

---

<sup>231</sup> Würde ein genetischer Algorithmus zur Optimierung angewandt, bevor für alle Patienten eines Behandlungstages die Termine vergeben wurden, würde sich die Einbestellungsreihenfolge mit jedem neu hinzukommenden Patienten wieder ändern.

<sup>232</sup> Im Sinne von Störgrößen.

vorliegenden Fall der Tagesklinik ist dieser Aspekt jedoch vernachlässigbar, da nur in seltenen Fällen Notfallpatienten behandelt werden müssen.

Der Aufbau des in dieser Arbeit verwendeten genetischen Algorithmus sieht wie folgt aus: Die Größe der Anfangspopulation wurde auf 20 Individuen festgelegt. Jedes Individuum enthält als Merkmal sowohl die jeweiligen Patienten-IDs wie auch die Einbestellungszeitpunkte am betrachteten Behandlungstag. Jedes einzelne „Gen“ eines „Chromosoms“ besteht somit aus zwei zusammengehörigen Zahlenwerten. Abbildung 151 zeigt den Aufbau am Beispiel eines Individuums.

Einbestellungszeitpunkt (Uhrzeit)	07:30	07:38	07:45	08:14	08:38	09:17 ... 11:27	11:58	12:17
Patienten-ID	9	21	33	14	34	10 ... 1	22	34

Abbildung 151: Grundlegender Aufbau des Individuums

Nachdem es das Ziel ist, die Wartezeiten über eine optimierte Reihenfolgeplanung der Patienten zu reduzieren, wurde für den Fitnessstest folgende Fitnessfunktion  $F(I)$  gewählt:

$$F(I) = 1 - \left( \left[ \frac{\sum_{P(i)=1}^n (\text{Wartezeiten}_p)}{n} \right] / \text{Max}(\emptyset \text{ Wartezeit}) \right)$$

mit:

$$\text{Max}(\emptyset \text{ Wartezeit}) = \text{Max}_I \left( \frac{\sum_{P(i)=1}^n (\text{Wartezeiten}_p)}{n} \right)$$

$n$  = Anzahl von Patienten je Individuum

$I$  = Individuum

$P$  = Patient

$F(I)$  = Fitnessfunktion

$\text{MAX}(\emptyset \text{ Wartezeit})$  = Maximale durchschnittliche Wartezeit der jeweils betrachteten Population

Ferner gilt:

$$F_{\max}(l) = 1, \text{ wenn gilt: } \emptyset \text{ Wartezeit} = 0$$

$$F_{\min}(l) = 0, \text{ wenn gilt: } \emptyset \text{ Wartezeit} = \text{Max}(\emptyset \text{ Wartezeit})$$

Beträgt die maximale durchschnittlich gemessene Wartezeit über alle Individuen einer Population z.B. 540 Minuten, so ergeben sich gemäß obiger Fitnessfunktion folgende Wertepaare:

Fitness:	0	...	0,25	...	0,5	...	0,75	...	1
	↑↓		↑↓		↑↓		↑↓		↑↓
Durchschnittliche Wartezeit:	540	...	405	...	270	...	135	...	0

Nachdem im vorliegenden Fall ein kombinatorisches Optimierungsproblem gelöst werden soll, ist bei Anwendung des Crossovers zu beachten, dass sichergestellt ist, dass keine Patienten-ID im jeweils betrachteten Individuum mehrfach vorkommt. In der Literatur werden zur Lösung dieses Problems sog. Sequenzoperatoren wie z.B. Uniform Order-Based Crossover oder Edge-Recombination vorgeschlagen [Niss97 S.58-62]. Weiterhin ist zu berücksichtigen, dass in dem betrachteten Optimierungsproblem jedes Gen eines Individuums selbst als Vektor mit den Entscheidungsvariablen „Patienten-ID“ und „Uhrzeit“ dargestellt wird.

In der vorliegenden Arbeit konnte die oben beschriebene Problematik trotz Anwendung eines 1-Punkt-Crossovers dadurch gelöst werden, indem Crossover nur auf die Uhrzeit angewandt wurde. Die nachfolgende Abbildung 152 zeigt die entsprechende Vorgehensweise beim Crossover:



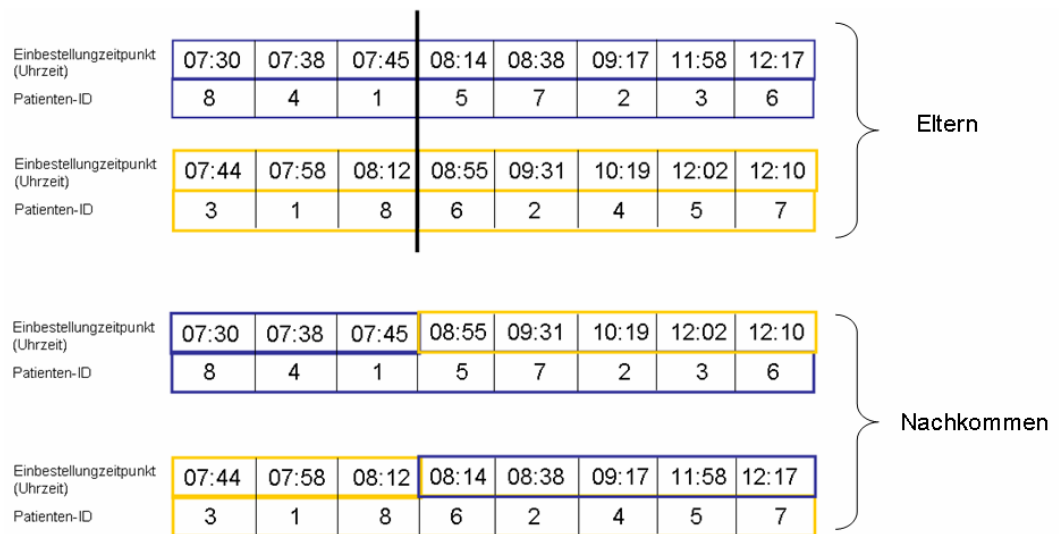


Abbildung 152: Darstellung der Vorgehensweise beim Crossover

Eine Besonderheit der Mutation für die hier verwendeten Individuen ist, dass Patienten-IDs und Uhrzeiten im Rahmen der Mutation getrennt voneinander betrachtet werden. Der Grund für das gewählte Vorgehen liegt darin, dass bei Mutation eines „vollständigen“ Gens – also bei gleichzeitiger Berücksichtigung der Uhrzeit und der Patienten-ID – die Reihenfolge der Patienten identisch bleiben würde. Dies liegt daran, dass die Reihenfolge direkt durch die jeweilige Uhrzeit und nicht durch die Position eines Gens auf dem Vektor festgelegt. Da es sich bei der Mutation der Patienten-IDs um einen „Platztausch“ handelt, findet diese nur dann statt, wenn mindestens zwei Gene zur Mutation ausgewählt wurden. Abbildung 153 zeigt das Vorgehen an einem Beispiel.

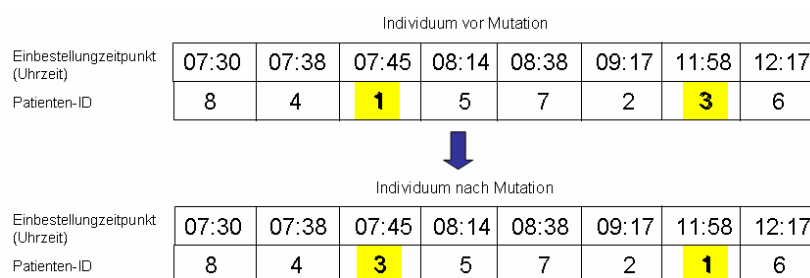


Abbildung 153: Darstellung der Vorgehensweise bei Mutation

Neben der Mutation der Patienten-IDs wurde auch die Uhrzeit mutiert. Für jedes Gen wurde stochastisch festgelegt, ob eine Mutation der Uhrzeit stattfindet oder nicht. Die Ausprägung jeder zur Mutation bestimmten Entscheidungsvariablen „Uhrzeit“ wurde durch Abziehen oder Hinzuaddieren einer gleichverteilten Zufallsgröße  $z$  in der Einheit „Minuten“ mutiert. Als Nebenbedingung galt hier, dass die konkreten Ausprägungen der Entscheidungsvariable „Uhrzeit“ nicht kleiner als 7:30 Uhr und nicht größer als 14:30 Uhr sein dürfen. Der Minimalwert ergibt sich, da die betrachtete Klinik nicht vor 7:30 Uhr öffnet. Der Maximalwert ergibt sich dadurch, dass Einbestellungen nach 14:30 Uhr im Ist-Modell dazu führten, dass die Behandlung dieser Patienten teilweise nicht bis um 17:00 Uhr (=Ende der Öffnungszeiten) abgeschlossen war.

Zur Bestimmung der Auswahlwahrscheinlichkeit der genetischen Operatoren wurde im Rahmen von Testdurchläufen untersucht, welche Wahrscheinlichkeiten in Bezug auf die eingesetzten genetischen Operatoren die besten Ergebnisse unter Berücksichtigung der jeweils benötigten Simulationsdauer erzielen. Dies führte zu folgenden Wahrscheinlichkeiten: (1) Reproduktion mit einer Wahrscheinlichkeit von 15%, (2) Crossover mit einer Wahrscheinlichkeit von 85% und (3) Mutation mit einer Wahrscheinlichkeit von 30%.<sup>233</sup> Aus Gründen der Vereinfachung ist die Mutationswahrscheinlichkeit zur Mutation der Uhrzeit und zur Mutation der Patienten-ID identisch. Als Wertebereich für die Zufallsgröße  $z$ , welche zur Mutation der Uhrzeit dient, ergab sich in den Testdurchläufen ein als für die vorliegende Problemstellung akzeptabler Wertebereich von ( $z=1, 2, \dots, 20$ ).

Auf Basis der o.g. Rahmenbedingungen wurde mit dem MLDesigner das in Abbildung 154 dargestellte Simulationsmodell entwickelt.

---

<sup>233</sup> Die Summe der Wahrscheinlichkeiten aus Reproduktion und Crossover ergibt 1.

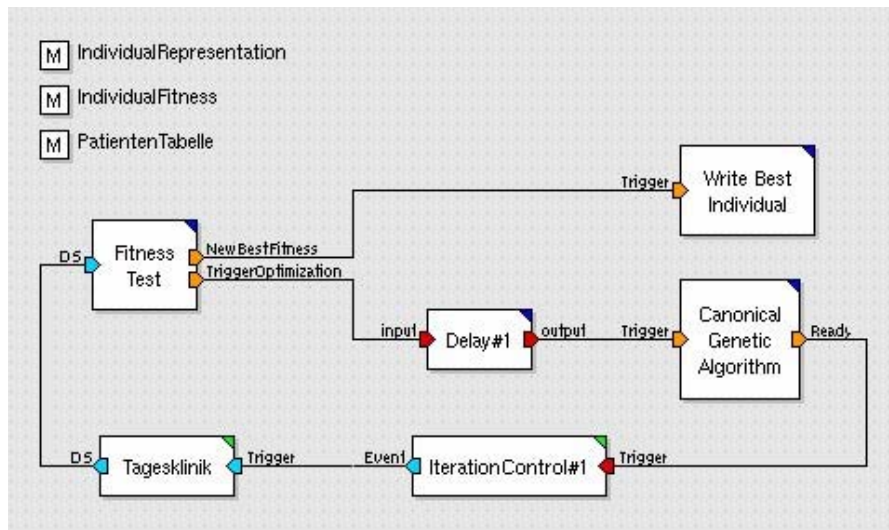


Abbildung 154: Screenshot des mit dem MLDesigner entwickelten Modells

Der Modellbaustein „FitnessTest“ bewertet das aktuelle Individuum anhand der vorgegebenen Fitness-Funktion. Das Primitiv „Write Best Individual“ schreibt die neue beste Fitness inklusive der aktuellen Parametrisierung<sup>234</sup> des jeweils neu gefundenen besten Individuums in eine Ausgabedatei. Der Modellbaustein „Canonical Genetic Algorithm“ generiert neue Individuen. Er wird dabei durch verschiedene Parameter, wie zum Beispiel die Populationsgröße und Mutationswahrscheinlichkeit<sup>235</sup>, gesteuert.

Das Modul „IterationControl#1“ sorgt dafür, den Vektor der Patiententabelle aus dem Modell der Tagesklinik und den des genetischen Algorithmus zu synchronisieren. Während der Simulation werden die Werte des Vektors, den der genetische Algorithmus verwendet, in den Vektor, den das Modell der Tagesklinik verwendet, geschrieben. Dieses Vorgehen war erforderlich, da der genetische Algorithmus aus Gründen der Modellflexibilität nicht direkt in das Modell der Tagesklinik integriert wurde.

<sup>234</sup> Im vorliegenden Fall die konkrete Einbestellungsreihenfolge.

<sup>235</sup> Die Mutationswahrscheinlichkeit (Mittelwert) definiert, wieviele Vektor-Werte durch den genetischen Algorithmus zufällig geändert werden.

#### 7.4.4.2.3 Ergebnisse der Anwendung des genetischen Algorithmus

Die Optimierung der Einbestellungsreihenfolge wurde anhand der gesammelten Daten von 40 Arbeitstagen durchgeführt. Zugrunde gelegt wurde das in Teilen bereits optimierte Soll-Modell der Tagesklinik (→ Eliminierung redundanter Arbeitsschritte gem. Kapitel 7.4.2) – allerdings noch ohne Optimierung der Einbestellungsreihenfolge der Patienten. Um qualifizierte Ergebnisse zu erhalten, betrug die Dauer eines jeden Simulationslaufs 24 Stunden.

Tabelle 6 zeigt beispielhaft an 5 ausgewählten Arbeitstagen die erzielten Verbesserungen. Gegenübergestellt werden jeweils die in den durchgeführten Simulationsexperimenten durchschnittlich erzielten Reduktionen der Wartezeit je Arbeitstag (in %).

	Arbeitstag 1	Arbeitstag 2	Arbeitstag 3	Arbeitstag 4	Arbeitstag 5
Erzielte Verbesserung: "Modell mit genetischem Algorithmus" gegenüber "Ist-Modell"	60,0%	59,1%	67,6%	62,2%	68,9%

Tabelle 6: Ergebnisse ausgewählter Simulationsläufe

Wie die Werte in obiger Tabelle zeigen, konnte der genetische Algorithmus signifikante Verbesserungen gegenüber dem Vergleichsmodell erzielen. Die in den 40 zugrunde gelegten Arbeitstagen gemessenen Verbesserungen der Wartezeit liegen zwischen 68,9% und 49,1%. Die nähere Analyse der Einbestellungsreihenfolge der Patienten ergab folgende Struktur: Die Einbestellung des überwiegenden Anteils von Therapie-Patienten erfolgte i.d.R. im ersten Drittel des jeweiligen Arbeitstages. Nach 12:00 Uhr wurden nur an 4 der betrachteten Arbeitstage zwischen 1 und maximal 3 Therapiepatienten einbestellt. Der überwiegende Anteil von „Patienten ohne Therapie“ wurde hingegen erst nach 9:30 Uhr einbestellt. Abbildung 155 zeigt dies exemplarisch am Beispiel von drei ausgewerteten Behandlungstagen.

Laufende Nr.	Uhrzeit (Einbestellung)	Patientengruppe	Laufende Nr.	Uhrzeit (Einbestellung)	Patientengruppe	Laufende Nr.	Uhrzeit (Einbestellung)	Patientengruppe
1	7:30	Therapie	1	7:30	Therapie	1	7:36	Therapie
2	7:37	Mitbehandler	2	7:30	Therapie	2	7:36	Mitbehandler
3	7:45	Therapie	3	7:30	Therapie	3	7:40	Therapie
4	7:56	Therapie	4	7:32	Mitbehandler	4	7:47	Therapie
5	8:02	Therapie	5	7:46	Therapie	5	8:00	Therapie
6	8:10	Therapie	6	8:00	Therapie	6	8:01	Therapie
7	8:29	Therapie	7	8:00	Therapie	7	8:21	Therapie
8	8:39	Therapie	8	8:00	Therapie	8	8:29	Therapie
9	8:44	Therapie	9	8:14	Therapie	9	8:40	Therapie
10	8:46	Mitbehandler	10	8:16	Therapie	10	8:42	Therapie
11	9:01	Therapie	11	8:22	Therapie	11	8:55	Therapie
12	9:14	Therapie	12	8:34	Therapie	12	9:07	Therapie
13	9:29	Therapie	13	8:34	Therapie	13	9:23	Mitbehandler
14	9:34	Therapie	14	9:08	Mitbehandler	14	9:30	Therapie
15	9:37	Therapie	15	9:50	Mitbehandler	15	9:38	Therapie
16	9:53	Mitbehandler	16	9:52	Therapie	16	9:52	Mitbehandler
17	9:58	Mitbehandler	17	10:02	Mitbehandler	17	10:03	Mitbehandler
18	10:10	Mitbehandler	18	10:20	Mitbehandler	18	10:18	Mitbehandler
19	10:14	Mitbehandler	19	10:30	Mitbehandler	19	10:24	Mitbehandler
20	10:20	Mitbehandler	20	10:38	Mitbehandler	20	10:33	Mitbehandler
21	10:23	Mitbehandler	21	10:48	Mitbehandler	21	10:35	Mitbehandler
22	10:26	Mitbehandler	22	10:48	Mitbehandler	22	10:39	Mitbehandler
23	10:27	Mitbehandler	23	11:04	Mitbehandler	23	10:48	Therapie
24	11:09	Mitbehandler	24	11:08	Mitbehandler	24	10:59	Mitbehandler
25	11:21	Mitbehandler	25	11:10	Mitbehandler	25	11:03	Mitbehandler
26	11:22	Therapie	26	11:34	Therapie	26	11:20	Mitbehandler
27	11:25	Mitbehandler	27	11:34	Mitbehandler	27	11:54	Mitbehandler
28	11:36	Mitbehandler	28	11:44	Mitbehandler	28	11:58	Mitbehandler
29	11:42	Mitbehandler	29	11:56	Mitbehandler	29	11:59	Mitbehandler
30	11:49	Mitbehandler	30	12:02	Mitbehandler	30	12:04	Therapie
31	11:51	Therapie	31	12:34	Mitbehandler	31	12:08	Mitbehandler
32	12:14	Mitbehandler	32	12:52	Mitbehandler	32	12:32	Mitbehandler
33	12:17	Mitbehandler	33	12:56	Mitbehandler	33	12:35	Mitbehandler
34	12:45	Mitbehandler	34	12:59	Mitbehandler	34	12:37	Mitbehandler
35	12:45	Mitbehandler	35	13:03	Mitbehandler	35	12:46	Mitbehandler
36	12:52	Mitbehandler	36	13:06	Mitbehandler	36	13:30	Mitbehandler
37	12:55	Mitbehandler	37	13:10	Mitbehandler	37	13:35	Mitbehandler
38	12:57	Mitbehandler				38	13:37	Mitbehandler
39	12:59	Mitbehandler				39	13:41	Mitbehandler
40	13:09	Mitbehandler				40	13:43	Mitbehandler
						41	13:46	Mitbehandler

Behandlungstag 1

Behandlungstag 2

Behandlungstag 3

Abbildung 155: Einbestellungsreihenfolge für drei exemplarische Behandlungstage bei Anwendung des genetischen Algorithmus

In der bisherigen Einbestellungspraxis der Tagesklinik wurden vor allem auch vor 9:30 Uhr vermehrt Mitbehandler, also Patienten ohne Therapie, einbestellt. Therapiepatienten kamen teilweise auch noch nach 15:00 Uhr in die Tagesklinik.

Das oben beschriebene Ergebnis des genetischen Algorithmus lässt sich wie folgt begründen: Therapiepatienten erhalten Chemotherapien bzw. Antikörper, deren Gabe bis zu mehreren Stunden dauern kann. Ferner dauern deren Vor-/Fachuntersuchungen i.d.R. ebenfalls länger als bei Patienten, die keine Chemotherapie erhalten. Je später Therapiepatienten folglich einbestellt werden, desto weiter verschiebt sich das Ende der letzten Behandlung zeitlich nach hinten

und beeinflusst damit auch die Öffnungszeit der Tagesklinik nachteilig. Eine frühzeitige Einbestellung der Therapiepatienten führt ferner zu einem insgesamt früheren Beginn der Liegenbelegung und damit zu einer besseren Auslastungsmöglichkeit der Patientenliegen.

#### **7.4.4.3 Übertragung der bisherigen Ergebnisse in das Soll-Modell und Durchführung weiterer Prozessoptimierungen**

Im Soll-Modell wurde in einem ersten Schritt die Reihenfolge der Einbestellung von Patienten dahingehend geändert, dass Patienten mit lang laufenden Therapien und längeren ärztlichen (Fach-)Untersuchungen zuerst einbestellt wurden. Vice versa wurden Patienten, die Mitbehandler<sup>236</sup> sind und daher kürzere Behandlungsdauern haben, i.d.R. nach der o.g. Patientengruppe einbestellt. Diese Erkenntnisse ergaben sich aufgrund der Optimierung der Einbestellungsreihenfolge durch den weiter oben beschriebenen genetischen Algorithmus. Im Soll-Modell führten die Veränderungen der Einbestellungsreihenfolge der Patienten zu einer Verbesserung der Wartezeiten um durchschnittlich 34% bzw. 54 Minuten. Gleichzeitig verbesserten sich auch die Durchlaufzeiten im Soll-Modell. Letzteres wird in der nachfolgenden Abbildung 156 dargestellt. Diese zeigt am Beispiel von 5 Arbeitstagen die Veränderung der Liegenbelegung inklusive dem Ende der letzten Chemotherapie. Das durchschnittliche Ende der letzten Chemotherapie liegt im Soll-Modell (Grafiken auf der rechten Seite der Abbildung) gegenüber dem Ist-Modell (Grafiken auf der linken Seite der Abbildung) deutlich früher.

---

<sup>236</sup> Patienten, die keine Chemotherapie bzw. keine Antikörper bekommen.

### Ist-Modell

### Soll-Modell nach Optimierung der Einbestellungsreihenfolge

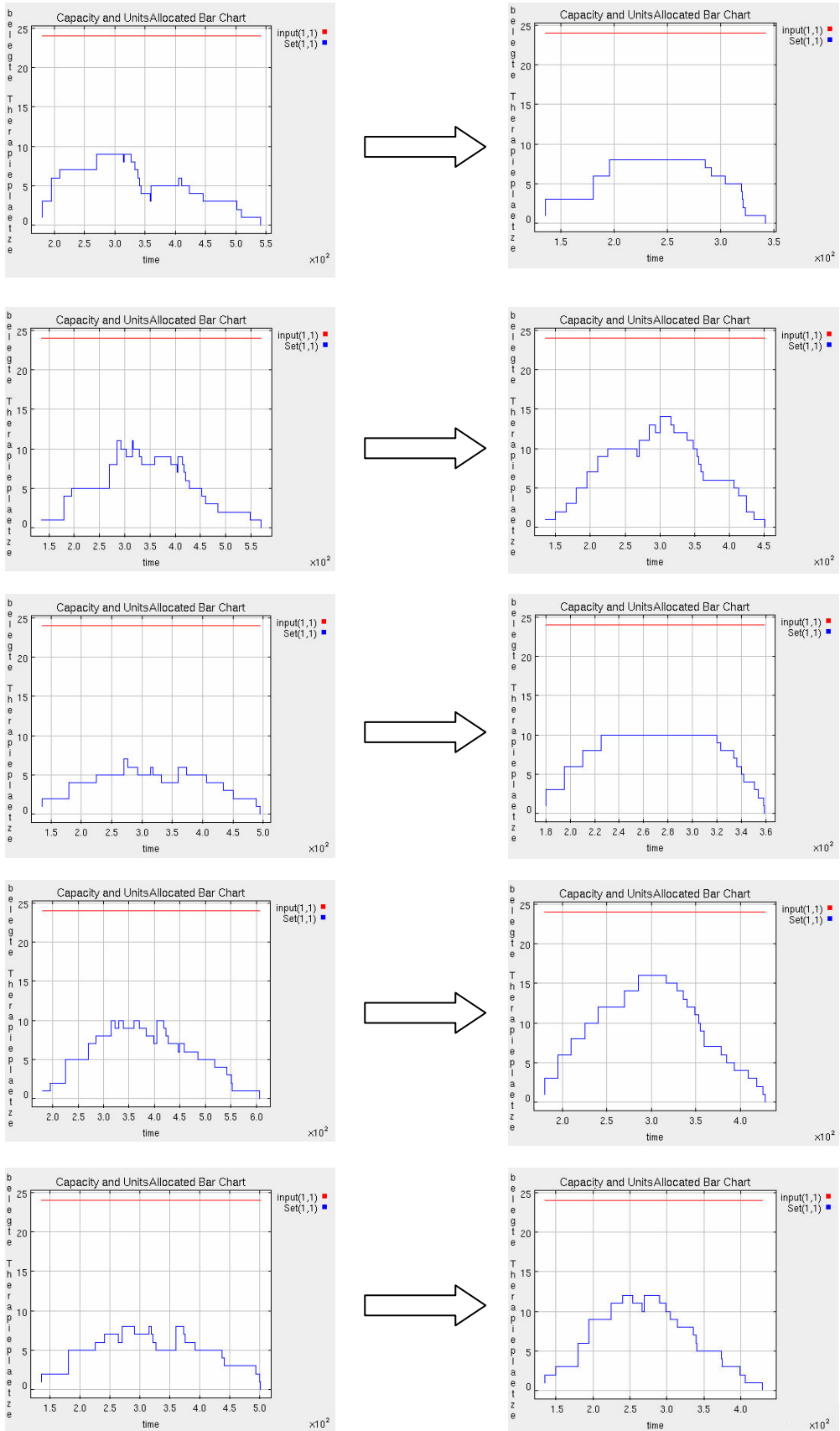


Abbildung 156: Veränderung der Liegenbelegung nach Optimierung der Patienteneinbestellung

Eine weitere Optimierung der Patienteneinbestellung brachte zusätzliche positive Effekte für die Warte- und Durchlaufzeiten: Im Soll-Modell wurden ausgewählte Patienten am Vortag der eigentlichen Therapie zur Blutentnahme einbestellt. Bei guten Laborwerten kann die Therapie dann bereits am Tag vor der Therapiegabe in der Apotheke bestellt werden. Dies entlastet gleichzeitig auch die Apotheke, da solche Therapien dann am Nachmittag - einer „ruhigeren“ Phase im Tagesablauf der Apotheke - zubereitet und an die Tagesklinik ausgeliefert werden können. Die betroffenen Patienten können dann am Tag der Therapiegabe bereits früh morgens angehängt werden<sup>237</sup> und müssen hier praktisch keine Wartezeiten mehr in Kauf nehmen.

Im Soll-Modell wird der vorgezogenen Blutentnahme und dem Ausliefern der Therapie am Vortag durch die Funktion des Primitive „Lese Blutwert“ und durch die Verzweigung „Blutwert vorhanden“ Rechnung getragen (vgl. Abb. 157).

---

<sup>237</sup> Als Anhängen bezeichnet man die Gabe der Therapie als Infusion.



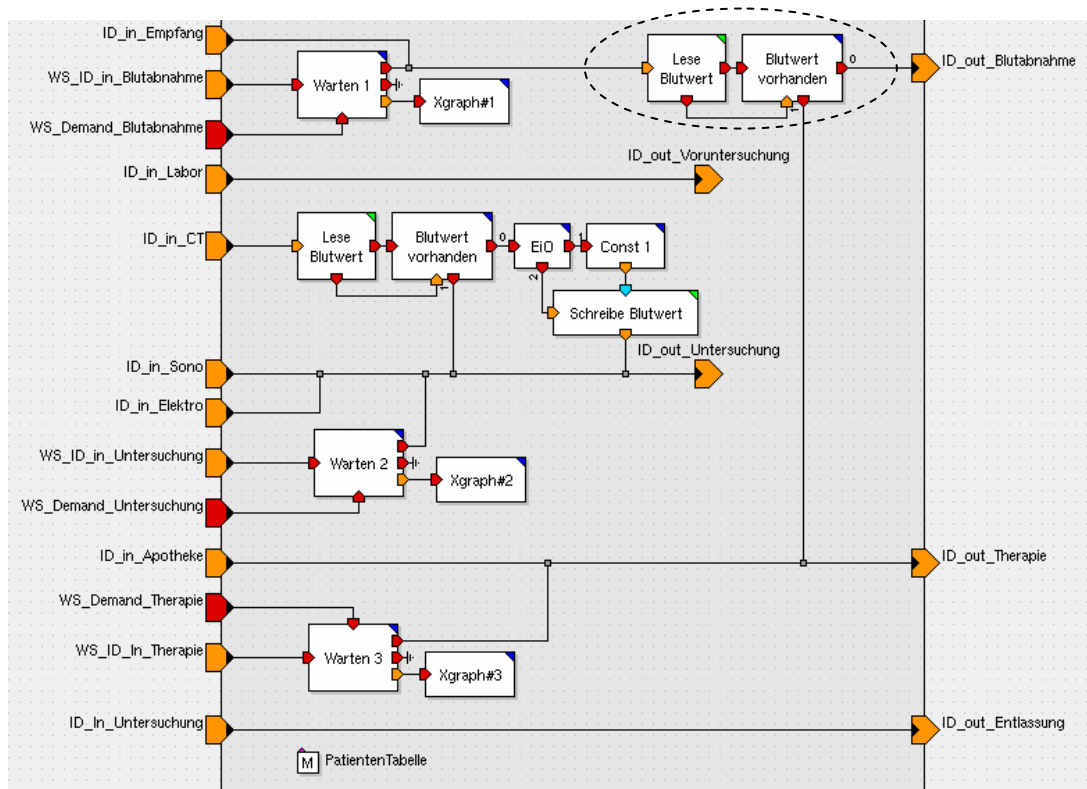


Abbildung 157: Optimiertes Modell zur Abbildung von bereits am Vortag erstellten Laborwerten  
[Kühn06 S.79]

Der optimierte Prozessablauf der Patienteneinbestellung auf Basis der weiter oben beschriebenen Optimierungen ist als Ablaufdiagramm in Abbildung 158 dargestellt. In den Abbildungen 173 bis 176 im Anhang 1 sind alle optimierten Soll-Prozesse der Tagesklinik in Form von Ablaufdiagrammen dargestellt.

Abbildung 159 zeigt am Beispiel von 5 Arbeitstagen den Vergleich der Liegenbelegung nach Implementierung der Einbestellungsreihenfolge mit der Liegenbelegung, die sich nach Implementierung der Einbestellungsreihenfolge und der vorgezogenen Blutentnahme im Soll-Modell ergeben. Hier zeigt sich, dass die Gabe der ersten Chemotherapien deutlich früher als im Vergleichsmodell beginnt. Dies bedeutet, dass die ersten Patienten nach Eintreffen in der Tagesklinik weniger lange auf den Beginn ihrer Therapiegabe warten müssen und damit deren Wartezeit verkürzt wird.

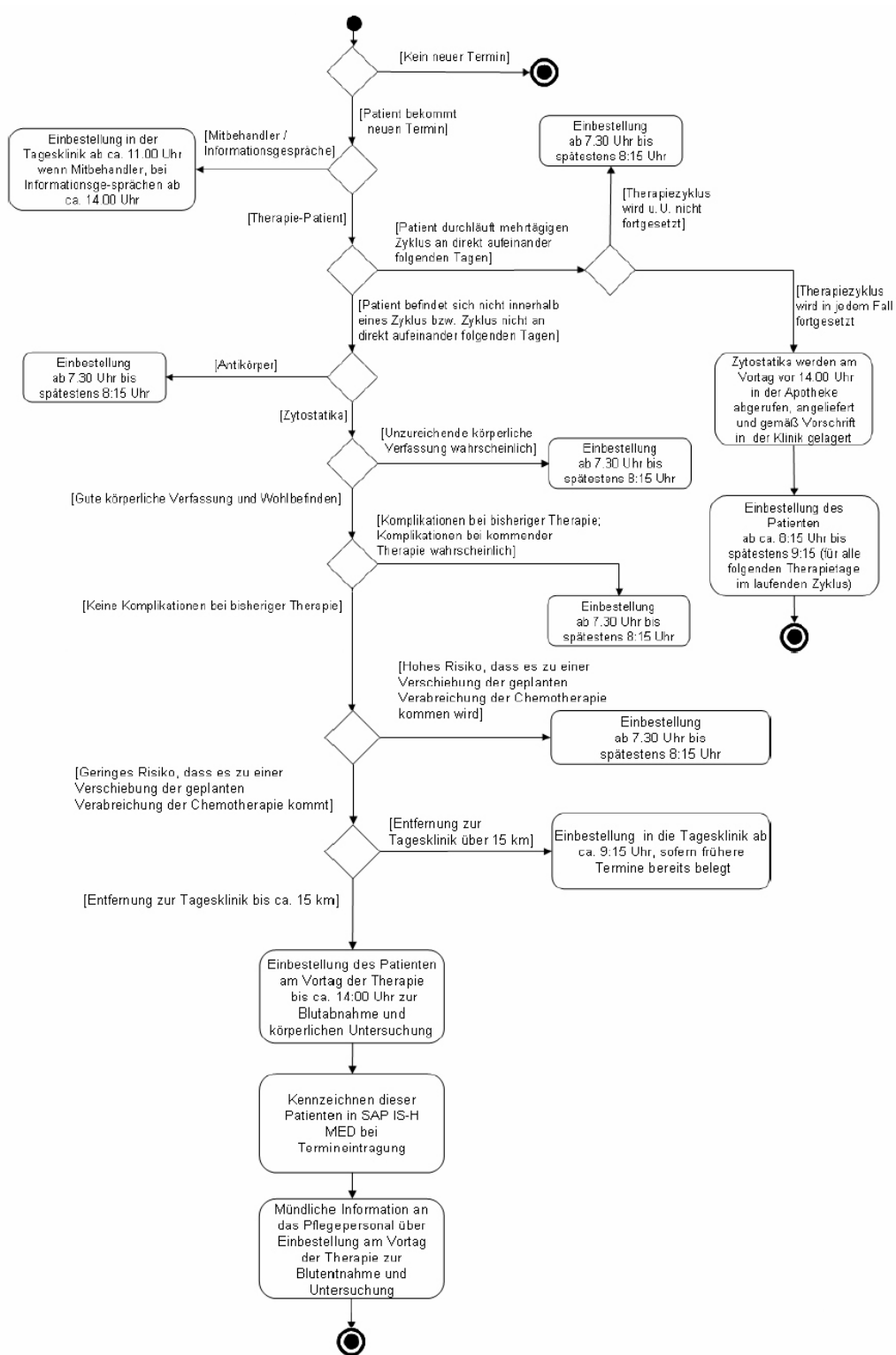


Abbildung 158: Optimierter Soll-Prozess für die Einbestellung von Patienten

Soll-Modell nach Optimierung  
der Einbestellungsreihenfolge

Soll-Modell nach Optimierung der Einbestellungsreihenfolge  
und vorgezogenen Blutentnahmen

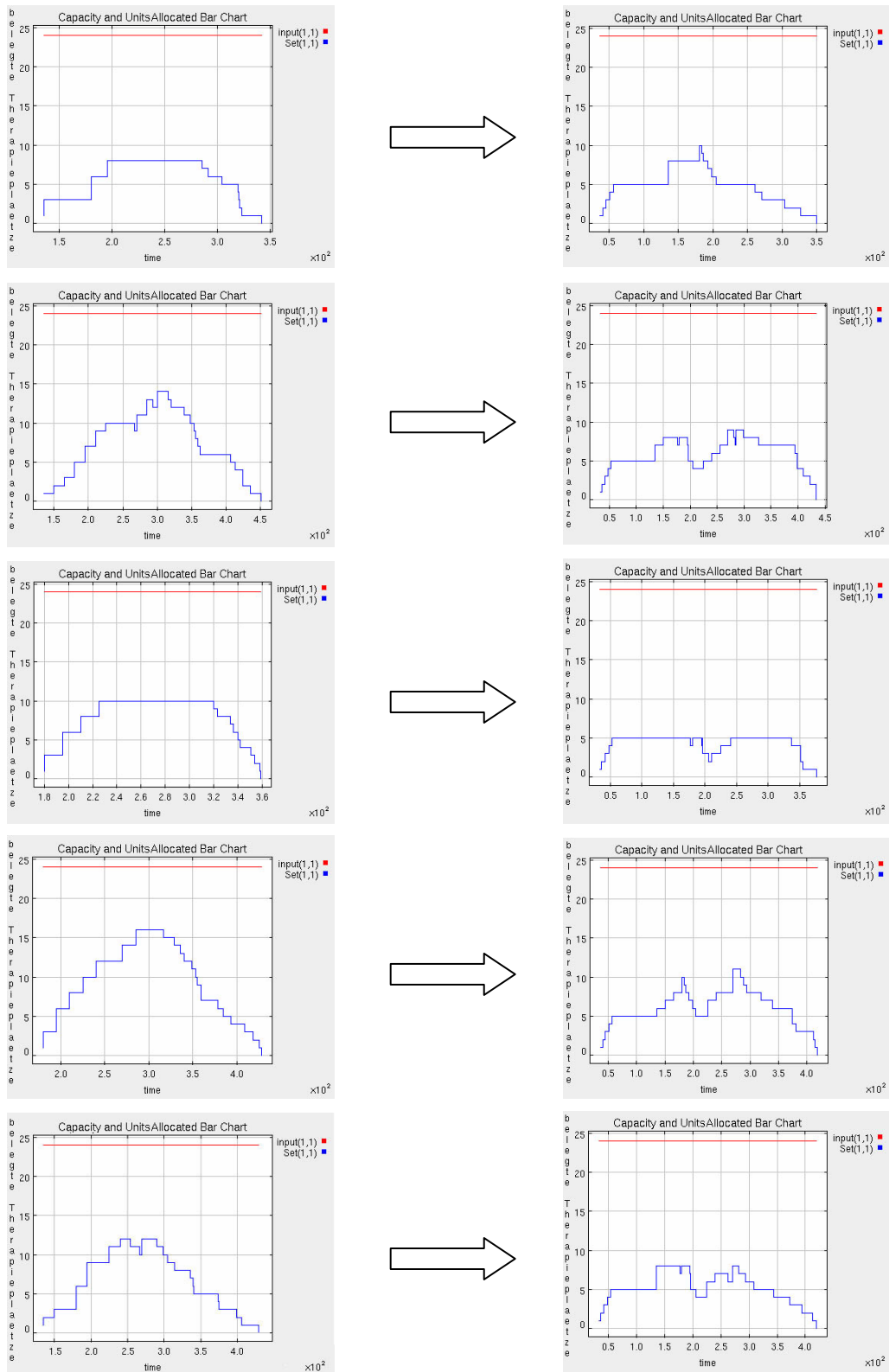


Abbildung 159: Vergleich der Verbesserung der Liegenbelegung nach Implementierung aller Optimierungen mit der Liegenbelegung nach Implementierung des ersten Optimierungsschrittes

Die in den Simulationsläufen ermittelten Werte ergaben sich auf Basis des optimierten Prozessablaufs im Soll-Modell und wurden je Arbeitstag mit den jeweiligen Werten des Ist-Modells verglichen. Die nachstehende Tabelle zeigt am Beispiel von 5 simulierten Arbeitstagen die betreffenden Ergebnisse in Bezug auf den Therapiebeginn, das Ende der letzten Therapie, das Ende des jeweiligen Arbeitstages<sup>238</sup> sowie die mittlere Gesamtwartezeit.

	Arbeitstag 1		Arbeitstag 2		Arbeitstag 3		Arbeitstag 4		Arbeitstag 5	
	Werte des optimierten Modells	Werte des Ist-Modells	Werte des optimierten Modells	Werte des Ist-Modells	Werte des optimierten Modells	Werte des Ist-Modells	Werte des optimierten Modells	Werte des Ist-Modells	Werte des optimierten Modells	Werte des Ist-Modells
Beginn der Therapien	08:03 Uhr	09:46 Uhr	08:03 Uhr	10:05 Uhr	08:03 Uhr	9:58 Uhr	08:03 Uhr	10:36 Uhr	08:03 Uhr	09:17 Uhr
Ende aller Therapien	13:38 Uhr	16:30 Uhr	14:10 Uhr	17:05 Uhr	12:39 Uhr	15:23 Uhr	14:08 Uhr	17:15 Uhr	14:16 Uhr	15:35 Uhr
Ende aller Behandlungen	15:05 Uhr	16:50 Uhr	15:04 Uhr	17:25 Uhr	13:43 Uhr	15:36 Uhr	14:52 Uhr	17:49 Uhr	14:58 Uhr	16:41 Uhr
Mittlere Gesamtwartezeit	83 min	150 min	87 min	162 min	91 min	158 min	90 min	160 min	86 min	151 min

Tabelle 7: Tagesgenauer Vergleich der Werte im optimierten Soll-Modell mit den Werten des Ist-Modells am Beispiel von 5 aufeinander folgenden Arbeitstagen

Die in Tabelle 7 gezeigten Verbesserungen der Ergebniswerte im optimierten Soll-Modell sind dabei repräsentativ für alle simulierten Arbeitstage. Sowohl das Ende der Therapien wie auch das Ende der letzten Behandlung finden deutlich früher statt als im suboptimalen Ist-Modell. Die durchschnittliche Gesamtwartezeit der Patienten verkürzte sich in der Simulationsstudie nach Implementierung aller oben dargestellten Optimierungsschritte auf 89 Minuten. Dies ist eine Verbesserung um 69 Minuten gegenüber der im Ist-Modell gemessenen durchschnittlichen Wartezeit von 2 Stunden 38 Minuten. Prozentual entspricht dies einer Verbesserung um ca. 44%. Berücksichtigt man die in Kapitel 7.3 beschriebene Modellgenauigkeit von 4% bezogen auf die durchschnittliche Gesamtwartezeit der Patienten, so ergibt sich im Mittel noch immer eine Verbesserung um ca. 66 Minuten bzw. ca. 42%.

<sup>238</sup> Entspricht dem Zeitpunkt, zu dem der letzte Patient die Tagesklinik verlässt.

Gleiche Effekte zeichneten sich nach der Implementierung der optimierten Prozesse in der onkologischen Tagesklinik ab. Über einen betrachteten Zeitraum von 25 Arbeitstagen<sup>239</sup> endete die letzte Behandlung aufgrund verkürzter Wartezeiten jeweils zwischen 14:00 Uhr und 15:50 Uhr. Die Tagesklinik hat daraufhin die neue Öffnungszeiten - wie ursprünglich angestrebt - von 17:00 Uhr auf 16:00 Uhr verkürzt. Aufgrund der verbesserten Effizienz der Prozesse wurde zudem ein seitens der Geschäftsführung ursprünglich angestrebter Personalaufbau obsolet.

---

<sup>239</sup> Gerechnet ab dem Zeitpunkt der Implementierung der optimierten Prozesse.

## 8 Fazit und Ausblick

Durch die Entwicklung der standardisierten, validierten und wieder verwendbaren Modellkomponenten und deren Einsatz bei der Modellierung, konnte auf effiziente Weise die ambulante Tagesklinik im Modell abgebildet und anschließend mit dem MLDesigner simuliert werden. Nach erfolgter Optimierung der Prozesse im Modell wurden diese in das Realsystem überführt. So wurde sichergestellt, dass bei der Prozessoptimierung keine ungewollten Nebeneffekte auftreten oder Patienten zu Schaden kommen. Zur Durchführung der Simulationsstudie war der Einsatz eines leistungsstarken Simulationssystems unabdinglich. Nur so konnten die komplexen Prozesse effizient modelliert, simuliert, analysiert und optimiert werden. Die Ergebnisse der Prozessoptimierung führten in der betrachteten Tagesklinik zu signifikanten Verbesserungen sowohl bei der Durchlaufzeit der Patienten als auch beim Ressourceneinsatz. Hält man sich vor Augen, dass angrenzende klinische Bereiche und Disziplinen<sup>240</sup>, die bei der Behandlung der Patienten der Tagesklinik involviert sind, nicht Teil der vorliegenden Prozessoptimierung waren, liegt die Vermutung nahe, dass es weitere, sehr wahrscheinlich beträchtliche Optimierungspotentiale in dem betrachteten Krankenhaus gibt.

Die enge Zusammenarbeit mit Ärzten und Pflegepersonal führte dazu, dass sich das klinische Personal sowohl mit dem Prozessreferenzmodell als auch mit dem Modell der Tagesklinik identifizieren konnte und diese ausreichend versteht. Das Prozessreferenzmodell stellt den konkreten Ansatz für die Standardisierung der Modellierung von Krankenhausprozessen dar. Aufgrund seines modularen Aufbaus kann das Prozessreferenzmodell jederzeit ergänzt bzw. erweitert werden. Dies kann zum einen durch eine weitere Granularisierung bzw. durch Einführen weiterer Prozessebenen erfolgen. Andererseits durch Aufnahme von nicht-klinischen Prozessen, wie z.B. Prozesse in der Verwaltung, dem Einkauf oder dem Management von Patientenbeziehungen<sup>241</sup>. Gleiches gilt für die daraus abgeleiteten validierten Modellkomponenten.

---

<sup>240</sup> Wie z.B. Chirurgie.

<sup>241</sup> Patienten-/Kundenbeziehungsmanagement

Die bereits entwickelten Modellkomponenten können zur Durchführung weiterer Simulationsstudien in Krankenhäuser eingesetzt werden. In diesem Zusammenhang können die Modellkomponenten z.B. zur Analyse und zur Optimierung des Zusammenwirkens zwischen ambulanter und stationärer Behandlung eingesetzt werden. Ein konkreter Anwendungsfall ist in diesem Zusammenhang die Analyse der Qualität der Zusammenarbeit zwischen dem stationären Bereich der Inneren Medizin und dem Herzkatheterlabor einer Klinik. Des Weiteren können die Modellkomponenten als Basis für den Systementwurf, also z.B. dem Neubau von Krankenhäusern, dienen. Ein entsprechend erweitertes Prozessreferenzmodell sowie die daraus abzuleitenden Modellkomponenten können zur Simulation der Aufbau- und Ablauforganisation - bereits in der Planungsphase eines neu zu errichtenden Krankenhauses - zum Einsatz kommen.

Bei der Optimierung der Prozesse der Tagesklinik zeigte sich der entwickelte genetische Algorithmus als besonders effektiv. Durch die optimierte Einbestellungsreihenfolge der Patienten wurden die durchschnittlichen Wartezeiten von Patienten der Tagesklinik signifikant reduziert.

Durch die Einbindung genetischer Algorithmen in bestehende Krankenhausmanagementsysteme könnten auch dynamische Optimierungen während des klinischen Tagesablaufs durchgeführt werden. Dies erscheint vor allem in klinischen Bereichen sinnvoll, wo sich aufgrund von auftretenden Notfällen (=Notfallpatienten) nicht planbare Störungen im Prozessablauf ergeben. Dem kann mit bisherigen IT-Systemen in Kliniken noch nicht Rechnung getragen werden.

In weiteren Untersuchungen könnte die Ausweitung der Betrachtungen dieser Arbeit auf die Optimierung der transsektoralen Integration<sup>242</sup> erfolgen. Dies ist dann von entscheidender Bedeutung, wenn Patienten aufgrund ihres Krankheitsbildes sowohl von ihrem niedergelassenen Arzt, im Krankenhaus und anschließend in einer Reha-Einrichtung behandelt werden. Die Zusammenarbeit zwischen niedergelassenen Ärzten, Krankenhäusern und Reha-Einrichtungen

---

<sup>242</sup> Unter transsektoraler Integration wird hier die effiziente Zusammenarbeit von niedergelassenen Ärzten, Krankenhäusern und Reha-Einrichtungen verstanden.

läuft noch immer suboptimal und führt zu Zusatzkosten im Gesundheitswesen und Zusatzbelastungen beim Patienten. Einer der Gründe ist z.B. die Durchführung gleicher Fachuntersuchungen (z.B. Röntgen, Erstellen eines Blutbildes) innerhalb kurzer Zeitspannen sowohl beim niedergelassenen Arzt wie auch im Krankenhaus. Würde zwischen diesen Sektoren eine ausreichende Abstimmung erfolgen, indem u.a. relevante Patienteninformationen und damit auch Untersuchungsergebnisse rechtzeitig an die weiterbehandelnde Stelle gegeben werden, könnten Doppeluntersuchungen am Patienten vermieden werden. Auch hier kann die Simulation einen Beitrag zur Optimierung leisten.

Noch steckt der Einsatz von Simulationssystemen in Krankenhäusern in den Anfängen. Aufgrund der Komplexität von Krankenhausprozessen sowie der Tatsache, dass hier Patienten, also Menschen, im Mittelpunkt stehen, und deshalb direkte, unmittelbare Eingriffe in das Realsystem ohne vorherige Simulation erhebliche Risiken bergen, wird die Simulation in Krankenhäusern in den nächsten Jahren immer mehr zum Einsatz kommen. Dies betrifft neben der Optimierung von klinischen Prozessen auch das Design von Prozessen, z.B. beim Neu- oder Umbau von Krankenhäusern. Diejenigen Anbieter von Gesundheitsdienstleistungen, die frühzeitig die Vorteile der Prozesssimulation erkennen und nutzen, werden sich einen klaren Wettbewerbsvorsprung am Markt verschaffen und ihre Wettbewerbsposition langfristig sichern.



## Literaturverzeichnis

- [ACT04] **Ahern D., Clouse A., Turner R.:** CMMI distilled – A Practical Introduction to Process Improvement, 2nd edition, Addison-Wesley Verlag, 2004
- [Aerz06] **Internet-Quelle:**  
<http://www.aerzteblatt.de/v4/news/news.asp?id=24050>  
Abruf vom 02. Juni 2006
- [Aff05] **Affenzeller M.:** Population genetics and evolutionary computation – Theoretical and practical aspects, in: Reihe Technik und Naturwissenschaften Nr. 46, Trauner Verlag, 2005
- [Ame04] **Ament-Rambow C.:** Die Basis der Corporate Identity, in: Krankenhaus Umschau, Ausgabe 4/2004, 73. Jahrgang, S. 280-283
- [Aris06] **O.V.:** Aris Simulation, Optimierungspotenziale in Geschäftsabläufen ermitteln und umsetzen, Produktbeschreibung der Firma IDS Scheer AG, Saarbrücken, 2006
- [Arzt06] **Internet-Quelle:**  
<http://www.aerztezeitung.de/docs/2000/04/18/072a1901.asp>  
Abruf vom 02. Juni 2006
- [BaBa89] **Bamberg G., Bauer F.:** Statistik, Oldenbourg Verlag, München/Wien 1989
- [BaSe01] **Baesler F., Sepúlveda J.:** Multi-objective Simulation Optimization for a Cancer Treatment Center, in: Proceedings of the 2001 Winter Simulation Conference, 2001

- [Baum90]** **Baumgarten B.:** Petri-Netze – Grundlagen und Anwendungen, Mannheim 1990
- [BauT07]** **Baumann T.:** Vorlesungsskript zum Fach Systemoptimierung, Technische Universität Ilmenau, Fakultät für Informatik und Automatisierung, Fachgebiet System- und Steuerungstheorie, 2007
- [BCNN01]** **Banks J., Carson J.S., Nelson B.L., Nicol D.M.:** Discrete- Event System Simulation, 3rd edition, Prentice Hall, New Jersey, 2001
- [Bec03]** **Becker E.M.:** Gewinnen und orientieren mit DRGs, in: Management und Krankenhaus, 22. Jahrgang, Ausgabe 10/2003, S. 7
- [BeDe94]** **Berens W., Delfmann W.:** Quantitative Planung: Konzeption, Methoden und Anwendungen, Schäffer-Poeschel, Stuttgart 1994
- [Berg04]** **Bergbauer A.:** Six Sigma in der Praxis – Das Programm für nachhaltige Prozessverbesserungen und Ertragssteigerungen, in: Kontakt & Studium, Band 654, expert verlag, 2004
- [BeSc96]** **Becker J., Schütte R.:** Handelsinformationssysteme, Verlag Moderne Industrie, Landsberg/Lech 1996
- [BKP76]** **Bobillier P. A., Kahan, B. C., Probst A. R.:** Simulation with GPSS and GPSS V, New Jersey 1976
- [BLR04]** **Biethahn J., Lackner A., Range M.:** Optimierung und Simulation, Oldenbourg Verlag, München 2004
- [BMB02]** **Barry R., Murcko A., Brubaker C.:** The Six Sigma Book for Healthcare – Improving Outcomes by Reducing Errors, Health Administration Press, Chicago 2002

- [Brau84] **Brauer W.:** Automatentheorie – Eine Einführung in die Theorie endliche Automaten, Stuttgart 1984
- [Brin87] **Brinkkötter D.:** Ein systemtheoretischer Ansatz zur computergestützten Prozess- und Projektplanung, Regensburg 1987 (Dissertation)
- [CaMe05] **O.V.:** Process Maturity Profile, CMMI v.1.1, SCAMPI v1.1 Class A Appraisal Results 2004 Year End Update, Präsentation des Software Engineering Institute, 2005
- [CARD05] **Internet-Quelle:**  
<http://www.cardiologe.de/patient/krankheiten/herzerkrankungen>,  
Abruf vom 12. Juli 2005
- [CBT00] **Cadroy Y., Bossavy J.P., Thalamas C., Sagnard I., Sakariassen K., Boneu B.:** Early potent antithrombotic effect with combined aspirin and a loading dose of clopidogrel on experimental arterial thrombogenesis in humans, in: Circulation 101 (2000), pages 2823-2828
- [CMMI05] **O.V.:** Internet-Abruf vom 08.September 2005:  
<http://www.sei.cmu.edu/cmmi/general/general.html#info#info>
- [Dave93] **Davenport T.H.:** Process Innovation: Reengineering Work through Information Technology, Harvard Business School Press, Boston, Massachusetts 1993
- [Dav91] **Davis L.:** Handbook of genetic algorithms, Van Nostrand Reinhold Verlag, New York, 1991
- [DIN96] **DIN, Deutsches Institut für Normung e.V. (Hrsg.):** Geschäftsprozessmodellierung und Workflow-Management:

Forschungs- und Entwicklungsbedarf im Rahmen der entwicklungsbegleitenden Normung; DIN-Fachbericht 50, Beuth, Berlin/Wien/Zürich 1996

**[Domi06] Internet-Quelle:**

<http://www.dominikus.de/html/presse/Presse2.html>

Abruf vom 03. Juni 2006

**[EFQM\_I03] O.V.:** EFQM - Excellence einführen, herausgegeben von der European Foundation for Quality Management, Brüssel 2003

**[EFQM\_F03] O.V.:** EFQM – Die Grundkonzepte der Excellence, herausgegeben von der European Foundation for Quality Management, Brüssel 2003

**[EFQM\_B97] O.V.:** EFQM – Der European Quality Award, Informationsbroschüre, herausgegeben von der European Foundation for Quality Management, Brüssel 1997

**[EFQM\_A97] O.V.:** EFQM Self-Assessment Guidelines for Companies, herausgegeben von der European Foundation for Quality Management, Brüssel 1997

**[Ethi06] Internet-Quelle:**

<http://www.ethik-21-medizin.de/abstract33.htm>,

Abruf vom 07. Juni 2006

**[FeSe04] Feldmayer J., Seidenschwarz W.:** Marktorientiertes Prozessmanagement – Wie Process Mass Customization Kundenorientierung und Prozessstandardisierung integriert, München/Starnberg 2004

- [Gai95]**      **Gaitanides M.:** Je mehr desto besser? In: Technologie & Management, 44 (1995) 2, S. 70-75, 1995
- [GBB02]**      **Grässle P., Baumann H., Baumann P.:** UML projektorientiert – Geschäftsprozessmodellierung, IT-System-Spezifikation und Systemintegration mit der UML, 2. Auflage, Galileo Press, Bonn 2003
- [GCY02]**      **George S., Chandraghatgi H., Yi P. et al:** A Simulation Study to evaluate Hospital Operating Room Performance, Internal Paper of the Department of Industrial Engineering, University of Buffalo, New York, 2002
- [GeBu08]**      **Internet-Quelle:**  
[http://www.gbe-bund.de/oowa921-install/servlet/oowa/aw92/WS0100/\\_XWD\\_PROC?\\_XWD\\_92/2/XWD\\_CUBE.DRILL/\\_XWD\\_122/D.000/3722,](http://www.gbe-bund.de/oowa921-install/servlet/oowa/aw92/WS0100/_XWD_PROC?_XWD_92/2/XWD_CUBE.DRILL/_XWD_122/D.000/3722)  
Abruf vom 18. April 2008
- [Gold89]**      **Goldberg D.E.:** Genetic algorithms in search, optimization and machine learning, Addison-Wesley Verlag, 1989
- [Gord72]**      **Gordon G.:** Systemsimulation, Oldenbourg Verlag, München 1972
- [Grei04]**      **Greiling M.:** Pfadmanagement, in: Krankenhaus Umschau KU Spezial, Nummer 24 – 5/2004, S. 21-23
- [Gülk04]**      **Gülke N.:** Prozessanalyse und Prozesssimulation – Eine anwendungsorientierte Modellentwicklung, Reihe Forschung Informationstechnik, Verlag Franzbecker, Hildesheim 2004

- [HAB04] **Hamm C.W., Arntz H.-R., Bode C., Giannitsis E., et al.:** Leitlinien Akutes Koronarsyndrom, in: Pocket Leitlinien Akutes Koronarsyndrom, Hrsg: Vorstand der Deutschen Gesellschaft für Kardiologie, Herz- und Kreislaufforschung e.V., Düsseldorf 2004
- [HAB04a] **Hamm C.W., Arntz H.-R., Bode C., Giannitsis E., et al.:** Leitlinien: Akutes Koronarsyndrom (ACS), in: Zeitschrift für Kardiologie, Band 93 Heft 1 (2004), Steinkopff Verlag 2004
- [HAB04b] **Hamm C.W., Arntz H.-R., Bode C., Giannitsis E., et al.:** Leitlinien: Akutes Koronarsyndrom (ACS), in: Zeitschrift für Kardiologie, Band 93 Heft 4 (2004), Steinkopff Verlag 2004
- [HBHS99] **Heeschen C., van den Brand M.J., Hamm C.W., Simoons M.L.:** Angiographic findings in patients with refractory unstable angina according to troponin T status, in Circulation 100 (1999), page 1509-1514, 1999
- [HGTH00] **Heeschen C., Goldmann B.U., Terres W., Hamm C.W.:** Cardiovascular risk and therapeutic benefit of coronary interventions for patients with unstable angina according to the Troponin T status, in European Heart Journal 21 (2000), page 1159-1166, 2000
- [Hess96] **Hess T.:** Entwurf betrieblicher Prozesse: Grundlagen – Methoden – neue Ansätze, Gabler, Wiesbaden 1996
- [HesJ92] **Hesser J.:** Parameteroptimierung bei genetischer Algorithmen, Dissertation, Heidelberg, 1992

- [HoDe99]** **Holmvang L., Dellborg A.K.:** Relative contribution of a single-admission 12-lead electrocardiogram and early 24-hour continuous monitoring for early risk stratification in patients with unstable coronary artery disease, in: American Journal of Cardiology 83 (1999), page 667-674
- [IDS06a]** **Internet-Quelle:**  
<http://www.ids-scheer.de/germany/49525>  
Abruf vom 10. Juni 2006
- [IDS06b]** **Internet-Quelle:**  
[http://www.ids-scheer.de/germany/products/aris\\_strategy\\_platform/49513](http://www.ids-scheer.de/germany/products/aris_strategy_platform/49513)  
Abruf vom 04. August 2006
- [IDW06]** **Internet-Quelle:**  
<http://idw-online.de/pages/de/news1011>  
Abruf vom 07. Juni 2006
- [Joc03]** **Jockwig A.:** Anpassungsschritte im Hinblick auf die Einführung der DRGs, in: Management und Krankenhaus, 22. Jahrgang, Ausgabe 11/2003, S. 9
- [KaKu05]** **Kahl T., Kupsch F.:** Transformation und Mapping von Prozessmodellen in verteilten Umgebungen mit der ereignisgesteuerten Prozesskette, Institut für Wirtschaftsinformatik im deutschen Forschungszentrum für künstliche Intelligenz, Saarbrücken, 2005
- [Kauf03]** **Kaufmann R.:** Zwei Jahre Frankfurter Klinikallianz, in: Management und Krankenhaus, 22. Jahrgang, Ausgabe 11/2003, S. 4

- [KFCH01]** Kaul P., Fu Y., Chang W.-C., Harrington R.A. et al: Prognostic Value of ST segment depression in acute coronary syndromes: insights from PARAGON-A applied to GUSTO-IIb in: Journal of American Cardiologists 38 (2001), Page 64-71
- [KFG99]** Krallmann H., Frank H., Gronau N.: Systemanalyse in Unternehmen, Vahlen Verlag, München 1999
- [KHH03]** Kong D.F., Hasselblad V., Harrington R.A., White H.D., Tcheng J.E. et al: Metaanalysis of survival with platelet glycoprotein IIb/IIIa antagonists for percutaneous coronary interventions, in American Journal of Cardiology 92 (2003), page 651-655, 2003
- [Kien02]** O.V.: Kienbaum Krankenhausstudie 2002 - Entwicklungen im Management der Krankenhäuser, Düsseldorf 2002
- [Kinn94]** Kinnebrock W.: Optimierung mit genetischen und selektiven Algorithmen, Oldenbourg Verlag, 1994
- [Klin06]** **Internet-Quelle:**  
[http://www.kliniksterben.de/friedhof/2003\\_11.html](http://www.kliniksterben.de/friedhof/2003_11.html)  
Abruf vom 02. Juni 2006
- [Klo03]** Klober K.: G-DRGs – Chancen erkennen und nutzen, in: Management und Krankenhaus, 22. Jahrgang, Ausgabe 10/2003, S. 8
- [Kneu03]** Kneuper R.: CMMI – Verbesserung von Softwareprozessen mit Capability Maturity Model Integration, dpunkt Verlag, 2003
- [Kobl06]** **Internet-Quelle:**  
<http://www.uni-koblenz.de/~moeh/projekte/emergency.html>  
Abruf vom 13. Juni 2006



- [Kral96]** **Krallmann H.:** Systemanalyse im Unternehmen: Geschäftsprozessoptimierung, partizipative Vorgehensmodelle, objektorientierte Analyse, 2. Aufl., Oldenbourg Verlag, München/Wien 1996
- [Krei04]** **Kreische D.:** Geschäftsprozessmodellierung mit der „Unified Modeling Language (UML)“, Erlangen 2004 (Dissertation)
- [Kuch04]** **Kuchinke B.A.:** Krankenhausdienstleistungen und Effizienz in Deutschland – Eine industrieökonomische Analyse, Gesundheitsökonomische Beiträge, Band 43, Nomos Verlagsgesellschaft, Baden-Baden 2004 (Dissertation)
- [Kuch05]** **Kuchinke B.A.:** Qualitätswettbewerb zwischen deutschen Akutkrankenhäusern unter besonderer Berücksichtigung von DRG und Budgets, Diskussionspapier Nr. 45, Institut für Volkswirtschaftslehre, Technische Universität Ilmenau, Juni 2005
- [Kühn06]** **Kühn M.:** Simulation und Optimierung einer Tagesklinik – Simulationsstudie mit dem MLDesigner, Diplomarbeit, Technische Universität Ilmenau, Juni 2006
- [Lang97]** **Lang K.:** Gestaltung von Geschäftsprozessen mit Referenzprozessbausteinen, Gabler, Wiesbaden 1997
- [LaKe82]** **Law A. M., Kelton D.:** Simulation modelling and analysis, New York 1982

- [LHK02]** Lagerqvist B., Husted S., Kontny F. et al.: A long-term perspective on the protective effects of an early invasive strategy in unstable coronary artery disease, in Journal of American Coll Cardiology 40 (2002), page 1902-1914, 2002
- [Miel02]** Mielke C.: Geschäftsprozesse: UML-Modellierung und Anwendungsgenerierung, Dissertation, Spektrum Verlag, Berlin/Heidelberg 1982
- [MKB01]** Magnusson K., Kroslid D., Bergman B.: Six Sigma umsetzen – Die neue Qualitätsstrategie für Unternehmen, Hanser Verlag, München/Wien 2001
- [MLD04]** O.V.: MLDesigner Documentation, Version 2.5, MLDesign Technologies Inc., Palo Alto, California, Stand: July 21, 2004
- [MoEi03]** Moos G., Eisenreich T.: Positive Perspektiven durch DRGs, in: Management und Krankenhaus, 22. Jahrgang, Ausgabe 10/2003, S. 10
- [Muen96]** Muench, E.: Hat das Krankenhaus eine Zukunft?, in: Reformen im Krankenhaus – Notwendigkeit, Möglichkeiten und Grenzen, P.C.O. Verlag, Bayreuth 1996
- [Müll73]** Müller-Merbach H.: Operations Research, 3. Aufl., Franz Vahlen Verlag, München 1973
- [NFSH99]** Negele H., Fricke E., Schrepfer L., Härtle N.: Modeling of Integrated Product Development Processes. In: Proceeding of the 9<sup>th</sup> Annual International Symposium of INCOSE Systems Engineering: Sharing The Future, Brighton, UK, 6.-11. Juni 1999
- [Niem73]** Niemeyer G.: Systemsimulation, Frankfurt am Main 1973

- [Niem77] **Niemeyer G.:** Kybernetische System- und Modelltheorie system dynamics, Franz Vahlen Verlag, München 1977
- [Niem89] **Niemeyer G.:** Betriebliche Führungssysteme – Eine system- und informationsorientierte Managementtheorie, o. A. des Verlages, 2. Auflage, Regensburg 1989
- [Niem90] **Niemeyer G.:** AMTOS Release 2.0, o. A. des Verlages, Regensburg 1990
- [Niem90a] **Niemeyer G.:** TIMER Release 2.0 August 1990, o. A. des Verlages, 1990
- [Niss97] **Nissen V.:** Einführung in evolutionäre Algorithmen – Optimierung nach dem Vorbild der Evolution, vieweg Verlag, 1997
- [Ober04] **Oberender und Partner, Unternehmensberatung für das Gesundheitswesen:** Zukünftige Herausforderungen des Gesundheitsmarktes im akutstationären und ambulanten Sektor, Vortragsfolien, 19. Januar 2004
- [OECD08] **Internet-Quelle:**  
[http://www.gbe-bund.de/gbe10/trecherche.prc\\_them\\_rech?tk=20420&tk2=20473&p\\_uid=gast&p\\_aid=43289746&p\\_sprache=D&cnt\\_ut=10&ut=20474;](http://www.gbe-bund.de/gbe10/trecherche.prc_them_rech?tk=20420&tk2=20473&p_uid=gast&p_aid=43289746&p_sprache=D&cnt_ut=10&ut=20474)  
Abruf vom 18. April 2008
- [OWG96] **Oler A., Wholey M.A., Oler J., Grady D.:** Adding heparin to aspirin reduces the incidence of myocardial infarction and death in patients with unstable angina. A meta-analysis, in: JAMA 276 (1996), pages 811-815

- [Pach02]** **Pachinger O.:** Der akute Herzinfarkt: Optimale Reperfusionstrategie 2002? In: Journal für Kardiologie, Österreichische Zeitschrift für Herz-Kreislaferkrankungen, Heft 9 (6), Seite 225-225, Verlag Krause & Pachernegg GmbH, Gablitz 2002
- [Patz82]** **Patzak G.:** Systemtechnik – Planung komplexer innovativer Systeme: Grundlagen, Methoden, Techniken, Springer Verlag, Berlin/Heidelberg 1992
- [Peer98]** **Peer, V.:** Modellbasierte Planung von Geschäftsprozessabläufen, Gabler Verlag, Wiesbaden 1998
- [Pete81]** **Peterson J. L.:** Petri net theory and the modelling of systems, New Jersey 1981
- [PMC06]** **Internet-Quelle:**  
[http://www.pmc corp.com/casestudies/case\\_study.cs088ec.htm](http://www.pmc corp.com/casestudies/case_study.cs088ec.htm)  
Abruf vom 02. Mai 2006
- [PNC01]** **Pande P., Neumann R., Cavanagh R.:** Six Sigma erfolgreich einsetzen – Marktanteile gewinnen, Produktivität steigern, Kosten reduzieren, mi verlag moderne industrie, Landsberg/Lech 2001
- [Pro06a]** **Internet-Quelle:**  
<http://www.promodel.de/produkte/medmodel/index.php>  
Abruf vom 02. Juni 2006
- [Pro06b]** **Internet-Quelle:**  
[http://www.promodel.de/produkte/process\\_simulator/index.php](http://www.promodel.de/produkte/process_simulator/index.php)  
Abruf vom 04. August 2006

- [Rauc96]** **Rauchfuß, H.:** Niedergelassener Arzt und Krankenhaus: Partner oder Konkurrenten?, in: Reformen im Krankenhaus – Notwendigkeit, Möglichkeiten und Grenzen, P.C.O. Verlag, Bayreuth 1996
- [Reis82]** **Reisig W.:** Petrinetze – Eine Einführung, Berlin 1982
- [Reis85]** **Reisig W.:** Systementwurf mit Netzen, Berlin 1985
- [ReYu03]** **Rehbehn R., Yurdakul Z.B.:** Mit Six Sigma zu Business Excellence – Strategien, Methoden, Praxisbeispiele, Publicis Corporate Publishing Verlag, Erlangen, 2003
- [RHLJ03]** **Dr. Roeder N, Prof. Hindle D, Koskamp N, Juhra C, Dr. Hensen P, Bunzemeier H, Dr. Rochell B.:** Frischer Wind mit klinischen Behandlungspfaden (I). In: das Krankenhaus, Ausgabe 1/2003, S. 20-27; Hrsg. Deutsche Krankenhausgesellschaft, Verlag W. Kohlhammer
- [Risc90]** **RISC Study:** Risk of myocardial infarction and death during treatment with low dose aspirin and intravenous heparin in men with unstable coronary artery disease, in: The RISC Group. Lancet 336 (1990), pages 827-830
- [RLH96]** **Reinhart G., Lindemann U., Heinzl J.:** Qualitätsmanagement, Springer Verlag, Berlin/Heidelberg/New York 1996
- [Roe02]** **Roeder N.:** Klinische Behandlungspfade: Erfolgreich durch Standardisierung. Bessere Zusammenarbeit, klare Verantwortlichkeiten, Kostentransparenz und mehr Qualität, f&w, 5 – 2002, S. 462-464

- [Röh03]**     **Röhrig, S.:** Prozessmanagement im Krankenhaus, in: Management und Krankenhaus, 22. Jahrgang, Ausgabe 10/2003, S. 5
- [Rum04]**     **Rumpe, R.:** Modellierung mit UML – Sprache, Konzepte und Methodik, Springer Verlag, 2004
- [SAG01]**     **Savanitto S., Ardissino D., Granger C.B. et al.:** Prognostic value of the admission electrocardiogram in acute coronary syndroms, in JAMA 281 (1999), page 707-713
- [Sali93]**     **Saliger E.:** Betriebswirtschaftliche Entscheidungstheorie: Einführung in die Logik individueller und kollektiver Entscheidungen, Oldenbourg Verlag, München, Wien 1993
- [Salz07]**     **Salzwedel H.:** Grundlagen der System- und Steuerungstheorie, Vorlesungsskript, Fakultät für Informatik und Automatisierung, Technische Universität Ilmenau, 2007
- [Salz06]**     **Salzwedel H.:** Simulation und Animation integrierter Systeme, Vorlesungsskript, Fakultät für Informatik und Automatisierung, Technische Universität Ilmenau, 2006
- [SBR05]**     **Schneider H., Buzacott J., Rücker T.:** Operative Produktionsplanung und –steuerung – Konzepte und Modelle des Informations- und Materialflusses in komplexen Fertigungssystemen, R. Oldenbourg Verlag, München/Wien 2005
- [Sche98]**     **Scheer A.-W.:** ARIS – Vom Geschäftsprozess zum Anwendungssystem, Springer Verlag, Berlin 1998

- [Sche06]** **Internet-Quelle:**  
<http://www.ids-scheer.de/germany/products/53956>  
Abruf vom 06. Juni 2006
- [Schei87]** **Schein E.H.:** Process Consultation Vol. II – Lessons for Managers and Consultants, Addison Wesley, 1987
- [Schm78]** **Schmidt B.:** Schmidt B., GPSS-Fortran – Einführung in die Simulation diskreter Systeme, in: Informatik-Fachberichte, Brauer W. (Hrsg.), Berlin 1978
- [Schm85]** **Schmidt B.:** Systemanalyse und Modellaufbau, in: Fachberichte Simulation, Möller D. / Schmidt B. (Hrsg.), Berlin 1985
- [SHF94]** **Schöneberg E., Heinzmann F., Feddersen S.:** Genetische Algorithmen und Evolutionsstrategien – Eine Einführung in Theorie und Praxis der simulierten Evolution, Addison-Wessley Verlag, 1994
- [Schü98]** **Schütte R.:** Grundsätze ordnungsgemäßer Referenzmodellierung – Konstruktion konfigurations- und anpassungsorientierter Modelle, Gabler Verlag, Wiesbaden 1998
- [Selz03]** **Selzer, G.:** Geschäftsprozesse, Shaker Verlag, Aachen 2003
- [SiDe08]** **Sivanandam S.N., Deepa S.N.:** Introduction to genetic algorithms, Springer Verlag, 2008
- [SiFo04]** **Siviy J., Forrester E.:** Accelerating CMMI Adoption Using Six Sigma, Präsentation der Carnegie Mellon University, Pttsburgh, 2004

- [Sine03]**     **Sinem K.:** Ereignisgesteuerte Prozessketten (EPKn), Abstract der Universität Ulm, Lehrstuhl für Informatik, 2003
- [Spie06]**     **Internet-Quelle:**  
<http://www.spiegel.de/wirtschaft/0,1518,414096,00.html>  
Abruf vom 03. Juni 2006
- [SSK90]**     **Schoderbek P., Schoderbek C., Kefalas A.:** Management Systems - Conceptual Considerations, 4. Ausgabe, Boston 1990
- [StaB08]**     **Internet-Quelle:**  
<http://www.destatis.de/jetspeed/portal/cms/Sites/destatis/Internet/DE/Content/Statistiken/Gesundheit/Gesundheitsausgaben/Tabellen/Content50/Einrichtungen,templateId=renderPrint.psml>,  
Abruf vom 18. April 2008
- [Stac73]**     **Stachowiak H.:** Allgemeine Modelltheorie, Springer, Wien 1973
- [STB99]**     **Sepúlveda J., Thompson W., Baesler F. et al:** The use of simulation for process improvement in a cancer treatment center, in: Proceeding of the 1999 Winter Simulation Conference, 1999
- [Ste91]**     **Steinmüller W.:** Information, Modell, Informationssystem, Forschungsbericht Nr. 5/1991 des Fachbereichs Mathematik und Informatik der Universität Bremen, Bremen 1991
- [Stew87]**     **Stewart W.:** The genetic algorithm and biological development, in: Genetic Algorithms And Their Applications, Proceedings of the Second International Conference on Genetic Algorithms, MIT Cambridge / MA, 1987



- [TOM88]** **Thérroux P., Ouimet H., McCans J. et al.:** Aspirin, heparin or both to treat acute unstable angina, in: New England Journal of Medicine 319 (1988), pages 1105 – 1111
- [TWM93]** **Thérroux P., Waters D., Qui S., McCans J. et al.:** Aspirin versus heparin to prevent myocardial infarction during the acute phase of instable angina, in Circulation 88 (1993), pages 2045-2048
- [UniB06]** **Internet-Quelle:**  
[http://dwarf.umi.cs.tu-bs.de/full/research/mis/mosaikm/mosaikm\\_kurzdarstellung.html](http://dwarf.umi.cs.tu-bs.de/full/research/mis/mosaikm/mosaikm_kurzdarstellung.html)  
Abruf vom 02. Mai 2006
- [Unik06]** **Internet-Quelle:**  
[www.medizin.uni-koeln.de/kai/igmg/studien/telemedizin/telmed01.htm](http://www.medizin.uni-koeln.de/kai/igmg/studien/telemedizin/telmed01.htm)  
Abruf vom 06. Juni 2006
- [VoFi03]** **Voß S., Fink A.:** Modellierung von Informationssystemen: Einführung II, Vorlesungsskript, Universität Hamburg, Institut für Wirtschaftsinformatik, 2003
- [Wag04]** **Wagner S.:** Looking inside genetic algorithms, in: Reihe Technik und Naturwissenschaften Nr. 43, Trauner Verlag, 2004
- [WeHo99]** **Wenig M., Houshmand A.:** Health Care Simulation: A Case Study at a Local Clinic, in: Proceedings in the 1999 Winter Simulation Conference, 1999
- [Wil05]** **Wilson D.:** CMMI and Six Sigma Synergy, Präsentation der Firma Natural SPI, Partner der SEI, 2005

- [Wuer06]** **Internet-Quelle:**  
<http://ki.informatik.uni-wuerzburg.de/~vki/SPP.htm>  
Abruf vom 02. Mai 2006
- [YZM01]** **Yusuf S., Zaho F., Mehta S.R., Chrolavicius S. Togononi G., Fox K.K.:** Effects of clopidogrel in addition to aspirin in patients with acute coronary syndromes without ST-segment elevation, in: New England Journal of Medicine 345 (2001), pages 494-502
- [Zwic81]** **Zwicker E.:** Simulation und Analyse dynamischer Systeme in den Wirtschafts- und Sozialwissenschaften, deGruyter, Berlin/New York 1981

## Anhang

### Anhang 1: Abbildungen und Tabellen

#### Fragebogen zur Ermittlung des Prozessreifegrades in Kliniken

Öffentliche Klinik       Private Klinik

Bettenanzahl:       >200       >500       >1000

Welche Position haben Sie in Ihrem Krankenhaus?

- Arzt
- Oberarzt
- Chefarzt
- Ärztlicher Direktor
- .....

#### Wie beurteilen Sie die qualitative Prozessbeschreibung / -dokumentation in Ihrer Klinik?

Gibt es dokumentierte Prozessbeschreibungen?       Ja       Nein

Wie umfassend sind die vorhandenen Prozessbeschreibungen?

- Rudimentär
- Teils beschrieben / dokumentiert
- Beschreibung/Dokumentation ca. zur Hälfte fertig gestellt
- Überwiegend beschrieben / dokumentiert
- Umfassend / vollständig dokumentiert

Sind die Prozesse klar definiert?

- Nein, überhaupt nicht.
- Teilweise
- Mittelmäßig
- Gut
- Ja, sehr gut.

**Wie beurteilen Sie die qualitative Prozessbeschreibung / -dokumentation in Ihrer Klinik?** (Fortsetzung)

Decken die Prozessbeschreibungen unterschiedliche Prozesslevel ab?  Ja  Nein

Wenn ja, bis zu welchem Prozesslevel?

- Level 1 (Hauptprozesse)
- Level 2
- Level 3
- Level 4
- Level 5

Wurden die Prozessbeschreibungen im Unternehmen ausreichend kommuniziert?

- Nein, überhaupt nicht.
- Ansatzweise kommuniziert
- Teilweise kommuniziert
- Überwiegend kommuniziert
- Umfassend / flächendeckend kommuniziert

**Ist ein Prozessverantwortlicher benannt?**

- Kein Prozessverantwortlicher benannt
- Rolle des Prozessverantwortlichen definiert, verantwortliche Person noch nicht benannt
- Prozessverantwortlicher benannt, Verantwortungsbereich noch nicht konkretisiert/festgelegt
- Prozessverantwortlicher benannt, Verantwortungsbereich festgelegt
- Prozessverantwortlicher benannt und aktiv, Verantwortungsbereich festgelegt und kommuniziert

**Sind Prozesskennzahlen zur Messung der Prozessperformance definiert worden?**

- Keine Prozesskennzahlen definiert
- Prozesskennzahlen nur ansatzweise definiert
- Prozesskennzahlen teilweise definiert
- Prozesskennzahlen überwiegend definiert
- Prozesskennzahlen vollständig/umfassend definiert

**Werden definierte Prozesskennzahlen kontinuierlich gemessen?**

- Prozesskennzahlen wurden noch nicht gemessen
- Prozesskennzahlen wurden nur einmalig gemessen
- Prozesskennzahlen werden nur selten gemessen
- Prozesskennzahlen werden in unregelmäßigen Abständen / nicht ausreichend oft gemessen
- Prozesskennzahlen werden regelmäßig und ausreichend oft gemessen

**Werden Verbesserungsmaßnahmen - ausgehend von den Prozesskennzahlen - abgeleitet?**

- Bisher wurden keine Verbesserungsmaßnahmen abgeleitet
- Bisher wurden nur einmal Verbesserungsmaßnahmen abgeleitet
- Bisher wurden unregelmäßig und eher selten Verbesserungsmaßnahmen abgeleitet
- Bisher wurden schon mehrmals aber unregelmäßig Verbesserungsmaßnahmen abgeleitet
- Es werden regelmäßig Verbesserungsmaßnahmen abgeleitet

**Wie beurteilen Sie die Prozessmanagement-Erfahrung in Ihrer Klinik?**

- Prozessmanagement-Erfahrung ist nicht vorhanden
- Prozessmanagement-Erfahrung ist nur rudimentär vorhanden
- Prozessmanagement-Erfahrung ist teilweise vorhanden
- Prozessmanagement-Erfahrung ist überwiegend vorhanden, aber noch nicht in ausreichendem Umfang
- Prozessmanagement-Erfahrung ist ausreichend vorhanden

**Werden aus Ihrer Sicht bestehende Prozesse eingehalten / genutzt?**

- Prozesse werden nicht genutzt/"gelebt"
- Prozesse werden nur rudimentär genutzt/"gelebt"
- Prozesse werden teilweise genutzt/"gelebt"
- Prozesse werden gut genutzt/"gelebt"
- Prozesse werden vollständig/umfassend genutzt/"gelebt"

**Wie schätzen Sie die abteilungsübergreifende Prozessintegration ein?**

- Völlig unzureichend
- Mangelhaft
- Befriedigend
- Gut
- Sehr gut

**Sind Ihre vorhandenen Prozesse ausreichend durch IT-Verfahren unterstützt?**

- Prozesse werden nicht durch IT-Verfahren unterstützt
- Prozesse werden durch IT-Verfahren unterstützt, jedoch in unzureichender Art und Weise
- Prozesse werden nur teilweise durch adäquate IT-Verfahren unterstützt
- Prozesse werden überwiegend durch adäquate IT-Verfahren unterstützt
- Prozesse werden flächendeckend durch adäquate IT-Verfahren effizient unterstützt

**Wie hoch schätzen Sie das Verbesserungspotential der vorhandenen Prozesse?**

- Sehr hoch
- Hoch
- Mittel
- Gering
- Sehr gering

**Glauben Sie, dass das Thema Prozessoptimierung ein ausschlaggebender Faktor für die langfristige Wettbewerbspositionierung Ihrer Klinik ist?**

- Ja, in jedem Fall
- Ja, aber nur bedingt
- Nein

Abbildung 160: Fragebogen zum Prozessreifegrad

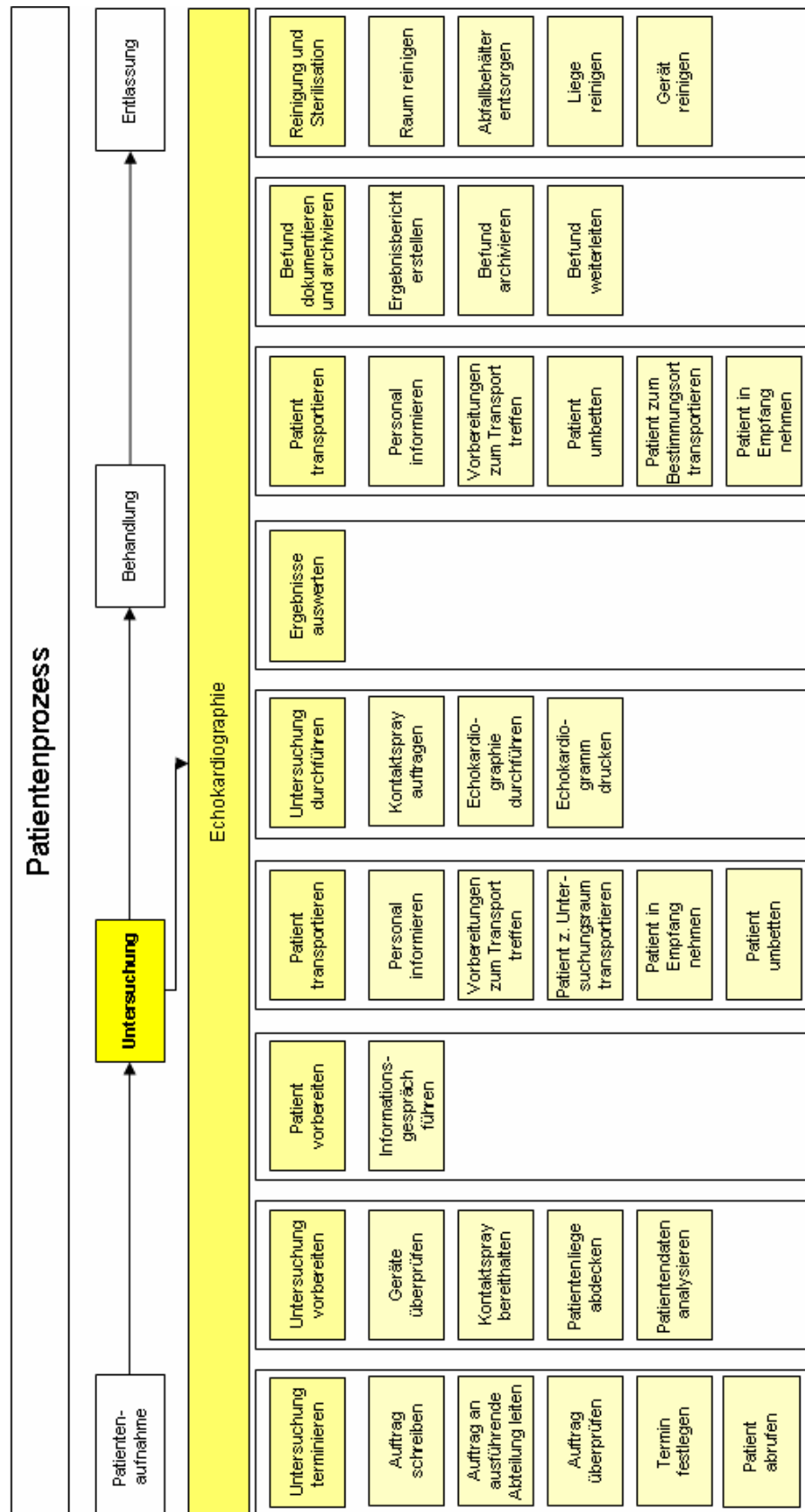


Abbildung 161: Prozessbeschreibung der Echokardiographie

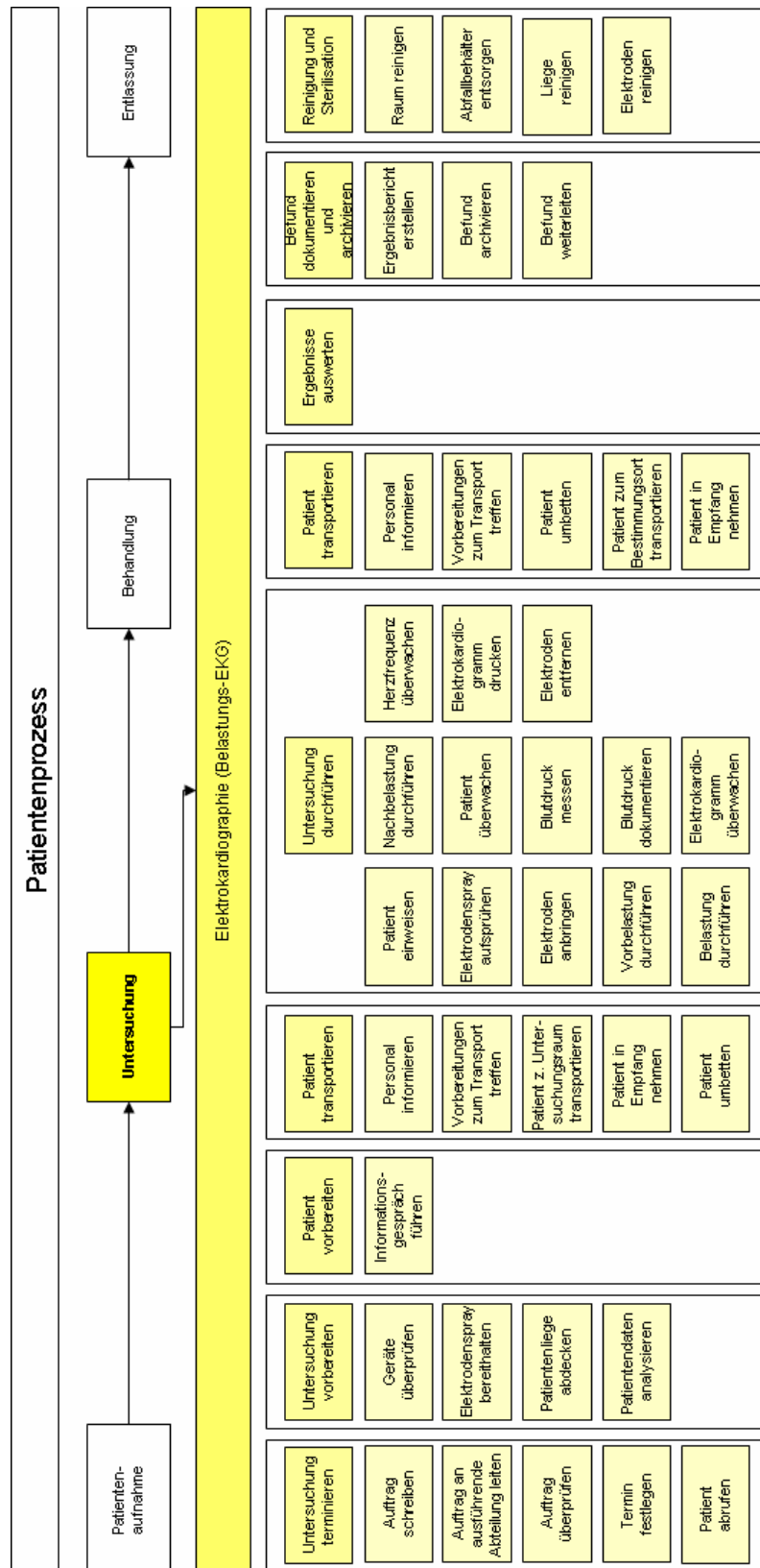


Abbildung 162: Prozessbeschreibung des Belastungs-EKGs



<b>In Kraft getreten am:</b>	<b>Gesetz:</b>
1.1.1985	Gesetz zur Neuordnung der Krankenhausfinanzierung
1.1.1989	Gesundheits-Reformgesetz (GRG)
1.1.1993	Gesundheits-Strukturgesetz (GSG)
1.1.2000	Gesetz zur Reform der gesetzlichen Krankenversicherung ab dem Jahr 2000 (GKV – Gesundheitsreform 2000)

Tabelle 1: Auswahl an wichtigen Novellierungen im Gesundheitswesen [Kuch04 S.17]

§135 a SGB V	§135 a SGB V schreibt vor, dass Vertragsärzte, Krankenhäuser und stationäre Vorsorge- und Rehabilitationseinrichtungen zur Einführung und Weiterentwicklung eines einrichtungsinternen Qualitätsmanagements verpflichtet sind. Sie sind verpflichtet, an einrichtungsübergreifenden Maßnahmen der Qualitätssicherung teilzunehmen. Diese haben insbesondere zum Ziel, die Ergebnisqualität der Anbieter von Gesundheitsdienstleistungen zu verbessern.
§95 SGB V	Nach §95 SGB V können Leistungserbringer, die an der medizinischen Versorgung von GKV-Versicherten teilnehmen, sog. medizinische Versorgungszentren gründen. Medizinische Versorgungszentren sind fachübergreifende Einrichtungen, in denen Ärzte, die in das Arztregister eingetragen sind, als Angestellte oder Vertragsärzte tätig sind.
§116b SGB V	Nach §116b SGB V können Krankenhäuser im Rahmen von Disease-Management-Programmen (DMP) ambulante Behandlungen, deren Erbringung aufgrund besonderer Leistungs- und Qualitätsanforderungen in einem DMP vorgesehen sind, vornehmen. Die Krankenkassen, die Landesverbände der Krankenkassen oder die Verbände der Ersatzkassen können direkt mit zugelassenen Krankenhäusern Verträge über die ambulante Erbringung hoch spezialisierter Leistungen sowie zur Behandlung seltener Erkrankungen und Erkrankungen mit besonderen Krankheitsverläufen schließen.
§§140a bis d SGB V	Die §§140a bis d SGB V schaffen die gesetzliche Grundlage für eine integrierte Versorgung. Gegenstand ist eine verschiedene Leistungssektoren übergreifende Versorgung der Versicherten oder eine interdisziplinär-fachübergreifende Versorgung. Die Teilnahme der Versicherten an der integrierten Versorgung ist allerdings freigestellt.

Tabelle 2: Änderungen im Sozialgesetzbuch – Fünftes Buch (SGB V)

Name des Bearbeiters: .....							Zeitpunkt des Beginns der Untersuchung durch den behandelnden Arzt	Ende der Untersuchung durch den behandelnden Arzt	Zeitpunkt des Abrufs der Therapie (Zytostatika) in der Apotheke	Zeitpunkt des Eintreffens der Therapie (Zytostatika) in der Tagesklinik	Zeitpunkt des Beginns der Prämedika- tion/Vorlauf am heutigen Behandlungs- tag	Ende der Prämedikation Vorlauf am heutigen Behandlungs- tag	Zeitpunkt des Beginns der Chemotherapie am heutigen Behandlungs- tag	Ende der Chemotherapie (inkl. aller Infusionen) am heutigen Behandlungs- tag	Entlassung (Uhrzeit)
Datum	Zyklusstag	Ankunftszeit im Empfangs- bereich	Beginn (Uhrzeit) der Aufnahme	Ende (Uhrzeit) der Aufnahme	Beginn (Uhrzeit) der Blutabnahme	Ende (Uhrzeit) der Blutabnahme									
<Name Patient>															
<Name Patient>															
<Name Patient>															
<Name Patient>															
<Name Patient>															
<Name Patient>															
<Name Patient>															
<Name Patient>															

Tabelle 8: Musterdatenblatt zur Datenerhebung

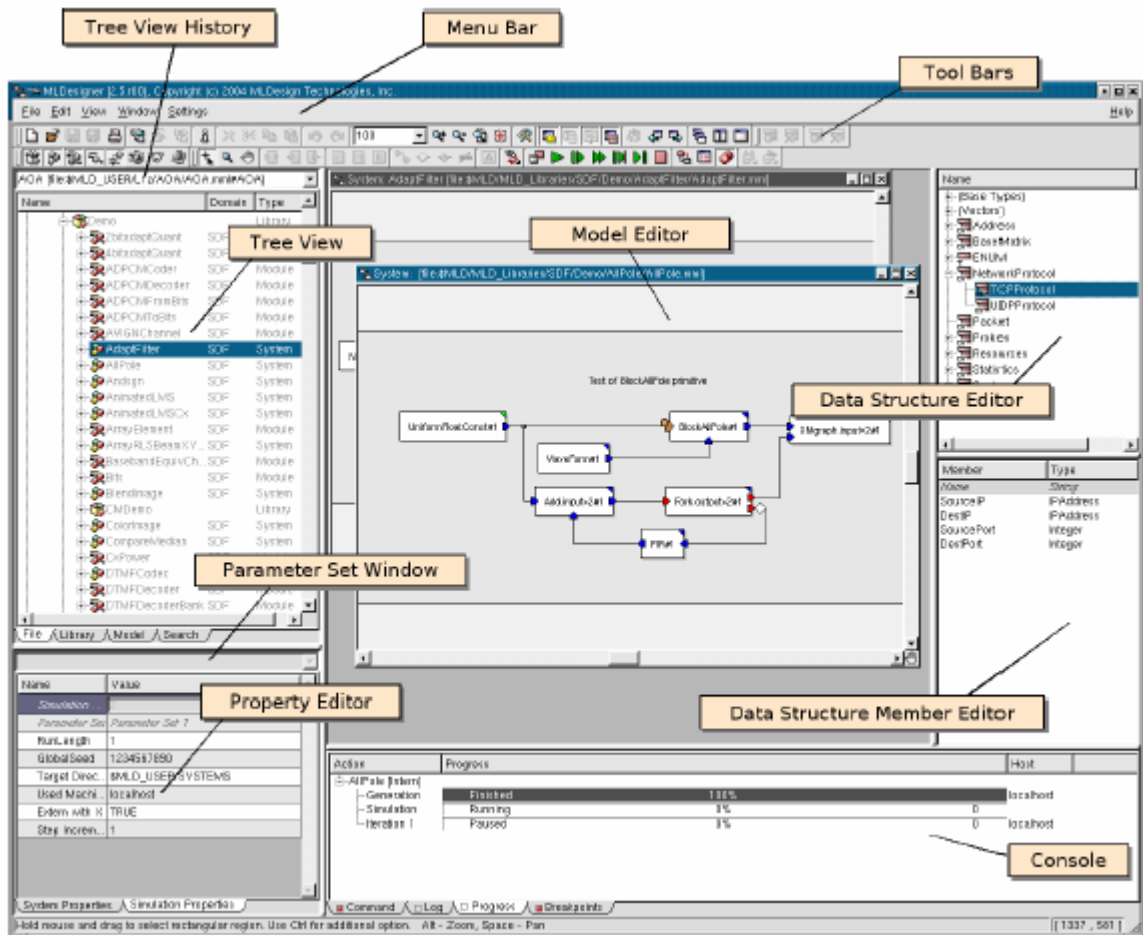


Abbildung 163: Graphical User Interface (GUI) des MLDesigners

		Ankunft - Blutabnahme	Dauer Blutabnahme	Ende Blutabnahme - Beginn Untersuchung	Dauer Untersuchung	Ende Untersuchung - Beginn Chemo	Dauer Chemo	Ende Chemo - Ende Behandlung	Behandlungsdauer (Ges)	Wartezeit (Gesamt)	Durchlaufzeit (Gesamt)
Werte Simulation	Durchschnitt	10	12	73	19	65	143	47	174	194	369
	Min	3	7	23	9	56	131	18	147	100	247
	Max	25	18	127	28	75	160	340	206	567	773
Werte Realsystem	Durchschnitt	22	12	67	18	83	145	34	175	206	381
	Min	5	4	1	3	5	5	5	12	16	28
	Max	75	32	270	60	215	325	225	417	785	1.202

Tabelle 9: Tagesgenauer Vergleich der Ergebnisse der Simulation mit den Werten des Realsystems (Werte in Minuten)

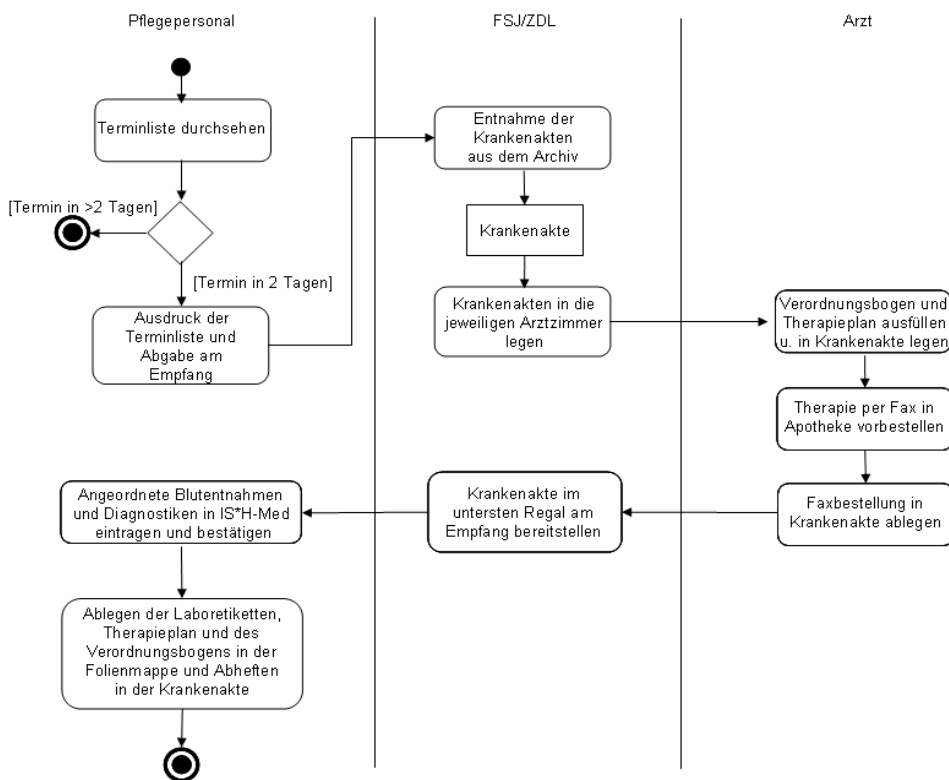


Abbildung 164: Prozessbeschreibung der Besuchsvorbereitung (Ist-Prozess)

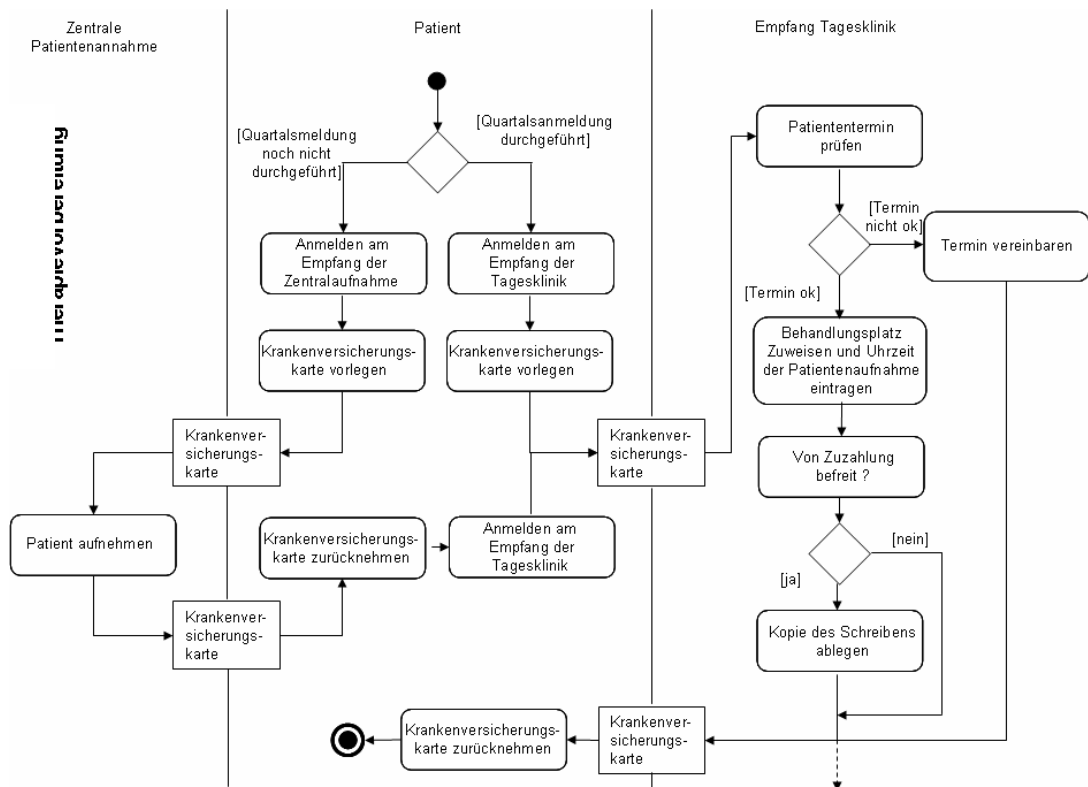


Abbildung 165: Prozessbeschreibung der Patientenaufnahme (Ist-Prozess)

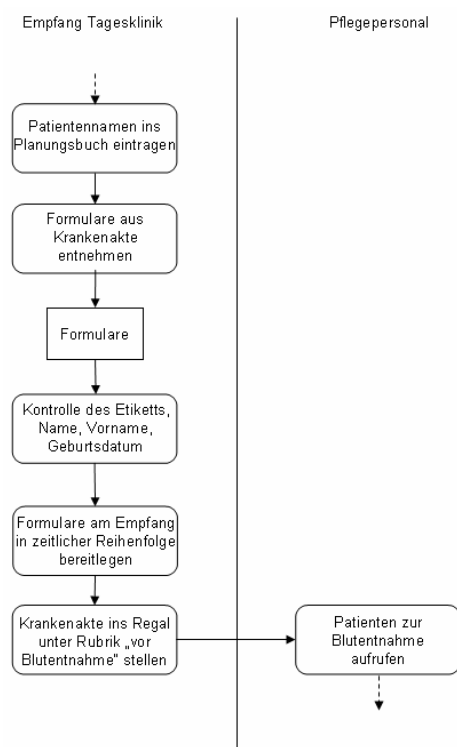


Abbildung 166: Prozessbeschreibung Patientenaufnahme und Blutentnahme (Ist-Prozess)

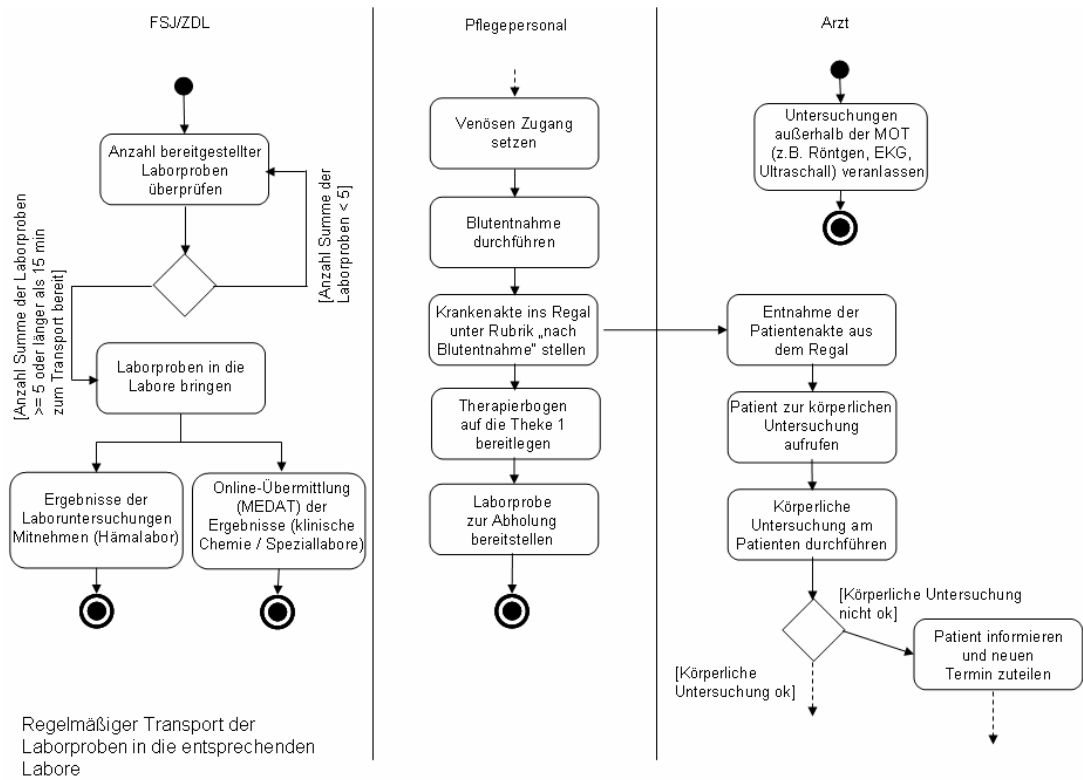


Abbildung 167: Prozessbeschreibung der Blutentnahme und Laboruntersuchung sowie ärztliche Untersuchung (Ist-Prozess)

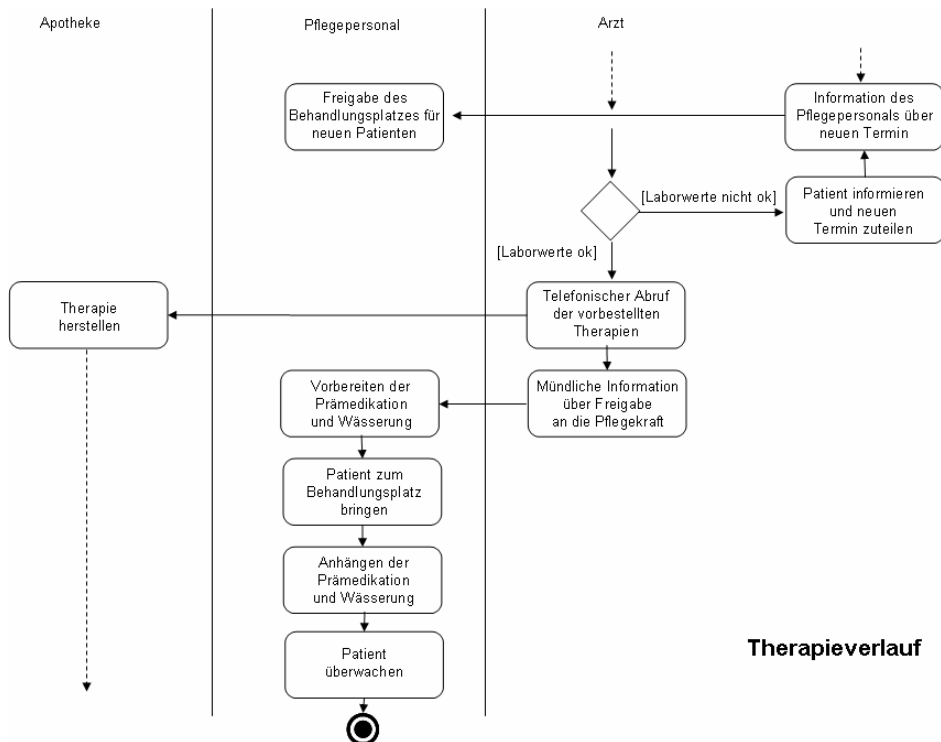


Abbildung 168: Prozessbeschreibung der Therapievorbereitung (Ist-Prozess)

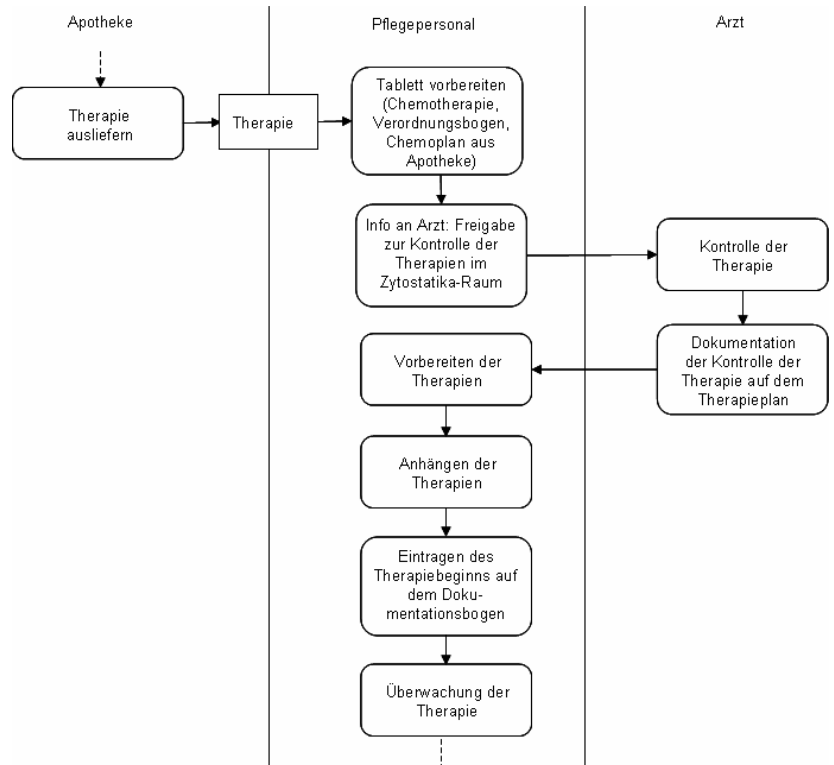


Abbildung 169: Prozessbeschreibung Therapiebeginn und Therapieüberwachung (Ist-Prozess)

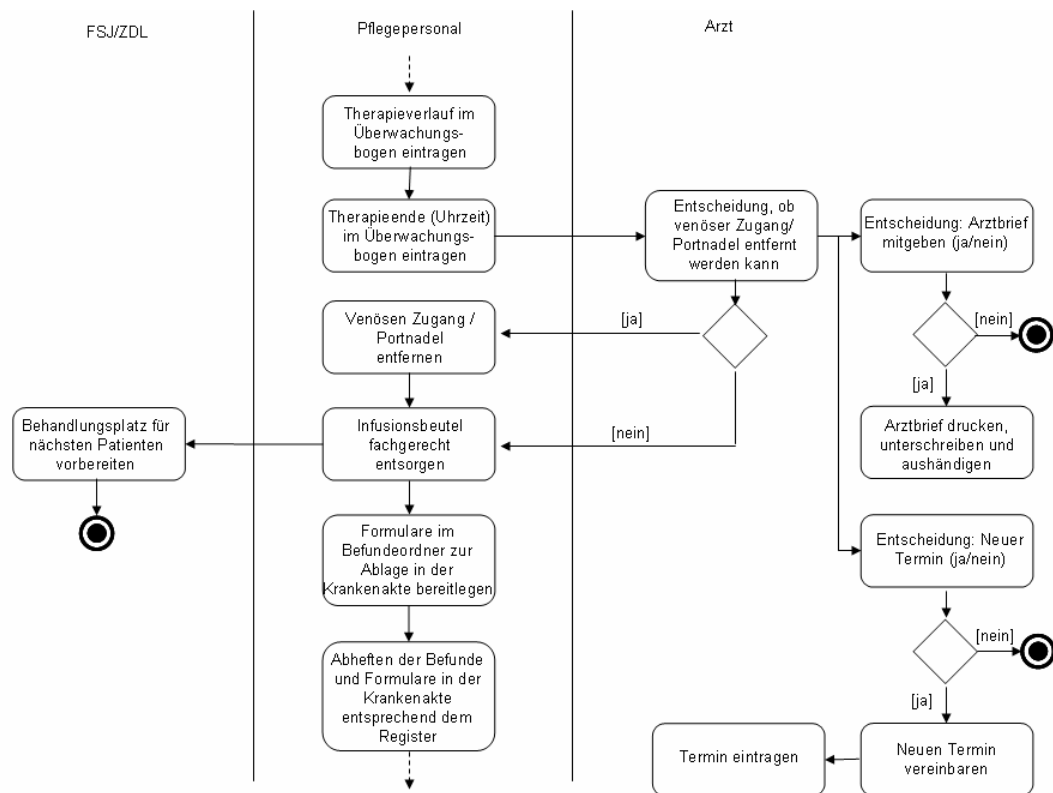


Abbildung 170: Prozessbeschreibung Therapie-Ende und Entlassung (Ist-Prozess)

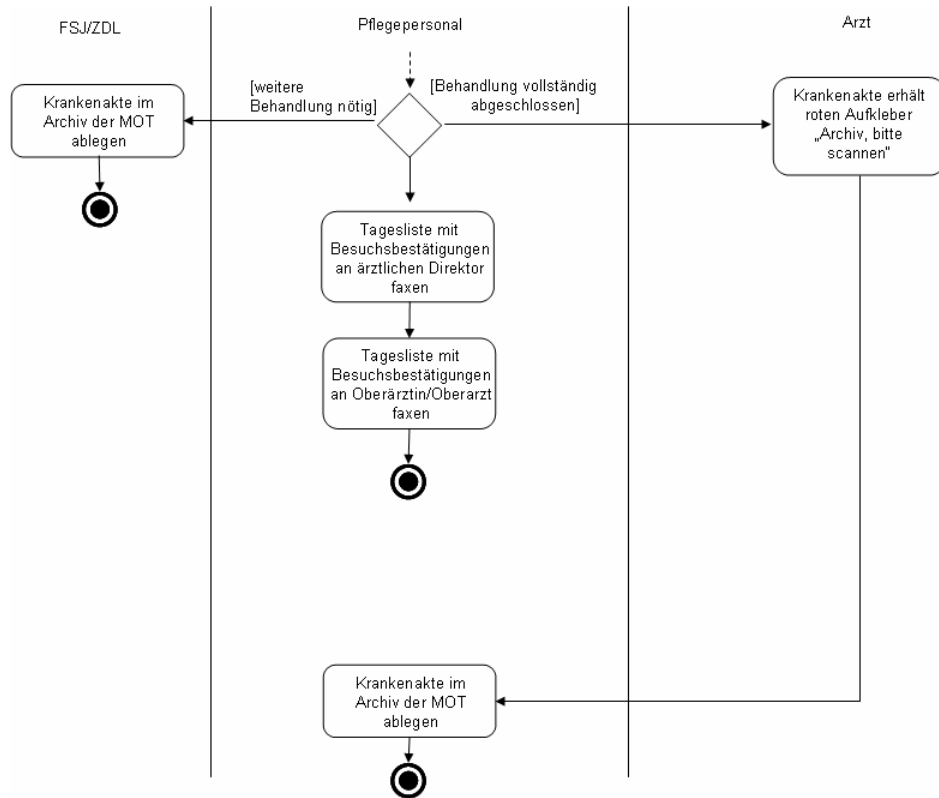


Abbildung 171: Prozessbeschreibung Nachbearbeitung (Ist-Prozess)

		Wartezeit des Patienten bis Beginn Blutabnahme	Dauer der Blutabnahme	Wartezeit zwischen Ende Blutentnahme und Beginn der ärztlichen Untersuchung	Dauer der ärztlichen Untersuchung	Wartezeit zwischen Ende ärztlicher Untersuchung und Beginn der Chemotherapie	Dauer der Therapie	Ende der Therapie bis Ende der Behandlung
<b>Im Simulationsmodell ermittelte Werte</b>	Durchschnitt	17	11	90	17	78	139	33
	Min.	3	7	29	12	60	113	26
	Max.	31	17	191	26	104	164	45
<b>Ist-Werte der Datenaufnahme</b>	Durchschnitt	22	12	80	18	83	145	34
	Min.	5	5	22	9	46	102	18
	Max.	45	24	210	28	124	177	62

Anmerkung: Zeiten in Minuten

Tabelle 10: Vergleich der Ist-Werte mit den Werten des Simulationsmodells (Beispiel eines ausgewählten Arbeitstages)

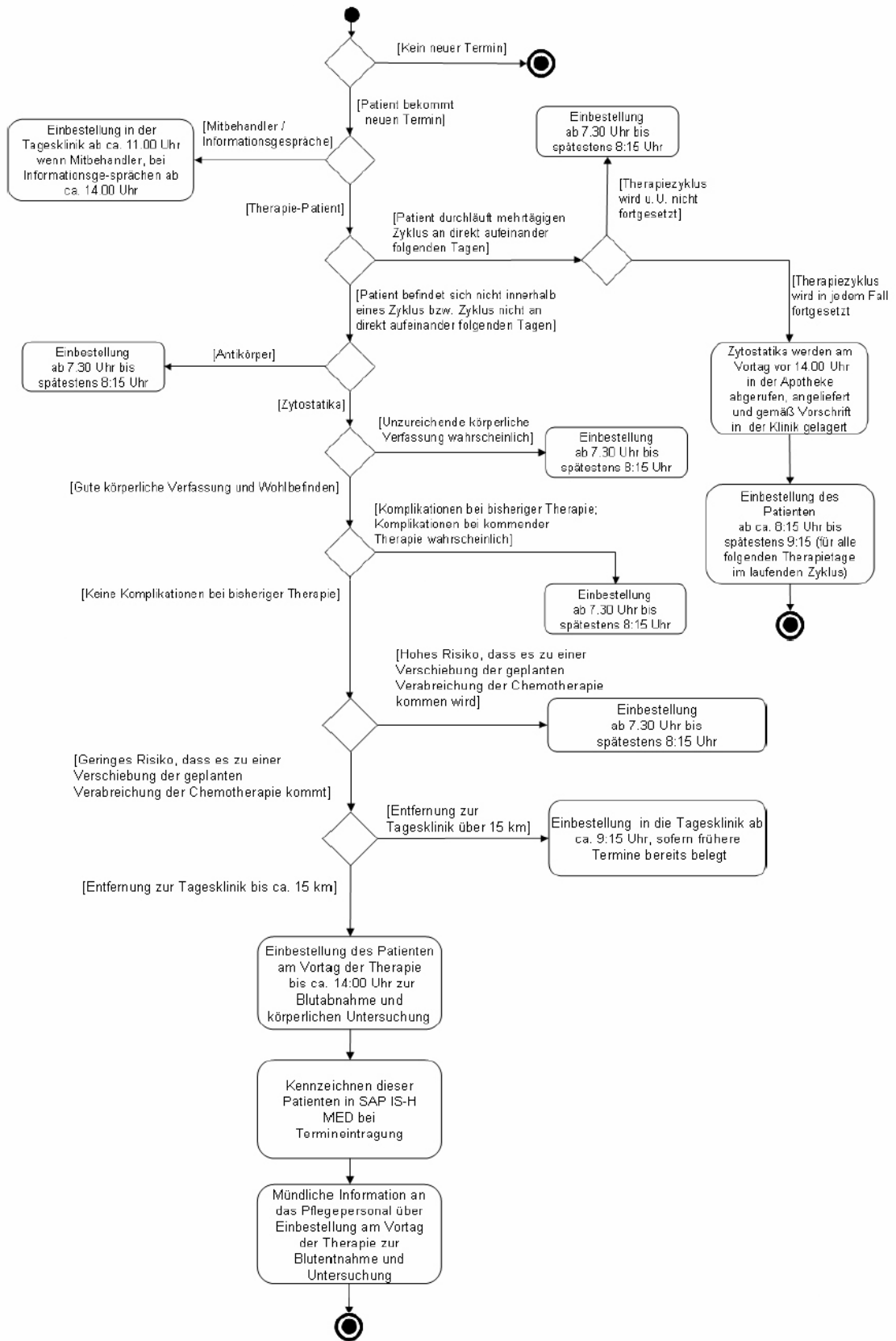


Abbildung 172: Prozessbeschreibung der optimierten Patienteneinbestellung (Soll-Prozess)



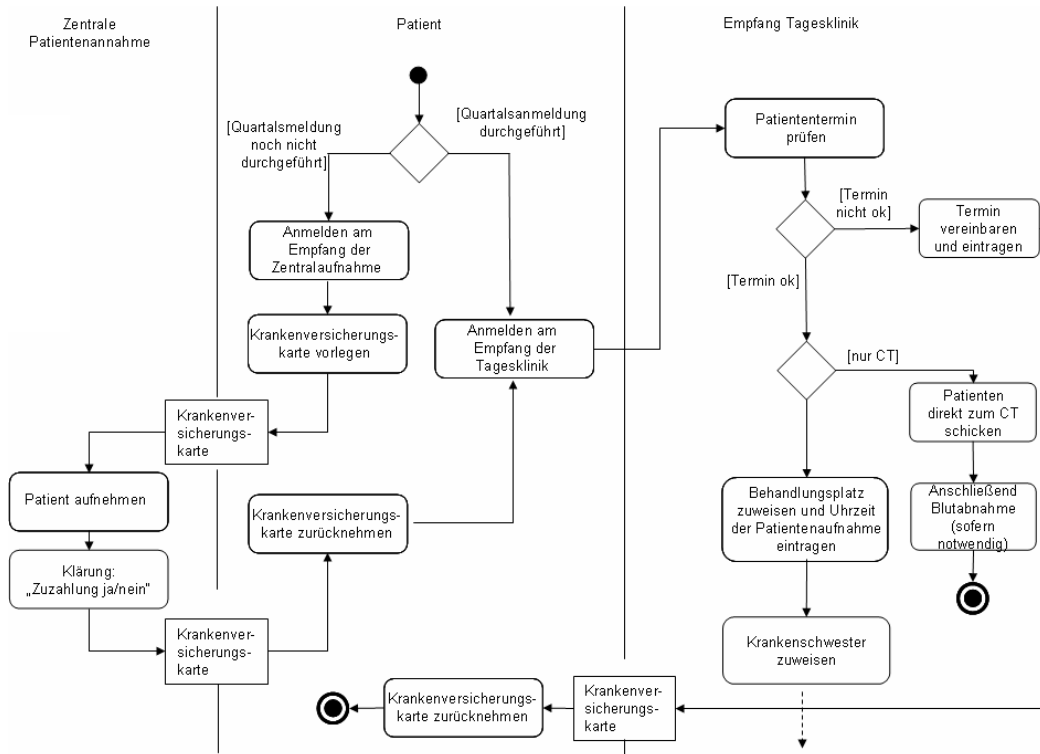


Abbildung 173a: Prozessbeschreibung der Patientenaufnahme (Soll-Prozess)

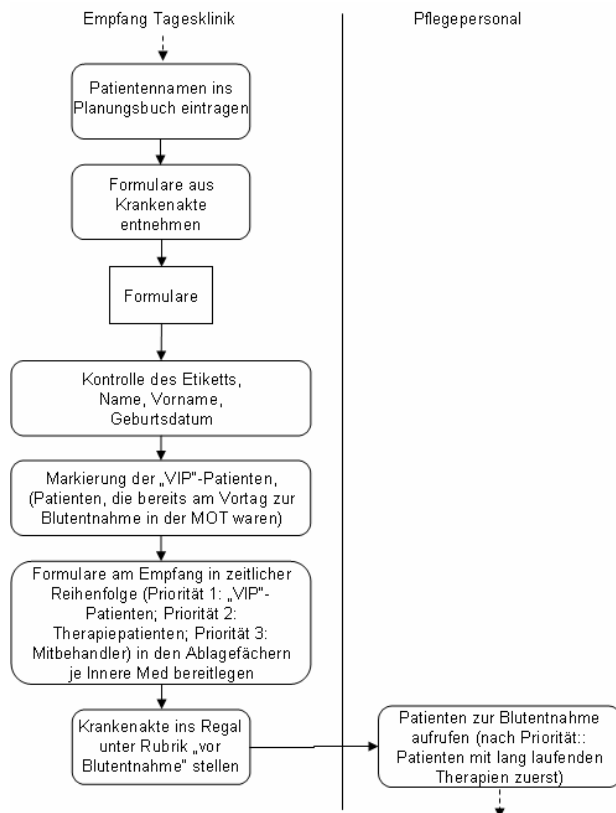


Abbildung 173b: Prozessbeschreibung der Patientenaufnahme (Soll-Prozess)

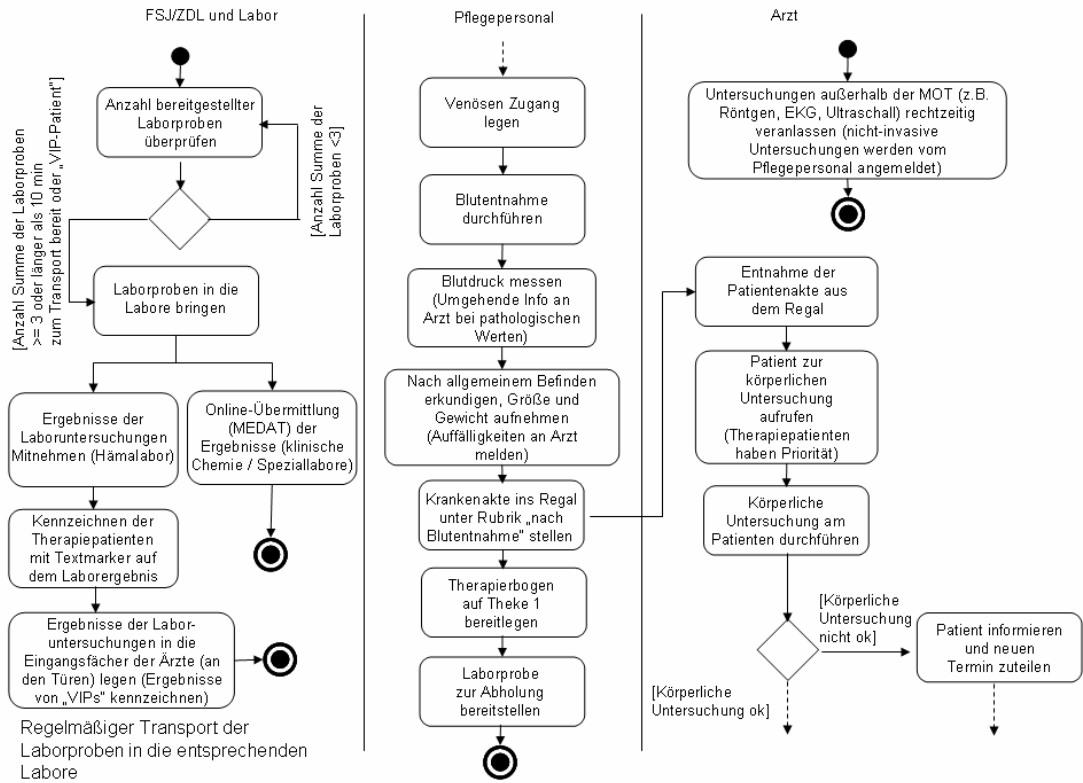


Abbildung 174: Prozessbeschreibung der Blutentnahme und Laboruntersuchung (Soll-Prozess)

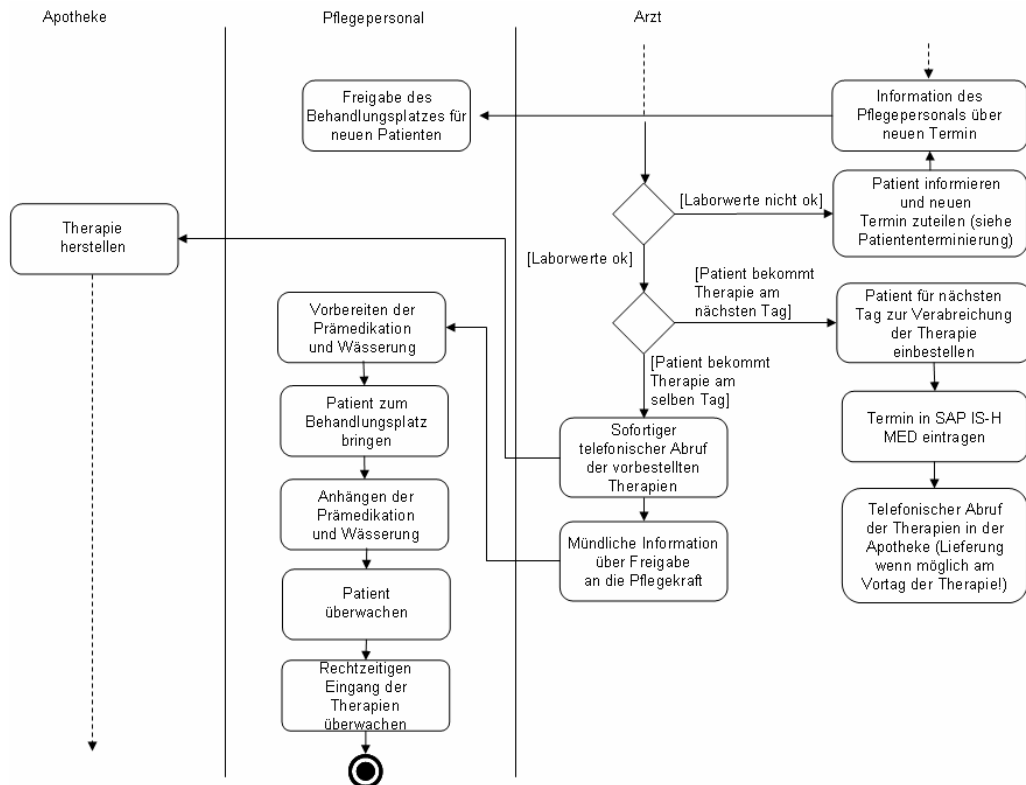


Abbildung 175: Prozessbeschreibung Therapievorbereitung (Soll-Prozess)

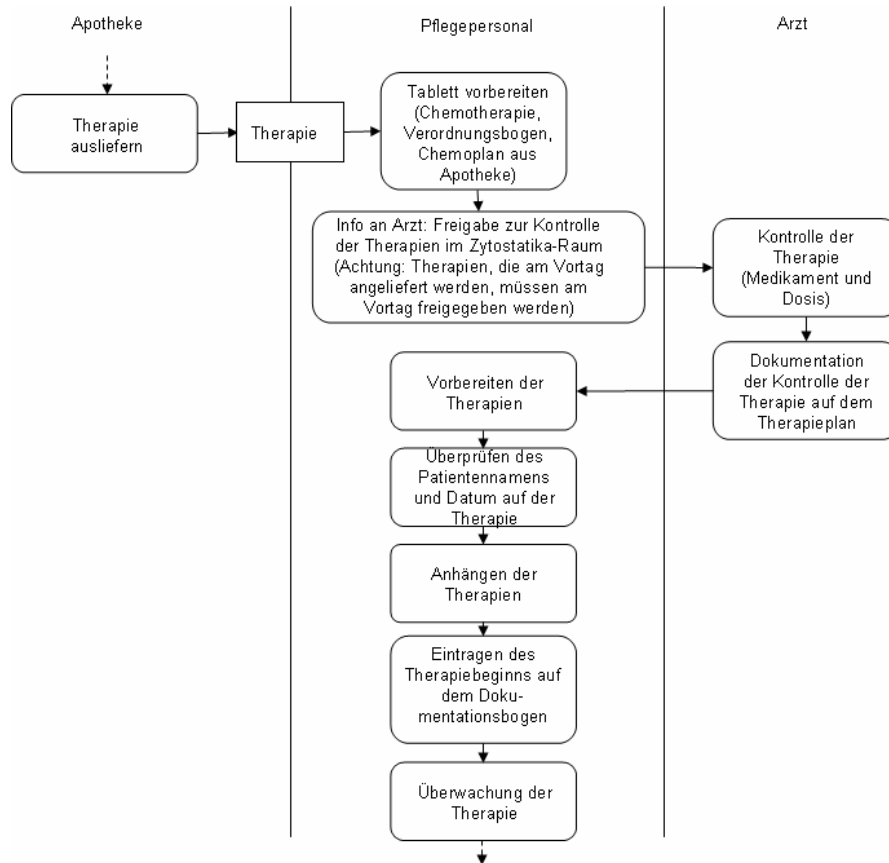


Abbildung 176a: Prozessbeschreibung Therapiedurchführung und -überwachung (Soll-Prozess)

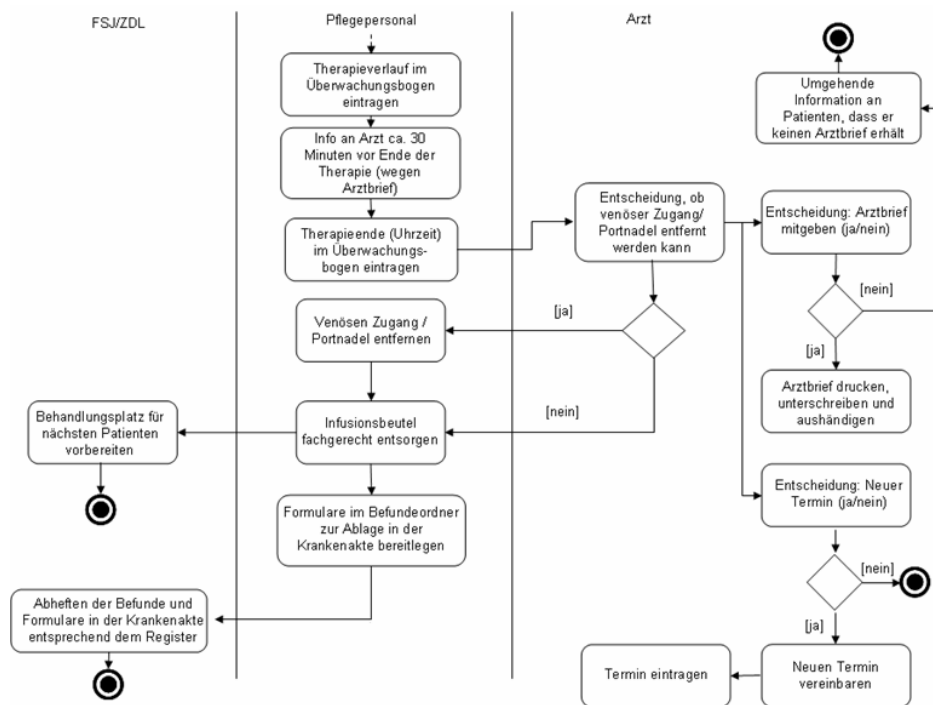


Abbildung 176b: Prozessbeschreibung Therapiedurchführung und -überwachung (Soll-Prozess)

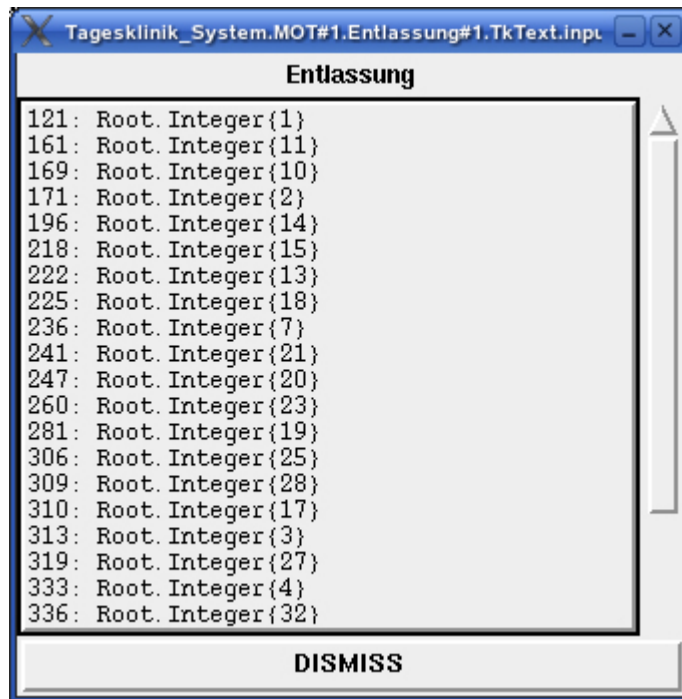


Abbildung 177: Ende-Zeitpunkt der jeweiligen Transaktion (in Minuten) und zugehörige Patienten-ID

## Anhang 2: Beschreibung der Modellkomponenten<sup>243</sup>

### Modul-Name: Empfangsbereich

#### Kurzbeschreibung:

Dieses Modul repräsentiert den Empfangsbereich, in dem die Transaktionen erzeugt werden. Notwendige Patienteninformationen können aus einer Textdatei eingelesen oder alternativ automatisch generiert werden (u.a. die Patienten-ID und die Ankunftszeit des Patienten). Jeder Patient wird in Form eines Vektors repräsentiert, dem die entsprechenden eingelesenen Datenfelder zugewiesen sind. Der Vektor wird im Memory „Patiententabelle“ abgelegt. Die erzeugten Transaktionen werden nunmehr nur mit einer ID gekennzeichnet.

#### Input:

WS_Demand_Aufnahme	Beschreibung:	Abruf der nächsten Transaktion aus der vorgelagerten Warteschlange.
	Typ:	Scalar anytype
WS_ID_in_Aufnahme	Beschreibung:	Nicht bearbeitete Transaktionen werden über diesen Port empfangen und in einer Warteschlange gepuffert.
	Typ:	Scalar int

#### Output:

ID_out_Aufnahme	Beschreibung:	Weitergabe der generierten Transaktionen an das nachgelagerte Modul „Patientenaufnahme“.
	Typ:	Scalar int

---

<sup>243</sup> Module sind mit dem Symbol  gekennzeichnet, Primitive mit dem Symbol .















Parameter:

AbstandPatienten	Beschreibung:	Mittlere Ankunftszeit der über eine Poissonverteilung generierten Transaktionen.
	Typ:	Float
	Wert:	0
	Bereich:	External
AnzahlPatienten	Beschreibung:	Anzahl zu generierender Patienten
	Typ:	Int
	Wert:	0
	Bereich:	External
FileName	Beschreibung:	Name der einzulesenden Datei, welche die Patienteninformationen enthält.
	Typ:	Filename
	Wert:	0
	Bereich:	External

Memory:

PatientenTabelle	Beschreibung:	Speichert Werte auf Basis der hinterlegten Datenstruktur.
	Typ:	Root.Vector.PatientenTabelle
	Bereich	External

Instanzen:

2InOutSynchronize#1	
ChangeLengthVector#1	
FIFOQueue#1	
LengthOfVector#1	
MemoryRead#1	
MemoryWrite#1	
Merge.input=2#1	
FileSource#1	
PatientenQuelle_Generator#1	
PlusConstInt#1	
Schreibe_EZP#1	
SelectFieldDS#1	
SetElementVector#1	
Untersuchungsablauf_generieren#1	

## Modul-Name: Patientenaufnahme

### Kurzbeschreibung:

Dieses Modul repräsentiert die Patientenaufnahme und beinhaltet die drei Kategorien Notaufnahme, stationäre Aufnahme und ambulante Aufnahme.

#### Input:

ID_in_Aufnahme	Beschreibung	Zu bearbeitende Transaktionen werden über diesen Port empfangen.
	Typ:	Scalar int

#### Output:

ID_out_Untersuchung	Beschreibung	Weitergabe der fertig bearbeiteten Transaktionen an das nachgelagerte Modul „Untersuchung_Hauptmodul“.
	Typ:	Scalar int
WS_Demand_Aufnahme	Beschreibung	Abruf der nächsten zu bearbeitenden Transaktion aus der vorgelagerten Warteschlange.
	Typ:	Scalar anytype
WS_ID_out_Aufnahme	Beschreibung	Abgewiesene Transaktionen werden an die vorgelagerte Warteschlange geleitet.
	Typ:	Scalar int

#### Parameter:

maxDauer	Beschreibung:	Maximale Bearbeitungsdauer des Moduls „Patientenaufnahme“.
	Typ:	Float
	Wert:	0
	Bereich:	External
minDauer	Beschreibung:	Minimale Bearbeitungsdauer des Moduls „Patientenaufnahme“.
	Typ:	Float
	Wert:	0
	Bereich:	External
Nachbereitung	Beschreibung:	Durchschnittliche Dauer nachbereitender Tätigkeiten im Modul „Patientenaufnahme“.
	Typ:	Float
	Wert:	0
	Bereich:	External

Schnitt	Beschreibung:	Mittlere Bearbeitungsdauer des Moduls „Patientenaufnahme“.
	Typ:	Float
	Wert:	0
	Bereich:	External


Memory:

PatientenTabelle	Beschreibung:	Speichert Werte auf Basis der hinterlegten Datenstruktur.
	Typ:	Root.Vector.PatientenTabelle
	Bereich	External

Ressourcen:

Pflegekraefte	Beschreibung:	Beinhaltet die für das Modul „Patientenaufnahme“ zur Verfügung stehende Anzahl an Pflegekräften
	Typ:	Quantity
	Bereich	External

Instanzen:

Ambulante_Aufnahme#1	
Notaufnahme#1	
stationaere_Aufnahme#1	



## Modul-Name: Ambulante\_Aufnahme

### Kurzbeschreibung:

Dieses Modul beinhaltet die einzelnen Prozessschritte der ambulanten Aufnahme eines Patienten. Diese sind: Erfassung der Patientendaten, Transport des Patienten zum Point-of-Care, die eingehende Untersuchung des Patienten sowie die Definition des vorläufigen Behandlungsplans.

#### Input:

ID_in_Aufnahme	Beschreibung:	Zu bearbeitende Transaktionen werden über diesen Port empfangen.
	Typ:	Scalar int

#### Output:

ID_out_Untersuchung	Beschreibung:	Weitergabe der fertig bearbeiteten Transaktion an das nachgelagerte Modul „Untersuchung_Hauptmodul“
	Typ:	Scalar int
WS_Demand_Aufnahme	Beschreibung:	Abruf der nächsten Transaktion aus der vorgelagerten Warteschlange
	Typ:	Scalar anytype
WS_ID_out_Aufnahme	Beschreibung:	Zwischenpuffern der abgewiesenen Transaktionen in der vorgelagerten Warteschlange.
	Typ:	Scalar int

#### Parameter:

maxDauer	Beschreibung:	Maximale Bearbeitungsdauer des Moduls „ambulante_Aufnahme“.
	Typ:	Float
	Wert:	0
	Bereich:	External
minDauer	Beschreibung:	Minimale Bearbeitungsdauer des Moduls „ambulante_Aufnahme“.
	Typ:	Float
	Wert:	0
	Bereich:	External
Nachbereitung	Beschreibung:	Durchschnittliche Dauer nachbereitender Tätigkeiten des Moduls „ambulante_Aufnahme“.
	Typ:	Float
	Wert:	0
	Bereich:	External

Schnitt	Beschreibung:	Mittlere Bearbeitungsdauer des Moduls „ambulante_Aufnahme“.
	Typ:	float
	Wert:	0
	Bereich:	External






Memory:

PatientenTabelle	Beschreibung:	Speichert Werte auf Basis der hinterlegten Datenstruktur.
	Typ:	Root.Vector.PatientenTabelle
	Bereich	External

Ressourcen:

Pflegekraefte	Beschreibung:	Beinhaltet die für das Modul „ambulante_Aufnahme“ zur Verfügung stehende Anzahl an Pflegekräften
	Typ:	Quantity
	Bereich	External

Instanzen:

Aktualisieren_der_Krankengeschichte#1	
Eingehende_Untersuchung#1	
Erfassung_der_Patientendaten#1	
Transport_zum_POC#1	
vorl_Behandlungsplan_definieren#1	

## Modul-Name: Erfassung\_der\_Patientendaten

### Kurzbeschreibung:

Das Modul bildet die Erfassung der Patientendaten ab. Es besteht die Möglichkeit, für den Arbeitsgang notwendige Ressourcen in Anspruch zu nehmen. Die jeweilige Vorgangsdauer wird mittels einer Poissonverteilung errechnet und anschließend verzögert. Zudem kann der Beginn- und Endzeitpunkt erfasst und in ein entsprechendes Datenfeld des Memory geschrieben werden.

### Input:

ID_in_Aufnahme	Beschreibung:	Zu bearbeitende Transaktionen werden über diesen Port empfangen.
	Typ:	Scalar int

### Output:

ID_out	Beschreibung:	Weitergabe der fertig bearbeiteten Transaktionen an das nachgelagerte Modul.
	Typ:	Scalar int
WS_Demand	Beschreibung:	Abruf der nächsten Transaktion aus der vorgelagerten Warteschlange
	Typ:	Scalar anytype
WS_ID_out	Beschreibung:	Abgewiesene Transaktionen werden an die vorgelagerte Warteschlange geleitet.
	Typ:	Scalar int

### Parameter:

BeginnVorgang	Beschreibung:	Name des Datenfelds, in das der Ereigniszeitpunkt zu Beginn des Vorgangs geschrieben werden soll.
	Typ:	Datastructmembername
	Wert:	Beginn_ErfassungPatDaten
	Bereich:	External
DauerVorgang	Beschreibung:	Name des Datenfelds, in das die Bearbeitungsdauer des Vorgangs geschrieben werden soll.
	Typ:	Datastructmembername
	Wert:	Dauer_ErfassungPatDaten
	Bereich:	External

EndeVorgang	Beschreibung:	Name des Datenfelds, in das der Ereigniszeitpunkt zum Ende des Vorgangs geschrieben werden soll.
	Typ:	Datastructmembername
	Wert:	Ende_ErfassungPatDaten
	Bereich:	External

maxDauer	Beschreibung:	Maximale Bearbeitungsdauer des Moduls „Erfassung_der_Patientendaten“.
	Typ:	Float
	Wert:	0
	Bereich:	External

minDauer	Beschreibung:	Minimale Bearbeitungsdauer des Moduls „Erfassung_der_Patientendaten“.
	Typ:	Float
	Wert:	0
	Bereich:	External

Schnitt	Beschreibung:	Mittlere Bearbeitungsdauer des Moduls „Erfassung_der_Patientendaten“.
	Typ:	Float
	Wert:	0
	Bereich:	External







#### Memory:

PatientenTabelle	Beschreibung:	Speichert Werte auf Basis der hinterlegten Datenstruktur..
	Typ:	Root.Vector.PatientenTabelle
	Bereich:	External

#### Ressourcen:

Pflegekraefte	Beschreibung:	Beinhaltet die für das Modul „Erfassung_der_Patientendaten“ zur Verfügung stehende Anzahl an Pflegekräften
	Typ:	Quantity
	Bereich:	External

#### Instanzen:

AllocateParam#1	
FreeParam#1	
Poissonverteilung#1	
Schreibe_EZP#1	
Schreibe_EZP#2	
VarAndDelay#1	

## Modul-Name: Transport\_zum\_PoC

### Kurzbeschreibung:

Abbildung des Prozessschrittes „Transport des Patienten zum Point-of-Care“. Es besteht die Möglichkeit, für den Arbeitsgang notwendige Ressourcen in Anspruch zu nehmen. Die jeweilige Vorgangsdauer wird mittels einer Poissonverteilung errechnet und anschließend verzögert. Zudem kann der Beginn- und Endzeitpunkt erfasst und in ein entsprechendes Datenfeld des Memory geschrieben werden.

### Input:

ID_in	Beschreibung:	Zu bearbeitende Transaktionen werden über diesen Port empfangen.
	Typ:	Scalar int

### Output:

ID_out	Beschreibung:	Weitergabe der fertig bearbeiteten Transaktionen an das nachgelagerte Modul
	Typ:	Scalar int

### Parameter:

BeginnVorgang	Beschreibung:	Name des Datenfelds, in das der Ereigniszeitpunkt zu Beginn des Vorgangs geschrieben werden soll.
	Typ:	Datastructmembername
	Wert:	Beginn_TransportPoC
	Bereich:	External
DauerVorgang	Beschreibung:	Name des Datenfelds, in das die Bearbeitungsdauer des Vorgangs geschrieben werden soll.
	Typ:	Datastructmembername
	Wert:	Dauer_TransportPoC
	Bereich:	External
EndeVorgang	Beschreibung:	Name des Datenfelds, in das der Ereigniszeitpunkt zum Ende des Vorgangs geschrieben werden soll.
	Typ:	Datastructmembername
	Wert:	Ende_TransportPoC
	Bereich:	External

maxDauer	Beschreibung:	Maximale Bearbeitungsdauer des Moduls „Transport_zum_PoC“.
	Typ:	Float
	Wert:	0
	Bereich:	External
minDauer	Beschreibung:	Minimale Bearbeitungsdauer des Moduls „Transport_zum_PoC“.
	Typ:	Float
	Wert:	0
	Bereich:	External
Schnitt	Beschreibung:	Mittlere Bearbeitungsdauer des Moduls „Transport_zum_PoC“.
	Typ:	Float
	Wert:	0
	Bereich:	External








Memory:

PatientenTabelle	Beschreibung:	Speichert Werte auf Basis der hinterlegten Datenstruktur.
	Typ:	Root.Vector.PatientenTabelle
	Bereich:	External

Ressourcen:

Pflegekraefte	Beschreibung:	Beinhaltet die für das Modul „Transport_zum_PoC“ zur Verfügung stehende Anzahl an Pflegekräften.
	Typ:	Quantity
	Bereich:	External

Instanzen:

AllocateParam#1	
FIFOQueue#1	
FreeParam#1	
Poissonverteilung#1	
Schreibe_EZP#1	
Schreibe_EZP#2	
VarAndDelay#1	

## Modul-Name: Aktualisieren\_der\_Krankengeschichte

### Kurzbeschreibung:

Dieses Modul bildet den Prozessschritt „Erfassung und Aktualisierung der Krankengeschichte“ ab. Es besteht die Möglichkeit, für den Arbeitsgang notwendige Ressourcen in Anspruch zu nehmen. Die jeweilige Vorgangsdauer wird mittels einer Poissonverteilung errechnet und anschließend verzögert. Zudem kann der Beginn- und Endzeitpunkt erfasst und in ein entsprechendes Datenfeld des Memory geschrieben werden.

### Input:

ID_in	Beschreibung:	Zu bearbeitende Transaktionen werden über diesen Port empfangen.
	Typ:	Scalar int

### Output:

ID_out	Beschreibung:	Weitergabe der fertig bearbeiteten Transaktionen an das nachgelagerte Modul.
	Typ:	Scalar int

### Parameter:

BeginnVorgang	Beschreibung:	Name des Datenfelds, in das der Ereigniszeitpunkt zu Beginn des Vorgangs geschrieben werden soll.
	Typ:	Datastructmembername
	Wert:	Beginn_AktualKG
	Bereich:	External
DauerVorgang	Beschreibung:	Name des Datenfelds, in das die Bearbeitungsdauer des Vorgangs geschrieben werden soll.
	Typ:	Datastructmembername
	Wert:	Dauer_AktualKG
	Bereich:	External
EndeVorgang	Beschreibung:	Name des Datenfelds, in das der Ereigniszeitpunkt zum Ende des Vorgangs geschrieben werden soll.
	Typ:	Datastructmembername
	Wert:	Ende_AktualKG
	Bereich:	External

maxDauer	Beschreibung:	Maximale Bearbeitungsdauer der Moduls „Aktualisieren der Krankengeschichte“.
	Typ:	Float
	Wert:	0
	Bereich:	External
minDauer	Beschreibung:	Minimale Bearbeitungsdauer des Moduls „Aktualisieren der Krankengeschichte“.
	Typ:	float
	Wert:	0
	Bereich:	External
Schnitt	Beschreibung:	Mittlere Bearbeitungsdauer des Moduls „Aktualisieren der Krankengeschichte“.
	Typ:	float
	Wert:	0
	Bereich:	External








#### Memory:

PatientenTabelle	Beschreibung:	Speichert Werte auf Basis der hinterlegten Datenstruktur.
	Typ:	Root.Vector.PatientenTabelle
	Bereich	External

#### Ressourcen:

Pflegekraefte	Beschreibung:	Beinhaltet die für das Modul „Aktualisieren der Krankengeschichte“ zur Verfügung stehende Anzahl an Pflegekräften
	Typ:	Quantity
	Bereich	External

#### Instanzen:

AllocateParam#1	
FIFOQueue#1	
FreeParam#1	
Poissonverteilung#1	
Schreibe_EZP#1	
Schreibe_EZP#2	
VarAndDelay#1	



## Modul-Name: Eingehende\_Untersuchung

### Kurzbeschreibung:

Dieses Modul bildet den Prozessschritt „Eingehende physikalische Untersuchung der Patienten“ ab. Es besteht die Möglichkeit, für den Arbeitsgang notwendige Ressourcen in Anspruch zu nehmen. Die jeweilige Vorgangsdauer wird mittels einer Poissonverteilung errechnet und anschließend verzögert. Zudem kann der Beginn- und Endzeitpunkt erfasst und in ein entsprechendes Datenfeld des Memory geschrieben werden.

### Input:

ID_in	Beschreibung:	Zu bearbeitende Transaktionen werden über diesen Port empfangen.
	Typ:	Scalar int

### Output:

ID_out	Beschreibung:	Weitergabe der fertig bearbeiteten Transaktionen an das nachgelagerte Modul
	Typ:	Scalar int

### Parameter:

BeginnVorgang	Beschreibung:	Name des Datenfelds, in das der Ereigniszeitpunkt zu Beginn des Vorgangs geschrieben werden soll.
	Typ:	Datastructmembername
	Wert:	Beginn_Untersuchung
	Bereich:	External
DauerVorgang	Beschreibung:	Name des Datenfelds, in das die Bearbeitungsdauer des Vorgangs geschrieben werden soll.
	Typ:	Datastructmembername
	Wert:	Dauer_Untersuchung
	Bereich:	External
EndeVorgang	Beschreibung:	Name des Datenfelds, in das der Ereigniszeitpunkt zum Ende des Vorgangs geschrieben werden soll.
	Typ:	Datastructmembername
	Wert:	Ende_Untersuchung
	Bereich:	External

maxDauer	Beschreibung: Maximale Bearbeitungsdauer der Moduls „Eingehende_Untersuchung“.
	Typ: Float
	Wert: 0
	Bereich: External
minDauer	Beschreibung: Minimale Bearbeitungsdauer des Moduls „Eingehende_Untersuchung“.
	Typ: Float
	Wert: 0
	Bereich: External
Schnitt	Beschreibung: Mittlere Bearbeitungsdauer des Moduls „Eingehende Untersuchung“.
	Typ: Float
	Wert: 0
	Bereich: External








#### Memory:

PatientenTabelle	Beschreibung: Speichert Werte auf Basis der hinterlegten Datenstruktur.
	Typ: Root.Vector.PatientenTabelle
	Bereich: External

#### Ressourcen:

Pflegekraefte	Beschreibung: Beinhaltet die für das Modul „Eingehende Untersuchung“ zur Verfügung stehende Anzahl an Pflegekräften
	Typ: Quantity
	Bereich: External

#### Instanzen:

AllocateParam#1	
FIFOQueue#1	
FreeParam#1	
Poissonverteilung#1	
Schreibe_EZP#1	
Schreibe_EZP#2	
VarAndDelay#1	

## Modul-Name: vorl\_Behandlungsplan\_definieren

### Kurzbeschreibung:

Dieses Modul bildet den Prozessschritt „Definieren des vorläufigen Behandlungsplans“ ab. Es besteht die Möglichkeit, für den Arbeitsgang notwendige Ressourcen in Anspruch zu nehmen. Die jeweilige Vorgangsdauer wird mittels einer Poissonverteilung errechnet und anschließend verzögert. Zudem kann der Beginn- und Endzeitpunkt erfasst und in ein entsprechendes Datenfeld des Memory geschrieben werden.

### Input:

ID_in	Beschreibung:	Zu bearbeitende Transaktionen werden über diesen Port empfangen.
	Typ:	Scalar int

### Output:

ID_out	Beschreibung:	Weitergabe der fertig bearbeiteten Transaktionen an das nachgelagerte Modul.
	Typ:	Scalar int

### Parameter:

BeginnVorgang	Beschreibung:	Name des Datenfelds, in das der Ereigniszeitpunkt zu Beginn des Vorgangs geschrieben werden soll.
	Typ:	Datastructmembername
	Wert:	Beginn_vorl_BehandPlan
	Bereich:	External
DauerVorgang	Beschreibung:	Name des Datenfelds in das die Bearbeitungsdauer des Vorgangs geschrieben werden soll.
	Typ:	Datastructmembername
	Wert:	Dauer_vorl_BehandPlan
	Bereich:	External
EndeVorgang	Beschreibung:	Name des Datenfelds in das der Ereigniszeitpunkt zum Ende des Vorgangs geschrieben werden soll.
	Typ:	Datastructmembername
	Wert:	Ende_vorl_BehandPlan
	Bereich:	External

maxDauer	Beschreibung:	Maximale Bearbeitungsdauer der Moduls „vorl_Behandlungsplan_definieren“.
	Typ:	Float
	Wert:	0
	Bereich:	External
minDauer	Beschreibung:	Minimale Bearbeitungsdauer des Moduls „vorl_Behandlungsplan_definieren“.
	Typ:	Float
	Wert:	0
	Bereich:	External
Schnitt	Beschreibung:	Mittlere Bearbeitungsdauer des Moduls „vorl_Behandlungsplan_definieren“.
	Typ:	Float
	Wert:	0
	Bereich:	External








#### Memory:

PatientenTabelle	Beschreibung:	Speichert Werte auf Basis der hinterlegten Datenstruktur.
	Typ:	Root.Vector.PatientenTabelle
	Bereich	External

#### Ressourcen:

Pflegekraefte	Beschreibung:	Beinhaltet die für das Modul „vorl_Behandlungsplan_definieren“ zur Verfügung stehende Anzahl an Pflegekräften
	Typ:	Quantity
	Bereich	External

#### Instanzen:

AllocateParam#1	
FIFOQueue#1	
FreeParam#1	
Poissonverteilung#1	
Schreibe_EZP#1	
Schreibe_EZP#2	
VarAndDelay#1	

## Modul-Name: Untersuchung\_Hauptmodul

### Kurzbeschreibung:

Dieses Modul bildet den Prozessschritt „Untersuchung“ ab und beinhaltet die Blutabnahme (inklusive Weitergabe der Blutprobe an das Labor), notwendige Voruntersuchungen (darunter u.a. die Computertomographie und die Sonographie), die ärztliche, körperliche Untersuchung sowie das Abrufen der Medikation aus der Apotheke.

### Input:

ID_in_Blutabnahme	Beschreibung:	Zur Blutabnahme vorgesehene Transaktionen werden über diesen Port empfangen.
	Typ:	Scalar int
ID_in_Untersuchung	Beschreibung:	Zur ärztlichen Untersuchung vorgesehene Transaktionen werden über diesen Port empfangen.
	Typ:	Scalar int
ID_in_Voruntersuchung	Beschreibung:	Für mögliche Voruntersuchungen vorgesehene Transaktionen werden über diesen Port empfangen.
	Typ:	Scalar int

### Output:

ID_out_Apotheke	Beschreibung:	Weitergabe der fertig bearbeiteten Transaktionen an das nachgelagerte Modul „Apotheke_Hauptmodul“.
	Typ:	Scalar int
ID_out_CT	Beschreibung:	Weitergabe der fertig bearbeiteten Transaktionen an das nachgelagerte Modul „CT_Hauptmodul“.
	Typ:	Scalar int
ID_out_Elektro	Beschreibung:	Weitergabe der fertig bearbeiteten Transaktionen an das nachgelagerte Modul „EKG_Hauptmodul“.
	Typ:	Scalar int
ID_out_Labor	Beschreibung:	Weitergabe der fertig bearbeiteten Transaktionen an das nachgelagerte Modul „Labor_Hauptmodul“.
	Typ:	Scalar int

ID_out_Entlassung	Beschreibung: Weitergabe der fertig bearbeiteten Transaktionen an das nachgelagerte Modul „Entlassung_Hauptmodul“.
	Typ: Scalar int
ID_out_Sono	Beschreibung: Weitergabe der fertig bearbeiteten Transaktionen an das nachgelagerte Modul „Sono_Hauptmodul“.
	Typ: Scalar int
WS_Demand_Blutabnahme	Beschreibung: Abruf der nächsten Transaktion aus der vorgelagerten Warteschlange.
	Typ: Scalar anytype
WS_Demand_Untersuchung	Beschreibung: Abruf der nächsten Transaktion aus der vorgelagerten Warteschlange.
	Typ: Scalar anytype
WS_ID_out_Blutabnahme	Beschreibung: Weitergabe der abgewiesenen Transaktionen an die vorgelagerte Warteschlange.
	Typ: Scalar int
WS_ID_out_Untersuchung	Beschreibung: Weitergabe der abgewiesenen Transaktionen an die vorgelagerte Warteschlange.
	Typ: Scalar int

Parameter:

Blut_Nachbereitung	Beschreibung: Durchschnittliche Dauer der nachbereitenden Tätigkeiten des Moduls „Blutabnahme_Hauptmodul“.
	Typ: Float
	Wert: 0
	Bereich: External
maxBlut	Beschreibung: Maximale Bearbeitungsdauer des Moduls „Blutabnahme_Hauptmodul“
	Typ: Float
	Wert: 0
	Bereich: External
maxUnters	Beschreibung: Maximale Bearbeitungsdauer des Moduls „Untersuchung_aerztlich“.
	Typ: Float
	Wert: 0
	Bereich: External
minBlut	Beschreibung: Minimale Bearbeitungsdauer des Moduls „Blutabnahme_Hauptmodul“
	Typ: Float
	Wert: 0
	Bereich: External

minUnters	Beschreibung: Minimale Bearbeitungsdauer des Moduls „Untersuchung_aerztlich“.
	Typ: Float
	Wert: 0
	Bereich: External
SchnittBlut	Beschreibung: Durchschnittliche Bearbeitungsdauer des Moduls „Blutabnahme_Hauptmodul“
	Typ: Float
	Wert: 0
	Bereich: External

SchnittUnters	Beschreibung: Durchschnittliche Bearbeitungsdauer des Moduls „Untersuchung_aerztlich“.
	Typ: Float
	Wert: 0
	Bereich: External
Unters_Nachbereitung	Beschreibung: Durchschnittliche Dauer der nachbereitenden Tätigkeiten des Moduls „Untersuchung_aerztlich“.
	Typ: Float
	Wert: 0
	Bereich: External





Memory:

PatientenTabelle	Beschreibung: Speichert Werte auf Basis der hinterlegten Datenstruktur.
	Typ: Root.Vector.PatientenTabelle
	Bereich: External

Ressourcen:

Pflegekraefte	Beschreibung:	Beinhaltet die für das Modul „Untersuchung_Hauptmodul“ zur Verfügung stehende Anzahl an Pflegekräften
	Typ:	Quantity
	Bereich	External
Aerzte	Beschreibung:	Beinhaltet die für das Modul „Untersuchung_Hauptmodul“ zur Verfügung stehende Anzahl an Ärzten
	Typ:	Quantity
	Bereich	Internal
Untersuchungszimmer	Beschreibung:	Beinhaltet die für das Modul „Untersuchung_Hauptmodul“ zur Verfügung stehende Anzahl an Untersuchungszimmer
	Typ:	Quantity
	Bereich	Internal

Instanzen:

Blutabnahme#1	
Therapie_abrufen#1	
Untersuchungsablauf#1	
Untersuchung_#1	



## Modul-Name: Blutabnahme\_Hauptmodul

### Kurzbeschreibung:

Dieses Modul bildet den Prozessschritt „Blutabnahme“ ab. Es beinhaltet die Vorbereitung und Durchführung der Blutabnahme sowie den Transport der Blutproben in das Labor.

### Input:

ID_in_Blutabnahme	Beschreibung: Zur Blutabnahme vorgesehene Transaktionen werden über diesen Port empfangen.
	Typ: Scalar int

### Output:

ID_out_Labor	Beschreibung: Weitergabe der fertig bearbeiteten Transaktionen an das nachgelagerte Modul „Labor_Hauptmodul“.
	Typ: Scalar int
WS_Demand_Blutabnahme	Beschreibung: Abruf der nächsten Transaktion aus der vorgelagerten Warteschlange.
	Typ: Scalar anytype
WS_ID_out_Blutabnahme	Beschreibung: Weitergabe der abgewiesenen Transaktionen an die vorgelagerte Warteschlange.
	Typ: Scalar int

### Parameter:

Blut_Nachbereitung	Beschreibung: Durchschnittliche Dauer der Nachbereitung des Moduls „Blutabnahme_Hauptmodul“.
	Typ: Float
	Wert: 0
	Bereich: External
Intervall	Beschreibung: Intervall (zeitlicher Abstand) der Bluttransporte
	Typ: Float
	Wert: 0
	Bereich: External
maxDauer	Beschreibung: Maximale Bearbeitungsdauer des Moduls „Blutabnahme_Hauptmodul“.
	Typ: Float
	Wert: 0
	Bereich: External

minDauer	Beschreibung: Minimale Bearbeitungsdauer des Moduls „Blutabnahme_Hauptmodul“.
	Typ: Float
	Wert: 0
	Bereich: External
Schnitt	Beschreibung: Durchschnittliche Bearbeitungsdauer des Moduls „Blutabnahme_Hauptmodul“.
	Typ: Float
	Wert: 0
	Bereich: External
Transportmenge	Beschreibung: Anzahl (Umfang) der Blutproben, bei dessen Erreichen ein Transport stattfinden soll
	Typ: Float
	Wert: 0
	Bereich: External




#### Memory:

PatientenTabelle	Beschreibung: Speichert Werte auf Basis der hinterlegten Datenstruktur.
	Typ: Root.Vector.PatientenTabelle
	Bereich: External

#### Ressourcen:

Pflegekraefte	Beschreibung: Beinhaltet die für das Modul „Blutabnahme_Hauptmodul“ zur Verfügung stehende Anzahl an Pflegekräften
	Typ: Quantity
	Bereich: External

#### Instanzen:

Blutabnahme_durchfuehren#1	
Blutabnahme_vorbereiten#1	
Bluttransport#1	

## Modul-Name: Blutabnahme\_vorbereiten

### Kurzbeschreibung:

Dieses Modul bildet den Prozessschritt „Vorbereitung der Blutabnahme“ ab. Es besteht die Möglichkeit, für den Arbeitsgang notwendige Ressourcen in Anspruch zu nehmen. Die jeweilige Vorgangsdauer wird mittels einer Poissonverteilung errechnet und anschließend verzögert. Zudem kann der Beginn- und Endzeitpunkt erfasst und in ein entsprechendes Datenfeld des Memory geschrieben werden.

### Input:

ID_in	Beschreibung:	Zur Vorbereitung der Blutabnahme vorgesehene Transaktionen werden über diesen Port empfangen.
	Typ:	Scalar int

### Output:

ID_out	Beschreibung:	Weitergabe der fertig bearbeiteten Transaktionen an das nachgelagerte Modul „Blutabnahme_durchführen“.
	Typ:	Scalar int
WS_ID_out	Beschreibung:	Weitergabe der abgewiesenen Transaktionen an die vorgelagerte Warteschlange.
	Typ:	Scalar int

### Parameter:

BeginnVorgang	Beschreibung:	Name des Datenfelds, in das der Ereigniszeitpunkt zu Beginn des Vorgangs geschrieben werden soll.
	Typ:	Datastructmembername
	Wert:	Beginn_BlutVor
	Bereich:	External
DauerVorgang	Beschreibung:	Name des Datenfelds, in das die Bearbeitungsdauer des Vorgangs geschrieben werden soll.
	Typ:	Datastructmembername
	Wert:	Dauer_BlutVor
	Bereich:	External
EndeVorgang	Beschreibung:	Name des Datenfelds, in das der Ereigniszeitpunkt zum Ende des Vorgangs geschrieben werden soll.
	Typ:	Datastructmembername
	Wert:	Ende_BlutVor
	Bereich:	External

maxDauer	Beschreibung: Maximale Bearbeitungsdauer der Moduls „Blutabnahme_vorbereiten“.
	Typ: Float
	Wert: 0
	Bereich: External
minDauer	Beschreibung: Minimale Bearbeitungsdauer des Moduls „Blutabnahme_vorbereiten“.
	Typ: Float
	Wert: 0
	Bereich: External
Schnitt	Beschreibung: Mittlere Bearbeitungsdauer des Moduls „Blutabnahme_vorbereiten“.
	Typ: Float
	Wert: 0
	Bereich: External








#### Memory:

PatientenTabelle	Beschreibung: Speichert Werte auf Basis der hinterlegten Datenstruktur.
	Typ: Root.Vector.PatientenTabelle
	Bereich: External

#### Ressourcen:

Pflegekraefte	Beschreibung: Beinhaltet die für das Modul „Blutabnahme_vorbereiten“ zur Verfügung stehende Anzahl an Pflegekräften
	Typ: Quantity
	Bereich: External

#### Instanzen:

AllocateParam#1	
FIFOQueue#1	
FreeParam#1	
Poissonverteilung#1	
Schreibe_EZP#1	
Schreibe_EZP#2	
VarAndDelay#1	

## Modul-Name: Blutabnahme\_durchfuehren

### Kurzbeschreibung:

Dieses Modul bildet den Prozessschritt „Durchführung der Blutabnahme“ ab. Es besteht die Möglichkeit, für den Arbeitsgang notwendige Ressourcen in Anspruch zu nehmen. Die jeweilige Vorgangsdauer wird mittels einer Poissonverteilung errechnet und anschließend verzögert. Zudem kann der Beginn- und Endzeitpunkt erfasst und in ein entsprechendes Datenfeld des Memory geschrieben werden.

### Input:

ID_in	Beschreibung:	Zur Durchführung der Blutabnahme vorgesehene Transaktionen werden über diesen Port empfangen.
	Typ:	Scalar int

### Output:

ID_out	Beschreibung:	Weitergabe der fertig bearbeiteten Transaktionen an das nachgelagerte Modul „Bluttransport“.
	Typ:	Scalar int
WS_Demand_Blutabnahme	Beschreibung:	Abruf der nächsten Transaktion aus der vorgelagerten Warteschlange.
	Typ:	Scalar anytype

### Parameter:

BeginnVorgang	Beschreibung:	Name des Datenfelds, in das der Ereigniszeitpunkt zu Beginn des Vorgangs geschrieben werden soll.
	Typ:	Datastructmembername
	Wert:	Beginn_Blutdurch
	Bereich:	External
Blut_Nachbereitung	Beschreibung:	Durchschnittliche Dauer der Nachbereitung des Moduls „Blutabnahme_durchfuehren“.
	Typ:	Float
	Wert:	0
	Bereich:	External

DauerVorgang	Beschreibung: Name des Datenfelds, in das die Bearbeitungsdauer des Vorgangs geschrieben werden soll. Typ: Datastructmembername Wert: Dauer_Blutdurch Bereich: External
EndeVorgang	Beschreibung: Name des Datenfelds, in den der Ereigniszeitpunkt zum Ende des Vorgangs geschrieben werden soll. Typ: Datastructmembername Wert: Ende_Blutdurch Bereich: External
maxDauer	Beschreibung: Maximale Bearbeitungsdauer der Moduls „Blutabnahme_durchfuehren“. Typ: Float Wert: 0 Bereich: External
minDauer	Beschreibung: Minimale Bearbeitungsdauer des Moduls „Blutabnahme_durchfuehren“. Typ: Float Wert: 0 Bereich: External
Schnitt	Beschreibung: Mittlere Bearbeitungsdauer des Moduls „Blutabnahme_durchfuehren“. Typ: Float Wert: 0 Bereich: External









#### Memory:

PatientenTabelle	Beschreibung: Speichert Werte auf Basis der hinterlegten Datenstruktur. Typ: Root.Vector.PatientenTabelle Bereich: External
------------------	---

#### Ressourcen:

Pflegekraefte	Beschreibung: Beinhaltet die für das Modul „Blutabnahme_durchfuehren“ zur Verfügung stehende Anzahl an Pflegekräften Typ: Quantity Bereich: External
---------------	--

Instanzen:

AllocateParam#1	
Delay#1	
FIFOQueue#1	
FreeParam#1	
Poissonverteilung#1	
Schreibe_EZP#1	
Schreibe_EZP#2	
VarAndDelay#1	

## Modul-Name: Bluttransport

### Kurzbeschreibung:

Dieses Modul bildet den Prozessschritt „Bluttransport“ ab. Es besteht die Möglichkeit, für den Arbeitsgang notwendige Ressourcen in Anspruch zu nehmen. Die jeweilige Vorgangsdauer wird mittels einer Poissonverteilung errechnet und anschließend verzögert. Zudem kann der Beginn- und Endzeitpunkt erfasst und in ein entsprechendes Datenfeld des Memory geschrieben werden.

#### Input:

ID_in	Beschreibung:	Zur Durchführung des Bluttransports vorgesehene Transaktionen werden über diesen Port empfangen.
	Typ:	Scalar int

#### Output:

ID_out	Beschreibung:	Weitergabe der fertig bearbeiteten Transaktionen an das nachgelagerte Modul „Labor_Hauptmodul“.
	Typ:	Scalar int

#### Parameter:

BeginnVorgang	Beschreibung:	Name des Datenfelds, in das der Ereigniszeitpunkt zu Beginn des Vorgangs geschrieben werden soll.
	Typ:	Datastructmembername
	Wert:	Beginn_BTransport
	Bereich:	External
DauerVorgang	Beschreibung:	Name des Datenfelds, in das die Bearbeitungsdauer des Vorgangs geschrieben werden soll.
	Typ:	Datastructmembername
	Wert:	Dauer_BTransport
	Bereich:	External
EndeVorgang	Beschreibung:	Name des Datenfelds, in das der Ereigniszeitpunkt zum Ende des Vorgangs geschrieben werden soll.
	Typ:	Datastructmembername
	Wert:	Ende_BTransport
	Bereich:	External



Intervall	Beschreibung: Intervall (zeitlicher Abstand) der Bluttransporte Typ: Float Wert: 0 Bereich: External
maxDauer	Beschreibung: Maximale Bearbeitungsdauer des Moduls „Bluttransport“. Typ: Float Wert: 0 Bereich: External
minDauer	Beschreibung: Minimale Bearbeitungsdauer des Moduls „Bluttransport“. Typ: Float Wert: 0 Bereich: External
Menge	Beschreibung: Anzahl (Umfang) der Blutproben, bei dessen Erreichen ein Transport stattfindet. Typ: Int Wert: 0 Bereich: External
Schnitt	Beschreibung: Mittlere Bearbeitungsdauer des Moduls „Bluttransport“. Typ: Float Wert: 0 Bereich: External










#### Memory:

PatientenTabelle	Beschreibung: Speichert Werte auf Basis der hinterlegten Datenstruktur. Typ: Root.Vector.PatientenTabelle Bereich: External
------------------	---

#### Ressourcen:

Pflegekraefte	Beschreibung: Beinhaltet die für das Modul „Bluttransport“ zur Verfügung stehende Anzahl an Pflegekräften Typ: Quantity Bereich: External
---------------	---

Instanzen:

AllocateParam#1	
EinstellbarerConst#1	
FIFOQueue#1	
FreeParam#1	
IntervalConst#1	
Poissonverteilung#1	
Schreibe_EZP#1	
Schreibe_EZP#2	
VarDelay#1	

## Modul-Name: Untersuchung\_aerztlich

### Kurzbeschreibung:

Dieses Modul bildet den Prozessschritt „ärztliche Untersuchung“ ab. Es besteht hier die Möglichkeit, für den Arbeitsgang notwendige Ressourcen (Untersuchungszimmer und Ärzte) in Anspruch zu nehmen. Die jeweilige Vorgangsdauer wird mittels einer Poissonverteilung errechnet und anschließend verzögert. Zudem kann der Beginn- und Endzeitpunkt erfasst und in ein entsprechendes Datenfeld des Memory geschrieben werden.

### Input:

ID_in_Untersuchung	Beschreibung:	Zur Durchführung der ärztlichen Untersuchung vorgesehene Transaktionen werden über diesen Port empfangen.
	Typ:	Scalar int

### Output:

ID_out_Entlassung	Beschreibung:	Weitergabe der fertig bearbeiteten Transaktionen an das nachgelagerte Modul „Entlassung_Hauptmodul“.
	Typ:	Scalar int
ID_out_Therapie	Beschreibung:	Weitergabe der fertig bearbeiteten Transaktionen an das nachgelagerte Modul „Therapie_Hauptmodul“.
	Typ:	Scalar int
WS_Demand_Untersuchung	Beschreibung:	Abruf der nächsten Transaktion aus der vorgelagerten Warteschlange.
	Typ:	Scalar anytype
WS_ID_out_Untersuchung	Beschreibung:	Weitergabe der abgewiesenen Transaktionen an die vorgelagerte Warteschlange.
	Typ:	Scalar int

### Parameter:

maxDauer	Beschreibung:	Maximale Bearbeitungsdauer des Moduls „Untersuchung_aerztlich“.
	Typ:	Float
	Wert:	0
	Bereich:	External

minDauer	Beschreibung: Minimale Bearbeitungsdauer des Moduls „Untersuchung_aerztlich“.
	Typ: Float
	Wert: 0
	Bereich: External
Schnitt	Beschreibung: Mittlere Bearbeitungsdauer des Moduls „Untersuchung_aerztlich“.
	Typ: Float
	Wert: 0
	Bereich: External
Unters_Nachbereitung	Beschreibung: Durchschnittliche Dauer der nachbereitenden Tätigkeiten des Moduls „Untersuchung_aerztlich“.
	Typ: Float
	Wert: 0
	Bereich: External

















#### Memory:

PatientenTabelle	Beschreibung: Speichert Werte auf Basis der hinterlegten Datenstruktur.
	Typ: Root.Vector.PatientenTabelle
	Bereich: External

#### Ressourcen:

Aerzte	Beschreibung: Beinhaltet die für das Modul „Untersuchung_aerztlich“ zur Verfügung stehende Anzahl an Ärzten.
	Typ: Quantity
	Bereich: External
Untersuchungszimmer	Beschreibung: Beinhaltet die für das Modul „Untersuchung_aerztlich“ zur Verfügung stehende Anzahl an Untersuchungszimmer.
	Typ: Quantity
	Bereich: External

Instanzen:

AllocateParam#1	
AllocateParam#2	
Befundung#1	
Case.output=2#1	
Delay#1	
FreeParam#1	
FreeParam#2	
GenIntConst#1	
GetTabElement#1	
Merge.input=2#1	
Merge.input=2#2	
Merge.input=2#3	
Poissonverteilung#1	
Schreibe_EZP#1	
Schreibe_EZP#2	
VarAndDelay#1	

## Modul-Name: Untersuchungsablauf

### Kurzbeschreibung:

Dieses Modul leitet die Transaktionen zu den durchzuführenden Voruntersuchungen (wie z.B. Sonographie und Computertomographie).

#### Input:

ID_in_Voruntersuchung	Beschreibung: Für mögliche Voruntersuchungen vorgesehene Transaktionen werden über diesen Port empfangen.
	Typ: Scalar int









#### Output:

ID_out_CT	Beschreibung: Weitergabe der fertig bearbeiteten Transaktionen an das nachgelagerte Modul „CT_Hauptmodul“.
	Typ: Scalar int
ID_out_Elektro	Beschreibung: Weitergabe der fertig bearbeiteten Transaktionen an das nachgelagerte Modul „EKG_Hauptmodul“.
	Typ: Scalar int
ID_out_Sono	Beschreibung: Weitergabe der fertig bearbeiteten Transaktionen an das nachgelagerte Modul „Sono_Hauptmodul“.
	Typ: Scalar int
ID_out_Untersuchung	Beschreibung: Transaktionen, für die keine Voruntersuchungen angesetzt sind, werden an das Modul „Untersuchung_aerztlich“ weitergegeben.
	Typ: Scalar int

#### Memory:

PatientenTabelle	Beschreibung: Speichert Werte auf Basis der hinterlegten Datenstruktur.
	Typ: Root.Vector.PatientenTabelle
	Bereich: External

Instanzen:

Case.output=2#1	
Case.output=2#2	
Case.output=2#3	
Case.output=2#4	
GetTabElement#1	
GetTabElement#2	
GetTabElement#3	
GetTabElement#4	

## Modul-Name: CT\_Hauptmodul

### Kurzbeschreibung:

Dieses Modul beinhaltet die einzelnen Prozessschritte der Computertomographie. Diese sind: Untersuchung terminieren, Untersuchung vorbereiten, Patient vorbereiten, Patient transportieren, Untersuchung durchführen, Ergebnisse auswerten, Befund dokumentieren, Reinigung und Sterilisation.

### Input:

ID_in_CT	Beschreibung:	Für die Computertomographie vorgesehene Transaktionen werden über diesen Port empfangen
	Typ:	Scalar int

### Output:

ID_out_Untersuchung	Beschreibung:	Weitergabe der fertig bearbeiteten Transaktionen an das nachgelagerte Modul „Untersuchung_Hauptmodul“.
	Typ:	Scalar int

### Parameter:

maxDauer	Beschreibung:	Maximale Bearbeitungsdauer der Computertomographie.
	Typ:	float
	Wert:	0
	Bereich:	External
minDauer	Beschreibung:	Minimale Bearbeitungsdauer der Computertomographie.
	Typ:	float
	Wert:	0
	Bereich:	External
Schnitt	Beschreibung:	Mittlere Bearbeitungsdauer der Computertomographie.
	Typ:	float
	Wert:	0
	Bereich:	External












Memory:

PatientenTabelle	Beschreibung:	Speichert Werte auf Basis der hinterlegten Datenstruktur.
	Typ:	Root.Vector.PatientenTabelle
	Bereich	External

Ressourcen:

Ressource	Beschreibung:	Beinhaltet die zur Verfügung stehende Anzahl an Arbeitskräften.
	Typ:	Quantity
	Bereich	External

Instanzen:

Befund_dokumentieren#1	
Ergebnisse_auswerten#1	
Patient_transportieren#1	
Patient_transportieren#2	
Patient_vorbereiten#1	
Reinigung_Sterilisation#1	
Untersuchung_durchfuehren#1	
Untersuchung_terminieren#1	
Untersuchung_vorbereiten#1	

## Modul-Name: Labor\_Hauptmodul

### Kurzbeschreibung:

Dieses Modul beinhaltet die einzelnen Prozessschritte der Laboruntersuchung. Diese sind: Untersuchung vorbereiten, Laborwerte ermitteln, Ergebnisse auswerten, Ergebnisse dokumentieren und archivieren, Ergebnisse weiterleiten.

#### Input:

ID_in_Labor	Beschreibung:	Für die Laboruntersuchung vorgesehene Transaktionen werden über diesen Port empfangen.
	Typ:	Scalar int

#### Output:

ID_out_Untersuchung	Beschreibung:	Weitergabe der fertig bearbeiteten Transaktionen an das nachgelagerte Modul „Untersuchung_Hauptmodul“.
	Typ:	Scalar int

#### Parameter:

maxDauer	Beschreibung:	Maximale Bearbeitungsdauer der Laboruntersuchung.
	Typ:	float
	Wert:	0
	Bereich:	External
minDauer	Beschreibung:	Minimale Bearbeitungsdauer der Laboruntersuchung.
	Typ:	float
	Wert:	0
	Bereich:	External
Schnitt	Beschreibung:	Mittlere Bearbeitungsdauer der Laboruntersuchung.
	Typ:	float
	Wert:	0
	Bereich:	External






#### Memory:

PatientenTabelle	Beschreibung:	Speichert Werte auf Basis der hinterlegten Datenstruktur.
	Typ:	Root.Vector.PatientenTabelle
	Bereich:	External

Ressourcen:

Ressource	Beschreibung:	Beinhaltet die zur Verfügung stehende Anzahl an Arbeitskräften.
	Typ:	Quantity
	Bereich	External

Instanzen:

Befund_dokumentieren#1	
Ergebnisse_auswerten#1	
Ergebnisseweiterleiten#1	
Untersuchung_durchfuehren#1	
Untersuchung_vorbereiten#1	

## Modul-Name: Apotheke\_Hauptmodul

### Kurzbeschreibung:

Dieses Modul beinhaltet die einzelnen Prozessschritte der Apotheke. Diese sind: Vorbereitung, Anfertigung und Versendung der Medikation.

#### Input:

ID_in_Apotheke	Beschreibung:	Für die Apotheke vorgesehene Transaktionen werden über diesen Port empfangen.
	Typ:	Scalar int

#### Output:

ID_out_Therapie	Beschreibung:	Weitergabe der fertig bearbeiteten Transaktionen an das nachgelagerte Modul „Therapie_Hauptmodul“.
	Typ:	Scalar int

#### Parameter:

maxDauer	Beschreibung:	Maximale Bearbeitungsdauer der Apotheke.
	Typ:	Float
	Wert:	0
	Bereich:	External
minDauer	Beschreibung:	Minimale Bearbeitungsdauer der Apotheke.
	Typ:	Float
	Wert:	0
	Bereich:	External
Schnitt	Beschreibung:	Mittlere Bearbeitungsdauer der Apotheke.
	Typ:	Float
	Wert:	0
	Bereich:	External




#### Memory:

PatientenTabelle	Beschreibung:	Speichert Werte auf Basis der hinterlegten Datenstruktur.
	Typ:	Root.Vector.PatientenTabelle
	Bereich:	External

Ressourcen:

Ressource	Beschreibung:	Beinhaltet die zur Verfügung stehende Anzahl an Arbeitskräften.
	Typ:	Quantity
	Bereich	External

Instanzen:

Medikation_anfertigen#1	
Medikation_versenden#1	
Vorbereitung#1	

## Modul-Name: Sono\_Hauptmodul

### Kurzbeschreibung:

Dieses Modul beinhaltet die einzelnen Prozessschritte der Sonographie. Diese sind: Untersuchung terminieren, Untersuchung vorbereiten, Patient vorbereiten, Patient transportieren, Untersuchung durchführen, Ergebnisse auswerten, Befund dokumentieren, Reinigung und Sterilisation.

### Input:

ID_in_Sono	Beschreibung:	Für die Sonographie vorgesehene Transaktionen werden über diesen Port empfangen
	Typ:	Scalar int

### Output:

ID_out_Untersuchung	Beschreibung:	Weitergabe der fertig bearbeiteten Transaktionen an das nachgelagerte Modul „Untersuchung_Hauptmodul“.
	Typ:	Scalar int

### Parameter:

maxDauer	Beschreibung:	Maximale Bearbeitungsdauer der Sonographie.
	Typ:	Float
	Wert:	0
	Bereich:	External
minDauer	Beschreibung:	Minimale Bearbeitungsdauer der Sonographie.
	Typ:	Float
	Wert:	0
	Bereich:	External
Schnitt	Beschreibung:	Mittlere Bearbeitungsdauer der Sonographie.
	Typ:	float
	Wert:	0
	Bereich:	External










### Memory:

PatientenTabelle	Beschreibung:	Speichert Werte auf Basis der hinterlegten Datenstruktur.
	Typ:	Root.Vector.PatientenTabelle
	Bereich:	External

Ressourcen:

Ressource	Beschreibung:	Beinhaltet die zur Verfügung stehende Anzahl an Arbeitskräften.
	Typ:	Quantity
	Bereich	External

Instanzen:

Befund_dokumentieren#1	
Ergebnisse_auswerten#1	
Patient_transportieren#1	
Patient_transportieren#2	
Patient_vorbereiten#1	
Reinigung_Sterilisation#1	
Untersuchung_durchfuehren#1	
Untersuchung_terminieren#1	
Untersuchung_vorbereiten#1	

## Modul-Name: EKG\_Hauptmodul

### Kurzbeschreibung:

Dieses Modul bildet den Prozessschritt „Elektrokardiographie“ ab und beinhaltet die beiden Kategorien Ruhe-EKG und Belastungs-EKG.

#### Input:

ID_in_Elektro	Beschreibung:	Für die Elektrokardiographie vorgesehene Transaktionen werden über diesen Port empfangen.
	Typ:	Scalar int

#### Output:

ID_out_Untersuchung	Beschreibung:	Weitergabe der fertig bearbeiteten Transaktionen an das nachgelagerte Modul „Untersuchung_Hauptmodul“.
	Typ:	Scalar int

#### Parameter:

maxDauer	Beschreibung:	Maximale Bearbeitungsdauer der Elektrokardiographie.
	Typ:	Float
	Wert:	0
	Bereich:	External
minDauer	Beschreibung:	Minimale Bearbeitungsdauer der Elektrokardiographie.
	Typ:	float
	Wert:	0
	Bereich:	External
Schnitt	Beschreibung:	Mittlere Bearbeitungsdauer der Elektrokardiographie.
	Typ:	float
	Wert:	0
	Bereich:	External

#### Memory:



PatientenTabelle	Beschreibung:	Speichert Werte auf Basis der hinterlegten Datenstruktur.
	Typ:	Root.Vector.PatientenTabelle
	Bereich:	External



Ressourcen:

Ressource	Beschreibung:	Beinhaltet die zur Verfügung stehende Anzahl an Arbeitskräften.
	Typ:	Quantity
	Bereich	External

Instanzen:

EKG_Belastung#1	
EKG_Ruhe#1	

## Modul-Name: EKG\_Belastung

### Kurzbeschreibung:

Dieses Modul beinhaltet die einzelnen Prozessschritte des Belastungs-EKGs. Diese sind: Untersuchung terminieren, Untersuchung vorbereiten, Patient vorbereiten, Patient transportieren, Untersuchung unter Belastung durchführen, Ergebnisse auswerten, Befund dokumentieren, Reinigung und Sterilisation.

#### Input:

ID_in	Beschreibung:	Für das Belastungs-EKG vorgesehene Transaktionen werden über diesen Port empfangen
	Typ:	Scalar int

#### Output:

ID_out	Beschreibung:	Weitergabe der fertig bearbeiteten Transaktionen an das nachgelagerte Modul.
	Typ:	Scalar int

#### Parameter:

BeginnVorgang	Beschreibung:	Name des Datenfelds, in das der Ereigniszeitpunkt zu Beginn der Elektrokardiographie (Belastungs-EKG) geschrieben werden soll.
	Typ:	Datastructmembername
	Wert:	Beginn_BelEKG
	Bereich:	External
DauerVorgang	Beschreibung:	Name des Datenfelds, in das die Dauer der Elektrokardiographie (Belastungs-EKG) geschrieben werden soll.
	Typ:	Datastructmembername
	Wert:	Dauer_BelEKG
	Bereich:	External
EndeVorgang	Beschreibung:	Name des Datenfelds, in das der Ereigniszeitpunkt zum Ende der Elektrokardiographie (Belastungs-EKG) geschrieben werden soll.
	Typ:	Datastructmembername
	Wert:	Ende_BelEKG
	Bereich:	External

maxDauer	Beschreibung:	Maximale Dauer der Elektrokardiographie (Belastungs-EKG).
	Typ:	Float
	Wert:	0
	Bereich:	External
minDauer	Beschreibung:	Minimale Dauer der Elektrokardiographie (Belastungs-EKG).
	Typ:	Float
	Wert:	0
	Bereich:	External
Schnitt	Beschreibung:	Mittlere Dauer der Elektrokardiographie (Belastungs-EKG).
	Typ:	Float
	Wert:	0
	Bereich:	External










#### Memory:

PatientenTabelle	Beschreibung:	Speichert Werte auf Basis der hinterlegten Datenstruktur.
	Typ:	Root.Vector.PatientenTabelle
	Bereich:	External

#### Ressourcen:

Ressource	Beschreibung:	Beinhaltet die zur Verfügung stehende Anzahl an Arbeitskräften.
	Typ:	Quantity
	Bereich:	External

#### Instanzen:

Befund_dokumentieren#1	
Ergebnisse_auswerten#1	
Patient_transportieren#1	
Patient_transportieren#2	
Patient_vorbereiten#1	
Reinigung_Sterilisation#1	
Untersuchung_durchfuehren#1	
Untersuchung_terminieren#1	
Untersuchung_vorbereiten#1	

## Modul-Name: EKG\_Ruhe

### Kurzbeschreibung:

Dieses Modul beinhaltet die einzelnen Prozessschritte des Ruhe-EKGs. Diese sind: Untersuchung terminieren, Untersuchung vorbereiten, Patient vorbereiten, Patient transportieren, Untersuchung durchführen, Ergebnisse auswerten, Befund dokumentieren, Reinigung und Sterilisation.

#### Input:

ID_in	Beschreibung:	Für das Ruhe-EKG vorgesehene Transaktionen werden über diesen Port empfangen.
	Typ:	Scalar int

#### Output:

ID_out	Beschreibung:	Weitergabe der fertig bearbeiteten Transaktionen an das nachgelagerte Modul.
	Typ:	Scalar int

#### Parameter:

BeginnVorgang	Beschreibung:	Name des Datenfelds, in das der Ereigniszeitpunkt zu Beginn der Elektrokardiographie (Ruhe-EKG) geschrieben werden soll.
	Typ:	Datastructmembername
	Wert:	Beginn_RuheEKG
	Bereich:	External
DauerVorgang	Beschreibung:	Name des Datenfelds, in das die Dauer der Elektrokardiographie (Ruhe-EKG) geschrieben werden soll.
	Typ:	Datastructmembername
	Wert:	Dauer_RuheEKG
	Bereich:	External
EndeVorgang	Beschreibung:	Name des Datenfelds, in das der Ereigniszeitpunkt zum Ende der Elektrokardiographie (Ruhe-EKG) geschrieben werden soll.
	Typ:	Datastructmembername
	Wert:	Ende_RuheEKG
	Bereich:	External

minDauer	Beschreibung:	Minimale Bearbeitungsdauer der Elektrokardiographie (Ruhe-EKG).
	Typ:	Float
	Wert:	0
	Bereich:	External
Schnitt	Beschreibung:	Mittlere Bearbeitungsdauer der Elektrokardiographie (Ruhe-EKG).
	Typ:	Float
	Wert:	0
	Bereich:	External










Memory:

PatientenTabelle	Beschreibung:	Speichert Werte auf Basis der hinterlegten Datenstruktur.
	Typ:	Root.Vector.PatientenTabelle
	Bereich:	External

Ressourcen:

Ressource	Beschreibung:	Beinhaltet die zur Verfügung stehende Anzahl an Arbeitskräften.
	Typ:	Quantity
	Bereich:	External

Instanzen:

Befund_dokumentieren#1	
Ergebnisse_auswerten#1	
Patient_transportieren#1	
Patient_transportieren#2	
Patient_vorbereiten#1	
Reinigung_Sterilisation#1	
Untersuchung_durchfuehren#1	
Untersuchung_terminieren#1	
Untersuchung_vorbereiten#1	

## Modul-Name: Patient\_transportieren

### Kurzbeschreibung:

Dieses Modul beinhaltet die einzelnen Prozessschritte des Patiententransports. Diese sind: Information des zuständigen Personals, Transportvorbereitungen treffen, Patiententransport durchführen, Patient in Empfang nehmen und Patient umbetten.

#### Input:

ID_in	Beschreibung:	Für den Patiententransport vorgesehene Transaktionen werden über diesen Port empfangen
	Typ:	Scalar int

#### Output:

ID_out	Beschreibung:	Weitergabe der fertig bearbeiteten Transaktionen an das nachgelagerte Modul.
	Typ:	Scalar int

#### Parameter:

BeginnVorgang	Beschreibung:	Name des Datenfelds, in den der Ereigniszeitpunkt zu Beginn des Patiententransports geschrieben werden soll.
	Typ:	Datastructmembername
	Wert:	Beginn_PatTrans
	Bereich:	External
DauerVorgang	Beschreibung:	Name des Datenfelds, in den die Bearbeitungsdauer des Patiententransports geschrieben werden soll.
	Typ:	Datastructmembername
	Wert:	Dauer_PatTrans
	Bereich:	External
EndeVorgang	Beschreibung:	Name des Datenfelds, in den der Ereigniszeitpunkt zum Ende des Patiententransports geschrieben werden soll.
	Typ:	Datastructmembername
	Wert:	Ende_PatTrans
	Bereich:	External

maxDauer	Beschreibung:	Maximale Bearbeitungsdauer des Patiententransports.
	Typ:	Float
	Wert:	0
	Bereich:	External
minDauer	Beschreibung:	Minimale Bearbeitungsdauer des Patiententransports.
	Typ:	Float
	Wert:	0
	Bereich:	External
Schnitt	Beschreibung:	Mittlere Bearbeitungsdauer des Patiententransports.
	Typ:	Float
	Wert:	0
	Bereich:	External






Memory:

PatientenTabelle	Beschreibung:	Speichert Werte auf Basis der hinterlegten Datenstruktur.
	Typ:	Root.Vector.PatientenTabelle
	Bereich:	External

Ressourcen:

Ressource	Beschreibung:	Beinhaltet die zur Verfügung stehende Anzahl an Arbeitskräften.
	Typ:	Quantity
	Bereich:	External

Instanzen:

Patient_empfangen#1	
Patient_transportieren#1	
Patient_umbetten#1	
Personal_informieren#1	
Vorbereitungen_treffen#1	

## Modul-Name: Therapie\_Hauptmodul

### Kurzbeschreibung:

Dieses Modul bildet den Prozess der Therapie (=Patientenbehandlung) ab. Dazu gehören folgende Prozessschritte: Behandlung vorbereiten, Behandlung durchführen, Ergebnisse dokumentieren, Patientnakte aktualisieren.

### Input:

ID_in_Therapie	Beschreibung:	Die zur Therapie vorgesehenen Transaktionen werden über diesen Port empfangen.
	Typ:	Scalar int

### Output:

ID_out_Entlassung	Beschreibung:	Weitergabe der fertig bearbeiteten Transaktionen an das nachgelagerte Modul „Entlassung_Hauptmodul“.
	Typ:	Scalar int
WS_Demand_Therapie	Beschreibung:	Abruf der nächsten Transaktion aus der vorgelagerten Warteschlange.
	Typ:	Scalar anytype
WS_ID_out_Therapie	Beschreibung:	Weitergabe der abgewiesenen Transaktionen an die vorgelagerte Warteschlange.
	Typ:	Scalar int

### Parameter:

Betreuung	Beschreibung:	Durchschnittliche Dauer der Betreuung des Patienten während der Behandlung.
	Typ:	float
	Wert:	0
	Bereich:	External
maxDauer	Beschreibung:	Maximale Bearbeitungsdauer der Therapie.
	Typ:	float
	Wert:	0
	Bereich:	External
minDauer	Beschreibung:	Minimale Bearbeitungsdauer der Therapie.
	Typ:	float
	Wert:	0
	Bereich:	External



Schnitt	Beschreibung:	Durchschnittliche Bearbeitungsdauer der Therapie.
	Typ:	float
	Wert:	0
	Bereich:	External






Memory:

PatientenTabelle	Beschreibung:	Speichert Werte auf Basis der hinterlegten Datenstruktur.
	Typ:	Root.Vector.PatientenTabelle
	Bereich	External

Ressourcen:

Pflegekraefte	Beschreibung:	Beinhaltet die für das Modul „Therapie_Hauptmodul“ zur Verfügung stehende Anzahl an Pflegekräften
	Typ: Bereich	Quantity External
Therapieplaetze	Beschreibung:	Beinhaltet die für das Modul „Therapie_Hauptmodul“ zur Verfügung stehende Anzahl an Therapieplätzen.
	Typ: Bereich	Quantity Internal

Instanzen:

Behandlung#1	
Therapie_vorbereiten#1	
Therapie_dokumentieren#1	
Patientenakte_aktualisieren#1	
QrGraph#1	

## Modul-Name: Behandlung\_vorbereiten

### Kurzbeschreibung:

Dieses Modul bildet den Prozessschritt „Vorbereitung der Behandlung“ ab. Es besteht die Möglichkeit, für den Arbeitsgang notwendige Ressourcen (darunter Pflegekräfte und Therapieplätze) in Anspruch zu nehmen. Die jeweilige Vorgangsdauer wird mittels einer Poissonverteilung errechnet und anschließend verzögert. Zudem kann der Beginn- und Endzeitpunkt erfasst und in ein entsprechendes Datenfeld des Memory geschrieben werden.

### Input:

ID_in	Beschreibung:	Für die Behandlung vorgesehene Transaktionen werden über diesen Port zur Behandlungsvorbereitung empfangen.
	Typ:	Scalar int

### Output:

ID_out	Beschreibung:	Weitergabe der fertig bearbeiteten Transaktionen an das nachgelagerte Modul „Behandlung_durchfuehren“.
	Typ:	Scalar int
WS_ID_out	Beschreibung:	Weitergabe der abgewiesenen Transaktionen an die vorgelagerte Warteschlange.
	Typ:	Scalar int

### Parameter:

BeginnVorgang	Beschreibung:	Name des Datenfelds, in das der Ereigniszeitpunkt zu Beginn des Vorgangs geschrieben werden soll.
	Typ:	Datastructmembername
	Wert:	Beginn_BehVorb
	Bereich:	External
DauerVorgang	Beschreibung:	Name des Datenfelds, in das die Bearbeitungsdauer des Vorgangs geschrieben werden soll.
	Typ:	Datastructmembername
	Wert:	Dauer_BehVorb
	Bereich:	External

EndeVorgang	Beschreibung: Name des Datenfelds, in das der Ereigniszeitpunkt zum Ende des Vorgangs geschrieben werden soll. Typ: Datastructmembername Wert: Ende_BehVorb Bereich: External
maxDauer	Beschreibung: Maximale Bearbeitungsdauer des Moduls „Behandlung_vorbereiten“. Typ: Float Wert: 0 Bereich: External
minDauer	Beschreibung: Minimale Bearbeitungsdauer des Moduls „Behandlung_vorbereiten“. Typ: Float Wert: 0 Bereich: External
Schnitt	Beschreibung: Mittlere Bearbeitungsdauer des Moduls „Behandlung_vorbereiten“. Typ: Float Wert: 0 Bereich: External












#### Memory:

PatientenTabelle	Beschreibung: Speichert Werte auf Basis der hinterlegten Datenstruktur. Typ: Root.Vector.PatientenTabelle Bereich: External
------------------	---

#### Ressourcen:

Pflegekraefte	Beschreibung: Beinhaltet die für das Modul „Behandlung_vorbereiten“ zur Verfügung stehende Anzahl an Pflegekräften Typ: Quantity Bereich: External
Therapieplaetze	Beschreibung: Beinhaltet die für das Modul „Behandlung_vorbereiten“ zur Verfügung stehende Anzahl an Therapieplätzen Typ: Quantity Bereich: External

Instanzen:

AllocateParam#1	
AllocateParam#2	
FreeParam#1	
FreeParam#2	
FreeParam#3	
Merge.input=2#1	
Poissonverteilung#1	
Schreibe_EZP#1	
Schreibe_EZP#2	
Schreibe_EZP#3	
VarAndDelay#1	

## Modul-Name: Behandlung\_durchfuehren

### Kurzbeschreibung:

Dieses Modul bildet den Prozessschritt „Durchführung der Behandlung“ ab. Es besteht die Möglichkeit, für den Arbeitsgang notwendige Ressourcen (darunter Pflegekräfte und Therapieplätze) in Anspruch zu nehmen. Die jeweilige Vorgangsdauer wird mittels einer Poissonverteilung errechnet und anschließend verzögert. Zudem kann der Beginn- und Endzeitpunkt erfasst und in ein entsprechendes Datenfeld des Memory geschrieben werden.

### Input:

ID_in	Beschreibung:	Die zur Behandlungsdurchführung vorgesehenen Transaktionen werden über diesen Port empfangen.
	Typ:	Scalar int

### Output:

ID_out	Beschreibung:	Weitergabe der fertig bearbeiteten Transaktionen an das nachgelagerte Modul.
	Typ:	Scalar int
Pflege_out	Beschreibung:	Weitergabe der fertig bearbeiteten Transaktionen an das nachgelagerte Modul „Ergebnisse_dokumentieren“.
	Typ:	Scalar int

### Parameter:

BeginnVorgang	Beschreibung:	Name des Datenfelds, in das der Ereigniszeitpunkt zu Beginn des Vorgangs geschrieben werden soll.
	Typ:	Datastructmembername
	Wert:	Beginn_BehDurch
	Bereich:	External
Betreuung	Beschreibung:	Durchschnittliche Dauer der Betreuung des Patienten während der Behandlung.
	Typ:	Float
	Wert:	0
	Bereich:	External
DauerVorgang	Beschreibung:	Name des Datenfelds, in das die Bearbeitungsdauer des Vorgangs geschrieben werden soll.
	Typ:	Datastructmembername
	Wert:	Dauer_BehDurch
	Bereich:	External

EndeVorgang	Beschreibung: Name des Datenfelds, in das der Ereigniszeitpunkt zum Ende des Vorgangs geschrieben werden soll. Typ: Datastructmembername Wert: Ende_BehDurch Bereich: External
maxDauer	Beschreibung: Maximale Bearbeitungsdauer des Moduls „Behandlung_durchfuehren“. Typ: Float Wert: 0 Bereich: External
minDauer	Beschreibung: Minimale Bearbeitungsdauer des Moduls „Behandlung_durchfuehren“. Typ: Float Wert: 0 Bereich: External
Schnitt	Beschreibung: Mittlere Bearbeitungsdauer des Moduls „Behandlung_durchfuehren“. Typ: Float Wert: 0 Bereich: External













#### Memory:

PatientenTabelle	Beschreibung: Speichert Werte auf Basis der hinterlegten Datenstruktur. Typ: Root.Vector.PatientenTabelle Bereich: External
------------------	---

#### Ressourcen:

Pflegekraefte	Beschreibung: Beinhaltet die für das Modul „Behandlung_durchfuehren“ zur Verfügung stehende Anzahl an Pflegekräften Typ: Quantity Bereich: External
Therapieplaetze	Beschreibung: Beinhaltet die für das Modul „Behandlung_durchfuehren“ zur Verfügung stehende Anzahl an Therapieplätzen Typ: Quantity Bereich: External

Instanzen:

AllocateParam#1	
AllocateParam#2	
Delay#1	
FIFOQueue#1	
FreeParam#1	
FreeParam#2	
FreeParam#3	
Merge.input=2#1	
Poissonverteilung#1	
Schreibe_EZP#1	
Schreibe_EZP#2	
VarAndDelay#1	

## Modul-Name: Ergebnisse\_dokumentieren

### Kurzbeschreibung:

Dieses Modul bildet den Prozessschritt „Dokumentieren der Ergebnisse“ ab. Es besteht die Möglichkeit, für den Arbeitsgang notwendige Ressourcen in Anspruch zu nehmen. Die jeweilige Vorgangsdauer wird mittels einer Poissonverteilung errechnet und anschließend verzögert. Zudem kann der Beginn- und Endzeitpunkt erfasst und in ein entsprechendes Datenfeld des Memory geschrieben werden.

### Input:

Pflege_in	Beschreibung:	Die zur Dokumentation der Ergebnisse vorgesehenen Transaktionen werden über diesen Port empfangen.
	Typ:	Scalar int

### Output:

Pflege_out	Beschreibung:	Weitergabe der fertig bearbeiteten Transaktionen an das nachgelagerte Modul „Patientenakte_aktualisieren“.
	Typ:	Scalar int

### Parameter:

BeginnVorgang	Beschreibung:	Name des Datenfelds, in das der Ereigniszeitpunkt zu Beginn des Vorgangs geschrieben werden soll.
	Typ:	Datastructmembername
	Wert:	Beginn_ErgDoku
	Bereich:	External
DauerVorgang	Beschreibung:	Name des Datenfelds, in das die Bearbeitungsdauer des Vorgangs geschrieben werden soll.
	Typ:	Datastructmembername
	Wert:	Dauer_ErgDoku
	Bereich:	External
EndeVorgang	Beschreibung:	Name des Datenfelds, in das der Ereigniszeitpunkt zum Ende des Vorgangs geschrieben werden soll.
	Typ:	Datastructmembername
	Wert:	Ende_ErgDoku
	Bereich:	External



maxDauer	Beschreibung: Maximale Bearbeitungsdauer des Moduls „Ergebnisse_dokumentieren“.
	Typ: Float
	Wert: 0
	Bereich: External
minDauer	Beschreibung: Minimale Bearbeitungsdauer des Moduls „Ergebnisse_dokumentieren“.
	Typ: Float
	Wert: 0
	Bereich: External
Schnitt	Beschreibung: Mittlere Bearbeitungsdauer des Moduls „Ergebnisse_dokumentieren“.
	Typ: Float
	Wert: 0
	Bereich: External








#### Memory:

PatientenTabelle	Beschreibung: Speichert Werte auf Basis der hinterlegten Datenstruktur.
	Typ: Root.Vector.PatientenTabelle
	Bereich: External

#### Ressourcen:

Pflegekraefte	Beschreibung: Beinhaltet die für das Modul „Ergebnisse_dokumentieren“ zur Verfügung stehende Anzahl an Pflegekräften
	Typ: Quantity
	Bereich: External

#### Instanzen:

AllocateParam#1	
FIFOQueue#1	
FreeParam#1	
Poissonverteilung#1	
Schreibe_EZP#1	
Schreibe_EZP#2	
VarAndDelay#1	

## Modul-Name: Patientenakte\_aktualisieren

### Kurzbeschreibung:

Dieses Modul bildet den Prozessschritt „Aktualisierung der Patientenakte“ ab. Es besteht die Möglichkeit, für den Arbeitsgang notwendige Ressourcen in Anspruch zu nehmen. Die jeweilige Vorgangsdauer wird mittels einer Poissonverteilung errechnet und anschließend verzögert. Zudem kann der Beginn- und Endzeitpunkt erfasst und in ein entsprechendes Datenfeld des Memory geschrieben werden.

### Input:

ID_Pflege_in	Beschreibung:	Für die Aktualisierung der Patientenakte vorgesehene Transaktionen werden über diesen Port empfangen.
	Typ:	Scalar int

### Output:

WS_Demand	Beschreibung:	Abruf der nächsten Transaktion aus der vorgelagerten Warteschlange.
	Typ:	Scalar anytype

### Parameter:

BeginnVorgang	Beschreibung:	Name des Datenfelds, in das der Ereigniszeitpunkt zu Beginn des Vorgangs geschrieben werden soll.
	Typ:	Datastructmembername
	Wert:	Beginn_AkteAktual
	Bereich:	External
DauerVorgang	Beschreibung:	Name des Datenfelds, in das die Bearbeitungsdauer des Vorgangs geschrieben werden soll.
	Typ:	Datastructmembername
	Wert:	Dauer_AkteAktual
	Bereich:	External
EndeVorgang	Beschreibung:	Name des Datenfelds, in das der Ereigniszeitpunkt zum Ende des Vorgangs geschrieben werden soll.
	Typ:	Datastructmembername
	Wert:	Ende_AkteAktual
	Bereich:	External

maxDauer	Beschreibung: Maximale Bearbeitungsdauer des Moduls „Patientenakte_aktualisieren“.
	Typ: Float
	Wert: 0
	Bereich: External
minDauer	Beschreibung: Minimale Bearbeitungsdauer des Moduls „Patientenakte_aktualisieren“.
	Typ: Float
	Wert: 0
	Bereich: External
Schnitt	Beschreibung: Mittlere Bearbeitungsdauer des Moduls „Patientenakte_aktualisieren“.
	Typ: Float
	Wert: 0
	Bereich: External








#### Memory:

PatientenTabelle	Beschreibung: Speichert Werte auf Basis der hinterlegten Datenstruktur.
	Typ: Root.Vector.PatientenTabelle
	Bereich: External

#### Ressourcen:

Pflegekraefte	Beschreibung: Beinhaltet die für das Modul „Patientenakte_aktualisieren“ zur Verfügung stehende Anzahl an Pflegekräften
	Typ: Quantity
	Bereich: External

#### Instanzen:

AllocateParam#1	
FIFOQueue#1	
FreeParam#1	
Poissonverteilung#1	
Schreibe_EZP#1	
Schreibe_EZP#2	
VarAndDelay#1	

## Modul-Name: Entlassung\_Hauptmodul

### Kurzbeschreibung:

Dieses Modul bildet den Prozessschritt „Entlassung der Patienten“ ab und beinhaltet das Schreiben des Arztbriefes. Es besteht die Möglichkeit, für den Arbeitsgang notwendige Ressourcen in Anspruch zu nehmen. Die jeweiligen Vorgangsdauern werden mittels einer Poissonverteilung errechnet und anschließend verzögert. Zudem kann der Beginn- und Endzeitpunkt erfasst und in ein entsprechendes Datenfeld des Memory geschrieben werden.

### Input:

ID_in_Therapie	Beschreibung:	Von der Therapie kommende und zur Entlassung vorgesehene Transaktionen werden über diesen Port empfangen.
	Typ:	Scalar int
ID_in_Untersuchung	Beschreibung:	Von der Untersuchung kommende und zur Entlassung vorgesehene Transaktionen werden über diesen Port empfangen.
	Typ:	Scalar int

### Parameter:

ArztmaxDauer	Beschreibung:	Maximale Dauer der Anfertigung des Arztbriefes.
	Typ:	Int
	Wert:	0
	Bereich:	External
ArztminDauer	Beschreibung:	Minimale Dauer der Anfertigung des Arztbriefes.
	Typ:	Int
	Wert:	0
	Bereich:	External
ArztSchnitt	Beschreibung:	Mittlere Dauer der Anfertigung des Arztbriefes.
	Typ:	Float
	Wert:	0
	Bereich:	External
ArztDauerVorgang	Beschreibung:	Name des Datenfeldes, in das die Bearbeitungsdauer des Arztbriefschreibens eingetragen wird.
	Typ:	Datastructmembername
	Wert:	Dauer_Arztbrief
	Bereich:	External

Ausgabe	Beschreibung: Bezeichnung der Ausgabedatei, in die die Ergebniswerte geschrieben werden sollen. Typ: Filename Wert: Ausgabe.txt Bereich: External
BeginnVorgang	Beschreibung: Name des Datenfelds, in das der Ereigniszeitpunkt zu Beginn der Entlassung geschrieben werden soll. Typ: Datastructmembername Wert: Beginn_Entlassung Bereich: External
DauerVorgang	Beschreibung: Name des Datenfelds, in das die Bearbeitungsdauer der Entlassung geschrieben werden soll. Typ: Datastructmembername Wert: Dauer_Entlassung Bereich: External
EndeVorgang	Beschreibung: Name des Datenfelds, in das der Ereigniszeitpunkt zum Ende der Entlassung geschrieben werden soll. Typ: Datastructmembername Wert: Ende_Entlassung Bereich: External
maxDauer	Beschreibung: Maximale Bearbeitungsdauer der Entlassung. Typ: Float Wert: 0 Bereich: External
minDauer	Beschreibung: Minimale Bearbeitungsdauer der Entlassung. Typ: Float Wert: 0 Bereich: External
Schnitt	Beschreibung: Mittlere Bearbeitungsdauer der Entlassung. Typ: Float Wert: 0 Bereich: External
















Memory:

PatientenTabelle	Beschreibung: Speichert Werte auf Basis der hinterlegten Datenstruktur. Typ: Root.Vector.PatientenTabelle Bereich: External
------------------	---

Ressourcen:

Aerzte	Beschreibung:	Beinhaltet die für das Modul „Entlassung_Hauptmodul“ zur Verfügung stehende Anzahl an Ärzten
	Typ:	Quantity
	Bereich	External

Instanzen:

AccessElementVector#1	
AllocateParam#1	
BlackHole#1	
ExecuteInOrder.Out=2#1	
FIFOQueue#1	
FileSinkDS#1	
FreeParam#1	
MemoryRead#1	
Merge.input=2#1	
Poissonverteilung #1	
Poissonverteilung#2	
Schreibe_EZP#1	
Schreibe_EZP#2	
TkText.input=1#1	
VarAndDelay#1	
VarAndDelay#2	