



JOHANNES GUTENBERG–UNIVERSITÄT MAINZ

Aspekte der Modellbildung:
Konzepte und Anwendung in der
Atmungsphysiologie

**Dissertation zur Erlangung des Grades
„Doktor der Naturwissenschaften“**

am Fachbereich Mathematik und Informatik

vorgelegt von
Oliver Noll
geboren in Pirmasens

Mainz, im Februar 2002

Das Kolloquium fand im Jahr 2002 statt.

Inhaltsverzeichnis

| | |
|---|-----------|
| Einleitung | 1 |
| Fragestellung | 1 |
| Übersicht | 4 |
| | |
| I Die Analyse der Grundlagen | 7 |
| | |
| 1 Systeme | 9 |
| 1.1 Objekte | 9 |
| 1.2 Definition der Zeit | 10 |
| 1.3 Relationale Systeme | 11 |
| 1.4 Dynamische Systeme | 14 |
| 1.5 Modulare Systeme | 15 |
| 1.6 Abbildungen von Systemen auf Systeme | 16 |
| | |
| 2 Modelle, Simulation und Modellentwicklung | 20 |
| 2.1 Modelle | 20 |
| 2.1.1 Begriffsbildung | 20 |
| 2.1.2 Der Modellzweck | 22 |
| 2.2 Simulation | 23 |
| 2.2.1 Simulationstypen | 23 |
| 2.2.2 Simulation zur Prognose | 25 |
| 2.3 Modellentwicklung | 25 |
| 2.3.1 Ebenen eines Modells | 26 |
| 2.3.2 Grundmodell | 27 |
| 2.3.3 Ebenenmodell | 28 |
| 2.3.4 Software-Lebenszyklen | 28 |
| | |
| 3 Netzmodelle | 30 |
| 3.1 Strukturnetze | 30 |
| 3.2 Bedingungs-Ereignisnetze und Stellen-Transitionsnetze | 32 |
| 3.3 Level-Raten-Modelle | 34 |
| 3.3.1 Begriffsbildung | 35 |
| 3.3.2 Level-Raten-Diagramme | 38 |
| 3.3.3 Kontinuierliche und diskrete Modellierung | 39 |

| | | |
|------------|---|-----------|
| II | Der Prozess der Operationalisierung | 41 |
| 4 | Das Koordinatensystem der Modellbildung | 43 |
| 4.1 | Der wissenschaftliche Prozess | 43 |
| 4.2 | Begründung des Koordinatensystems | 45 |
| 4.2.1 | Das Koordinatensystem für dingliche Objekte | 45 |
| 4.2.2 | Das Koordinatensystem für begriffliche Objekte | 47 |
| 4.2.3 | Das Koordinatensystem bei komplexen Strukturen | 48 |
| 4.3 | Das Koordinatensystem im Software Engineering | 49 |
| 4.3.1 | Vergleich mit den Ebenenmodellen | 49 |
| 4.3.2 | Prototyping | 50 |
| 5 | Modellentwicklung | 51 |
| 5.1 | Analyse in der Modellentwicklung | 51 |
| 5.1.1 | Wissensakquisition | 51 |
| 5.1.2 | Systemanalyse | 54 |
| 5.1.3 | Einordnung der Verfahren | 55 |
| 5.2 | Operationalisierung in der Modellentwicklung | 56 |
| 5.2.1 | Prototypen und Modellgeneratoren | 57 |
| 5.2.2 | Eine komponentenorientierte generische Struktur für Level-Raten- Modelle | 59 |
| 5.2.3 | Die Modellierung der Zeit | 63 |
| 5.3 | Kalibrierung | 65 |
| 5.3.1 | Begriffsbildung zur Kalibrierung | 65 |
| 5.3.2 | Standardmethoden der Kalibrierung | 66 |
| 5.3.3 | Kalibrierung durch Data Mining | 67 |
| 6 | Validierung | 70 |
| 6.1 | Analyse der Ansätze in der Literatur | 70 |
| 6.1.1 | Validität bei Bossel | 70 |
| 6.1.2 | Validität bei Uthmann | 72 |
| 6.1.3 | Validität in den Sozialwissenschaften | 73 |
| 6.2 | Validität im Koordinatensystem | 73 |
| 6.2.1 | Begriffsbildung | 74 |
| 6.2.2 | Integration der Ansätze in der Literatur | 75 |
| 6.3 | Validierung im Software Engineering | 76 |
| 6.3.1 | Begriffe | 76 |
| 6.3.2 | Anwendung des Koordinatensystems | 77 |
| III | Die Validierung am Beispiel der Atmungsphysiologie | 79 |
| 7 | EWS und Atmungsphysiologie | 81 |
| 7.1 | EWS — Das Evita-Weaning-System | 81 |
| 7.1.1 | Das EWS-Projekt | 81 |

| | | |
|----------|---|------------|
| 7.1.2 | Beatmung im Modus BIPAP-ASB am Respirator EVITA 4 | 83 |
| 7.2 | Die Ventilation | 85 |
| 7.2.1 | Die Atmungsmechanik | 85 |
| 7.2.2 | Aufteilung des Lungenvolumens | 88 |
| 7.2.3 | Das Gasgemisch | 88 |
| 7.2.4 | Ventilation in den seriellen Totraum | 89 |
| 7.3 | Die Perfusion | 90 |
| 7.3.1 | Die Blutzusammensetzung | 91 |
| 7.3.2 | Der Sauerstofftransport | 91 |
| 7.3.3 | Der Kohlendioxidtransport | 94 |
| 7.4 | Die Diffusion | 95 |
| 7.4.1 | Arterialisierung | 95 |
| 7.4.2 | Venöse Beimischung: Shunt | 97 |
| 7.4.3 | Nicht perfundierte Lungenareale: physiologischer Totraum | 98 |
| 7.4.4 | Zusammenfassung: Arterialisierungsfaktoren | 98 |
| 7.5 | Das Dreikompartimente-Modell | 99 |
| 7.6 | Die Gefäße | 100 |
| 8 | Das physiologische Modell | 101 |
| 8.1 | Der Zweck des physiologischen Modells | 101 |
| 8.2 | Die Zeitskalen im physiologischen Modell | 104 |
| 8.3 | Modellierung des Drucksignals | 105 |
| 8.4 | Modellierung der Ventilation | 106 |
| 8.4.1 | Das grundsätzliche Vorgehen | 106 |
| 8.4.2 | Das Grundmodell der Ventilation | 107 |
| 8.4.3 | Das Dreikompartimente-Modell in elektrotechnischer Formulierung | 109 |
| 8.4.4 | Das Atemzugvolumen | 110 |
| 8.5 | Modellierung von Sauerstoff- und Kohlendioxidaustausches | 111 |
| 8.5.1 | Das grundsätzliche Vorgehen | 111 |
| 8.5.2 | Modellierung der Trachea | 112 |
| 8.5.3 | Die Fraktionen in der Lunge | 113 |
| 8.5.4 | Zusammenfassung: Die Sauerstoff- und Kohlendioxidflüsse | 114 |
| 8.6 | Modellierung der Spontanaktivität | 115 |
| 8.6.1 | Das grundsätzliche Vorgehen | 115 |
| 8.6.2 | Die Durchführung | 116 |
| 8.6.3 | Das L/R-Modell der Spontanaktivität | 118 |
| 8.7 | Modellierung der Perfusion | 119 |
| 8.8 | Modellierung der Diffusion | 119 |
| 8.8.1 | Das grundsätzliche Vorgehen | 120 |
| 8.8.2 | Die Bindungskurven | 121 |
| 8.8.3 | Die alveolare Arterialisierung | 122 |
| 8.8.4 | Der Gasaustausch in den Gefäßen | 123 |
| 8.8.5 | Der venöse Rückfluss | 123 |
| 8.8.6 | Einfluss des RQ auf die Ventilation | 123 |

| | | |
|----------|--|------------|
| 8.9 | Algorithmen des physiologischen Modells | 124 |
| 8.9.1 | Eine Simulation | 124 |
| 8.9.2 | Kalibrierung der Mechanik | 125 |
| 8.9.3 | Bestimmung der Augmentierung | 127 |
| 8.9.4 | Einschwingen: Steady-State des physiologischen Modells | 128 |
| 8.9.5 | Berechnung der Normventilation | 129 |
| 8.9.6 | Die Kalibrierung der Blutgase | 130 |
| 9 | Aspekte der Validierung | 132 |
| 9.1 | Methodik und Grenzen der Validierung | 132 |
| 9.2 | Die Realisierung in einer generischen Struktur | 133 |
| 9.2.1 | Das Klassenmodell der Ventilation | 133 |
| 9.2.2 | Die Klassenmodelle von Diffusion und Perfusion | 136 |
| 9.2.3 | Das Objektmodell des physiologischen Modells | 136 |
| 9.3 | Validierung durch ein Fallbeispiel | 137 |
| 9.3.1 | Ventilation | 137 |
| 9.3.2 | Diffusion und Perfusion | 138 |
| 9.4 | Validierung des Koordinatensystems als Lebenszyklus | 144 |
| | Schlussbetrachtung | 146 |
| | Ergebnisse | 146 |
| | Ausblick | 148 |
| | Anhang | 150 |
| A | Numerische Integration und Runge-Kutta-Verfahren | 150 |
| A.1 | Problemstellung | 150 |
| A.2 | Cauchy-Euler-Integration und Level-Raten-Modelle | 150 |
| A.3 | Runge-Kutta-Verfahren | 151 |
| A.3.1 | Die Idee des Verfahrens | 151 |
| A.3.2 | Das allgemeine Runge-Kutta-Verfahren | 151 |
| A.3.3 | Das Runge-Kutta-Verfahren vierter Ordnung | 152 |
| A.3.4 | Die Implementation des Verfahrens | 153 |
| B | Backpropagation-Netze | 154 |
| B.1 | Überblick | 154 |
| B.2 | Netzstruktur | 155 |
| B.3 | Netzdynamik | 157 |
| B.4 | Die Lernphase | 158 |
| B.5 | Anwendung in der Atmungsphysiologie | 160 |
| C | Physiologisches Modell: Einheiten und Parameter | 162 |
| C.1 | Die Einheiten | 162 |
| C.2 | Die Parameter | 163 |

| | |
|--|------------|
| D Ein wenig Elektrotechnik | 166 |
| E Aspekte der Realisierung | 169 |
| E.1 Die generische Struktur: Das Beispiel eines Kompartimentes | 169 |
| E.1.1 Die Deklaration der Klasse | 169 |
| E.1.2 Die Implementation der Klasse | 170 |
| E.1.3 Die Anwendung der Klasse | 174 |
| E.2 Die Bindungskurven | 175 |
| E.2.1 Kohlendioxid-Bindungskurve | 175 |
| E.2.2 Sauerstoff-Bindungskurve | 178 |
| Literaturverzeichnis | 186 |
| Abbildungsverzeichnis | 189 |
| Danksagung | 190 |
| Index | 191 |

Einleitung

Fragestellung

Eine Arbeit, die sich mit informatischer Modellbildung befasst, enthält notwendig eine THEORETISCHE BETRACHTUNG und eine PRAKTISCHE ANWENDUNG: Wird keine Theorie der Modellbildung geliefert, so ist sie entweder ein Beitrag zum Software Engineering, gehört in die Ingenieurwissenschaften oder ist einer „Bindestrich-Informatik“ zuzuordnen. Umgekehrt kann keine Theorie zur Modellbildung entwickelt werden, die nicht mit den Füßen auf den Boden kommt: Die Theorie muss sich an einem angemessenen Beispiel bewähren. In diesem Spannungsfeld steht die vorliegende Arbeit ASPEKTE DER MODELLBILDUNG: KONZEPTE UND ANWENDUNG IN DER ATMUNGSPHYSIOLOGIE.

Wie so oft, sind die Ideen dieser Arbeit in einem „bottom-up“ Prozess entstanden. Ausgangspunkt war das EWS-PROJEKT . EWS steht für EVITA-WEANING-SYSTEM. Dieses wird in einer Kooperation zwischen der Klinik für Anästhesiologie und dem Institut für Informatik der Johannes Gutenberg-Universität Mainz unter Förderung der Drägerwerk AG, Lübeck, entwickelt. Beim EWS handelt es sich um ein PC-gestütztes Expertensystem, das regelbasiert am Beatmungsgerät Evita 4 der Firma Drägerwerk AG den Prozess der Entwöhnung („Weaning“) von beatmeten Patienten automatisiert. Bei der Entwicklung hat sich gezeigt, dass einzelne nicht ständig messbare Parameter sich nicht in den „Wenn-Dann-Regeln“ des Expertensystems abschätzen lassen. So ist die Idee entstanden, ein Modell der Atmungsphysiologie in das System zu integrieren, das diese Werte per Simulation liefert und so zur Verbesserung der Entscheidungen im Expertensystem beiträgt. Der Kontext innerhalb eines in Echtzeit am Patienten arbeitenden Expertensystems hat es von Anfang an notwendig gemacht, ein GLOBALES PHYSIOLOGISCHES MODELL zu entwickeln, das nicht nur Teilaspekte fokussiert. Dies bedeutet, dass das Modell alle für die Beatmung relevanten Aspekte von der zugeführten Luft bis zum Gasaustausch in den Zellen beinhaltet.

Die Modellierung der Atmungsphysiologie durch Computersimulation hat eine lange Tradition. Als klassisch ist die Arbeit von Grodin et al. [GBB67] aus dem Jahr 1967 zu bewerten. Mit dem System „McPuff“ [Dic77] (1977) ist ein großes Projekt zum Einsatz als Unterrichtssystem entwickelt worden. Mit dem Siegeszug der Rechner sind auch die Ansätze zur Simulation der Atmungsphysiologie Legion geworden — beispielhaft sei [CGU97] genannt. Allerdings ist dem Autor keine Modellierung der Atmungsphysiologie bekannt, die einem in Echtzeit am Patienten arbeitenden System als Entscheidungshilfe dient.

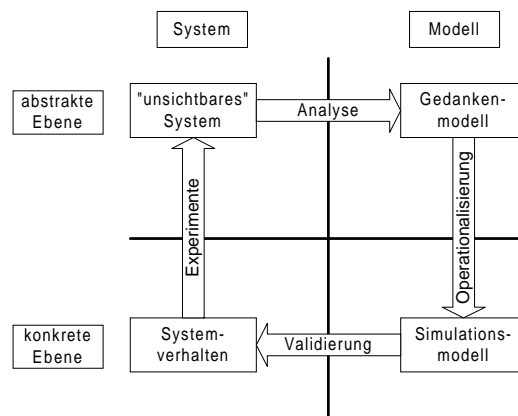


Abbildung 1: Das allgemeine Koordinatensystem der Modellbildung

Vor dem Hintergrund der Modellierung der Atmungsphysiologie im Rahmen des EWS gilt es, die notwendigen Konzepte und Methoden der Modellbildung bereitzustellen und zu entwickeln. Dies ist in erster Linie die Begriffsbildung für dynamische Systeme. Es wird ein geeigneter Systembegriff ausgewählt und eine passende Klasse von Modellen, die Level-Raten-Modelle, entwickelt, die schließlich in einem geeigneten Software-Entwurf münden muss. Da die zu betrachtenden Systeme komplex sind und daher strukturiert werden müssen, hat die gesamte Begriffsbildung modular zu erfolgen, so dass Teilsysteme lokalisiert werden können.

Während des Prozesses der Modellierung der Atmungsphysiologie ist dem Autor immer wieder aufgefallen, dass die Arbeit methodisch stark dem Software Engineering verwandt ist. Dieser Aspekt wird zwar in Lehrbüchern zur Modellbildung erwähnt, aber nicht konsequent verfolgt. Insbesondere haben sich in der Modellbildung nicht die mannigfaltigen LEBENSZYKLEN durchsetzen können. Der Grund hierfür mag darin liegen, dass Modellbildung im Allgemeinen ein weites Feld ist.

Dennoch lässt sich Modellbildung schon anhand des grundsätzlichen Dualismus zwischen dem, was ist — dem SYSTEM — und was daraus gemacht wird — dem MODELL — strukturieren. Diese beiden Dimensionen unterteilen sich jeweils für sich in zwei Ebenen: Auf der Systemseite können einzelne Aspekte beobachtet werden, während anderes für den Beobachter unsichtbar ist. Andererseits startet eine Modellbildung damit, dass man sich einen Begriff von der Sache macht, ein Gedankenmodell entwirft und dieses schließlich in simulationsfähiges Modell transferiert. Es ist keineswegs sicher, dass das Gedankenmodell und das erzeugte Modell übereinstimmen bzw. das erzeugte sich so verhält, wie man es vom gedachten Modell erwartet. Kurz: Auch auf der Modellseite kann zwischen einer ABSTRAKTEN und einer KONKRETEN Ebene unterschieden werden. Damit erscheint es plausibel, jegliche Modellbildung lokal in ein KOORDINATENSYSTEM DER MODELLBILDUNG einzubinden, wie es Abbildung 1 zeigt: Das Koordinatensystem trennt vertikal die System- von der Modellebene und horizontal die konkrete von der abstrakten Ebene. Auch wenn sich die Übergänge von einem Quadranten in den anderen in verschiedenen Problem-bereichen unterschiedlich konkretisieren, lässt sich dennoch auf der abstrakten Ebene ein typisches Vorgehen ausmachen:

Ausgangspunkt ist der III. Quadrant, der den beobachtbaren Teil des Systems symbolisiert. Mit Hilfe von Experimenten erfährt man etwas über die nicht manifesten Teile des Systems, die im zweiten Quadranten symbolisiert werden. Allerdings wird dieser Schritt nur über die Vernunft mit Hilfe von Begriffsbildungen und Analysen über ein Gedankenmodell (I. Quadrant) gemacht. In der Praxis ist es oft schwer, zwischen der (unsichtbaren) Systemstruktur und dem Begriff, den man davon hat, zu unterscheiden. Es ist aber gerade diese Differenzierung, die das Koordinatensystem von gängigen Modellbildungszyklen abhebt und so zu einer klareren Strukturierung beiträgt. Insgesamt wird zwischen einer Experimentier- und Analysephase unterschieden. Letztere stellt den Schritt des Übergangs vom System in ein Gedankenmodell dar. Dieses wird zu einem konkreten Simulationsmodell (IV. Quadrant) operationalisiert, das am beobachteten Systemverhalten validiert werden muss.

Diesen idealen Prozess wird man in reiner Form nie durchführen können. Es müssen mindestens Revisionsprozesse vorgesehen werden, die hier noch vernachlässigt werden. Mit dieser ersten Beschreibung soll das weitere Verfolgen der Arbeit erleichtert werden. Sie stellt das Ergebnis einer Modellbildung dar und hat daher dem Schema zu gehorchen. Daher ist hier die geeignete Stelle, kurz auf die Experimentierphase einzugehen. Im Zuge der Modellbildung ist die Experimentierphase im vorwissenschaftlichen Bereich. Sie dient dazu, sich über die Strukturen des betrachteten Problems klar zu werden, die Ergebnisse zu sammeln. Mit der eigentlichen Begründung hat diese Phase nichts zu tun.

Der entscheidende Impuls zum Aufstellen des Koordinatensystem war ein Artikel aus den empirischen Sozialwissenschaften: Jürgen W. Falter beschäftigte sich mit der „Validierung theoretischer Konstrukte — Wissenschaftstheoretische Aspekte des Validierungskonzepts.“ Dort wird anhand sozialwissenschaftlicher Konstrukte zwischen der MANIFESTEN und LATENTEN Objektdimension, sowie den daraus entwickelten begrifflichen KONSTRUKTEN und den sie messenden INDIKATOREN unterschieden. Im zweiten Teil der Arbeit wird auf diesen Ansatz genauer eingegangen. Von diesem Text ausgehend ist es gelungen, die verschiedenen Validierungskonzepte der informatischen Modellbildung in das Koordinatensystem einzupassen und so zu einer Struktur zu kommen, die diese Validierungskonzepte zusammenfasst und ihre Position in einen Gesamtkontext stellt. Dies erschien Grund genug, das Koordinatensystem als Grundstruktur der Modell konsequent begründen zu wollen.

Diese Ideen ließen sich mehr oder wenig ständig in der Praxis testen: Im Zuge der Entwicklung des physiologischen Modells musste sich der Autor immer wieder Rechenschaft ablegen, wo er gerade stand. Die Experimentierphase auf der praktischen Seite bestand wesentlich in der Entwicklung des physiologischen Modells. Dabei ist derart vorgegangen worden, dass zunächst ein funktionaler Prototyp in einem Delphiprogramm entwickelt worden ist. Dieser ist immer mehr verfeinert worden, bis das Modell in seiner Struktur und alle seine Algorithmen entwickelt waren, insbesondere diejenigen zur Kalibrierung des Modells. Erst dann ist eine generische Implementierung angefertigt worden, die zukünftige Weiterentwicklungen erleichtert.

Aus dieser Darstellung ergeben sich als Fragestellungen der Arbeit:

1. Darstellung und Entwicklung begrifflicher Grundlagen zur Modellbildung komplexer dynamischer Systeme.
2. Entwurf und Implementation einer geeigneten generischen Struktur zur Simulation komplexer dynamischer Systeme.
3. Begründung des Koordinatensystems der Modellbildung als deren Grundstruktur.
4. Entwicklung und Implementation eines physiologischen Modells im Rahmen des EWS-Projektes.

Übersicht

Diese Fragestellungen werden in der Arbeit derart behandelt, dass die Gliederung auch dem Koordinatensystem gehorcht und so der Aufbau der Arbeit selbst zur Begründung des Koordinatensystems beiträgt: Die Experimentierphase gehört in den vorwissenschaftlichen Bereich und findet sich nur in einleitenden Bemerkungen. Ansonsten ist die Arbeit in drei Teile gegliedert:

1. Die Analyse der Grundlagen
2. Der Prozess der Operationalisierung
3. Die Validierung am Beispiel der Atmungsphysiologie

Teil 1: Analyse. Hier werden die notwendigen Grundlagen zusammengestellt, sowie die Begriffsbildung für Systeme und Modelle entwickelt. Ausgangspunkt sind formale Definitionen von Systemen im ersten Kapitel. Es wird deutlich werden, dass sie kaum zur praktischen Arbeit benutzt werden können, aber stets den geeigneten Hintergrund bilden.

Dem Dualismus von System und Modell folgend, beschäftigt sich das folgende Kapitel mit den Grundbegriffen zu Modellen. Zunächst werden die Definitionen von Systemen als Modelle von Systemen im ersten Kapitel benutzt, um Allgemeines zu Modellen zu sagen. Anschließend wird das für die Arbeit Notwendige zur Simulation gesagt und schließlich werden einige Modellbildungszyklen vorgestellt.

Das dritte Kapitel stellt mit Netzmodellen eine Klasse von Modellen vor, die in verschiedenen Bereichen einsetzbar sind. Dabei wird ein Zugang vorgestellt, der von der syntaktischen Strukturebene der graphischen Repräsentation über unterschiedliche Interpretationen zu verschiedenen Modelltypen mit unterschiedlichen Simulationstypen führt. So führen Netze mit ganzzahligen Markierungen und ereignisorientierter Simulation zu Petri-Netzen. Mit reellen, zeitabhängigen Markierungen und zeitorientierter Simulation entstehen Level-Raten-Modelle. Damit kann diese Klasse im Problembereich von dynamischen Systemen wie z.B. der Atmungsphysiologie eingesetzt werden. Darüber hinaus kann auch der Prozess der Modellbildung selbst mit Petri-Netzen beschrieben werden.

Teil 2: Operationalisierung. Die Darstellung der Operationalisierung beginnt mit einer Analyse des Modellierungsprozesses. Das Koordinatensystem der Modellbildung wird im ersten Kapitel als die Grundstruktur der Modellbildung näher erläutert: Ausgehend von einer kurzen Analyse des wissenschaftlichen Prozesses wird das Koordinatensystem für verschiedene Strukturen begründet und schließlich anhand der Lebenszyklen im Software Engineering überprüft.

Das zweite Kapitel über Modellentwicklung teilt sich in drei Abschnitte auf. Im ersten Schritt der Modellentwicklung werden Systeme analysiert. Im Kontext dieser Arbeit sind dies die Methoden der Wissensakquisition und Systemanalyse. Die Operationalisierung erfolgt idealtypisch über Prototypen und Modellgeneratoren. Dieses allgemeine Konzept wird konkret in eine komponentenorientierte generische Struktur für modulare Level-Raten-Modelle überführt. Dieser Abschnitt schließt mit einem speziellen Aspekt, der Modellierung der Zeit. Zur Operationalisierung gehört die Kalibrierung eines Modells. Im dritten Abschnitt des zweiten Kapitels wird nach einer geeigneten Begriffsbildung ein Konzept zur Kalibrierung von Modellen durch Data Mining vorgestellt.

Der nächste Schritt der Modellentwicklung ist ein Revisionsschritt. Im dritten Kapitel wird die Validierung betrachtet. Zunächst werden bekannte Ansätze analysiert. Anschließend werden diese in eine Begriffsbildung, die sich aus dem Koordinatensystem ergibt, integriert und schließlich kurz auf Testkonzepte des Software Engineerings angewandt und dadurch überprüft.

Teil 3: Validierung. Im dritten Teil der Arbeit wird das physiologische Modell vorgestellt. Es ist zunächst für sich genommen eine eigenständige Arbeit. Allerdings sollen mit dem physiologischen Modell auch die Ideen der ersten beiden Teile überprüft werden.

Die Darstellung beginnt mit einer Analyse des EWS und Atmungsphysiologie im Rahmen eines unformalen Systemmodells. Zunächst wird der Kontext des physiologischen Modells im EWS beschrieben. Die Atmungsphysiologie wird anhand der Hauptprozesse Ventilation, Perfusion und Diffusion erläutert.

Die Erkenntnisse aus der Atmungsphysiologie werden in das simulierbare physiologische Modell übertragen. Der Schwerpunkt der Darstellung liegt auf der strukturellen Beschreibung des Modells. Am Ende werden die entwickelten Algorithmen vorgestellt. Insbesondere wird hier die im zweiten Teil entwickelte Methode zur Kalibrierung durch Data Mining beispielhaft realisiert.

Der dritte Teil schießt mit einem Kapitel, in dem Aspekte der Validierung betrachtet werden. Voraussetzung für eine Validierung und gleichzeitig Teil derselben ist eine geeignete Implementation: Zunächst wird die Übertragung des physiologischen Modells in die im zweiten Teil vorgestellte generische Struktur erläutert und ein systemanalytisches Objektmodell des physiologischen Modells gegeben. Anschließend werden einige Verhaltensphänomene des physiologischen Modells beschrieben. Eine exaktere Überprüfung ist in einer informatischen Arbeit nicht möglich. Das Kapitel schließt damit, dass am Beispiel der Modellierung der Spontanaktivität das Koordinatensystem als geeigneter Lebenszyklus dargelegt wird.

In der Schlussbetrachtung werden die wichtigsten Ergebnisse nochmals zusammengestellt, und es wird ein Ausblick auf weitere Forschungsfelder gegeben. Schließlich muss hier auf den recht umfangreichen Anhang aufmerksam gemacht werden. Dort werden cursorisch verschiedene Themen behandelt, die im Zuge der Modellbildung benötigt worden sind.

- (A) Numerische Integration und Runge-Kutta-Verfahren
- (B) Backpropagation-Netze
- (C) Einheiten und Parameter des physiologischen Modells
- (D) Ein wenig Elektrotechnik
- (E) Aspekte der Realisierung

Teil I

Die Analyse der Grundlagen

Am Anfang einer Arbeit gilt es, die Grundlagen derart auszuwählen und darzustellen, dass sie im weiteren Verlauf zur Verfügung stehen und auch deren Platz im Gesamtkontext deutlich wird. In diesem Sinne ist DIE ANALYSE DER GRUNDLAGEN mehr als eine Hintereinanderreihung von „Anhangswissen“. Vielmehr wird hier die Phase der Analyse als ein integraler Bestandteil der Modellbildung dargestellt. Die Analyse ist Teil des durch das Koordinatensystem der Modellbildung gegebenen Lebenszyklus.

SYSTEME stehen am Anfang der Modellbildung. Im ersten Kapitel werden auf einer formalen Ebene zeitbezogene Systembegriffe definiert. Insbesondere erfolgt eine Definition von modularen dynamischen Systemen. Aus den Definitionen werden Erkenntnisse für die Modellbildung gefolgert. Spätestens bei der Definition von Morphismen als Abbildungen von einem System in ein anderes zeigt sich, dass formale Modellbildung höchstens einen theoretischen Hintergrund bilden kann, wenn praktische Probleme im Vordergrund stehen.

Das zweite Kapitel beschreibt die weiteren Grundbegriffe MODELLE, SIMULATION UND MODELLENTWICKLUNG. Der Abschnitt über Modelle besteht erstens aus einer Analyse des ersten Kapitels unter dem Aspekt, dass der entwickelte Systembegriff schon ein Modell ist und zweitens einer Analyse der Anforderungsdefinition im Software Engineering. Die Begriffe zur Simulation werden wie die gesamte Theorie unter besonderer Berücksichtigung der Zeit eingeführt. Als Modelle der Modellierungsprozesse werden Lebenszyklen dargestellt.

Im letzten Kapitel dieses ersten Teils wird mit NETZMODELLEN eine Klasse von Modellen eingeführt, die in weiten Bereichen einsetzbar sind. Hier wird ein Zugang gewählt, der von der graphischen Darstellung ausgeht und durch unterschiedliche Interpretationen zu völlig verschiedenen Arten von Modellen führt. Unter anderem kommt man durch einen zeitabhängigen und reellwertigen Markierungsbegriff zusammen mit zeitorientierter Simulation zu Level-Raten-Modellen. Diese lassen sich modularisieren und liefern so eine geeignete Struktur zum Simulieren modularer dynamischer Systeme, wie sie im ersten Kapitel eingeführt worden sind.

Kapitel 1

Systeme

Systeme sind Ausgangspunkt und Gegenstand jeder Modellbildung. Systeme werden von Objekten gebildet. Diese stehen am Anfang der Betrachtungen. Es folgt eine formale Definition der Zeit. Im Zuge einer ständigen Konkretisierung werden zunächst die Systembegriffe definiert, um dann die für die Modellbildung inhaltlich relevanten Punkte aus den Definitionen abzuleiten: Wir beginnen mit dem System als Relation, definieren anschließend dynamische Systeme, erläutern, was modulare Systeme sind, und beschließen dieses Kapitel damit, Modellbildung formal als Abbildung von einem System auf ein anderes zu beschreiben.

1.1 Objekte

Die Brockhaus-Enzyklopädie definiert ein OBJEKT in der Philosophie als „das dem Menschen als dem Erkenntnissubjekt im Erkenntnisprozeß gegenüberstehende Moment, der Gegenstand (lat. res), wobei zw. sinnlich gegebenen und geistigen (intelligiblen) Objekt unterschieden werden kann.“¹ Diese Definition liefert im Sinne der informatischen Modellbildung wichtige Unterscheidungen. Die Objekte können als Gegenstand der Modellbildung zum einen konkrete Objekte und so „sinnlich gegeben“ sein. Zum anderen kann es sich um begriffliche Konstrukte handeln, die in Modellen weiter konkretisiert werden müssen; somit kann man von „geistigen“ Objekten sprechen. Stets sind Objekte wahrnehmbar, sei es mit Hilfe der Sinne oder als Gedanken. Schließlich sind Objekte nur vermittelt dem „Menschen als dem Erkenntnissubjekt“ wahrnehmbar und daher liegt im Erkennen der Objekte eine mögliche Fehlerquelle.

Formal werden die zu betrachtenden Objekte in einer Menge $\mathcal{U} = \{x, y, z, \dots\}$ festgelegt. Diese Konvention soll im konkreten Fall entweder unumstritten oder empirisch überprüfbar sein. Nicht in der Grundmenge enthaltene Elemente werden nicht betrachtet. Die interessierenden Objekte werden unter anderem vermittelt ihrer Eigenschaften erkannt.

Definition 1.1 (Eigenschaft als Abbildung). *Die Teilmenge der betrachteten Objekte \mathcal{U} , denen die Eigenschaft P zukommt, nennt man die Menge der Träger der EIGENSCHAFT*

¹[Bro86] Band 16 S. 76.

P und bezeichnet sie mit $D_P \subseteq \mathcal{U}$. Dann wird der Eigenschaft P eine Abbildung

$$p : D_P \longrightarrow W_P$$

zugeordnet. Die Menge W_P wird als Wertebereich der Eigenschaft P bezeichnet.

Demnach ist zwischen einer Eigenschaft und ihren Ausprägungen zu differenzieren. An dieser Stelle unterscheidet man auch zwischen Attributen und Eigenschaften. Mit Hilfe der Sinne erkennt der Mensch Objekte, und sie entstehen im Verstand mit Hilfe der Vernunft in Begriffen neu. Es kann also zwischen dem Begriff, den man der Eigenschaft eines Objektes zuordnet, und der Eigenschaft selbst, die das Objekt unabhängig vom Erkennen hat, unterschieden werden. Genau diese Differenzierung ist mit der Unterscheidung zwischen Attribut und Eigenschaft gemeint. Das ATTRIBUT wird im Verstand dem Objekt beigelegt. Im Grunde ist ein Attribut stets ein begriffliches Konstrukt, während die EIGENSCHAFT dagegen dem Objekt selbst zugeordnet werden soll.

1.2 Definition der Zeit

Definition 1.2 (Zeitskala). Eine Struktur $(\mathbb{T}, <)$ heißt ZEITSKALA, falls

1. \mathbb{T} eine Menge — die Zeitmenge und
2. „ $<$ “ eine Ordnungsrelation auf T , d.h. „ $<$ “ transitiv, antireflexiv und antisymmetrisch,

ist.

Die Elemente aus \mathbb{T} werden als Zeitpunkte bezeichnet. Stehen je zwei Zeitpunkte in Relation zueinander, so wird die Ordnung total genannt. Handelt es sich bei \mathbb{T} um eine höchstens abzählbare Menge, spricht man von einer DISKRETEN, ansonsten von einer KONTINUIERLICHEN ZEITSKALA. Hat die Zeitskala ein kleinstes (bzw. ein größtes) Element t_m (bzw. t_M), heißt dieses Startzeitpunkt (bzw. Endzeitpunkt). Die Begriffe Zukunft, Vergangenheit, Zeitintervall und ähnliches werden in der gewohnten Weise festgelegt.

Im Folgenden ist der Einfachheit halber mit der Zeitmenge \mathbb{T} gleichzeitig auch die Struktur $(\mathbb{T}, <)$ gemeint. In konkreten Beispielen wird, solange nichts anderes gesagt wird, davon ausgegangen, dass es sich bei der Zeitmenge um $\mathbb{T} = \mathbb{R}_0^+ = \{r \in \mathbb{R} \mid r \geq 0\}$ die nichtnegativen reellen Zahlen handelt und „ $<$ “ die übliche Kleiner-Relation ist. Werden diskrete Zeitmengen verwendet, so kann man diese mit einer geeigneten Teilmenge der natürlichen Zahlen identifizieren.

Definition 1.3 (Signale). Sei X eine Menge und \mathbb{T} eine Zeitskala.

1. Eine Funktion $x : \mathbb{T} \longrightarrow X$ heißt SIGNAL.
2. Sei $\mathbb{T}' \subseteq \mathbb{T}$. Dann nennt man die Einschränkung $x|_{\mathbb{T}'} : \mathbb{T}' \longrightarrow X$ von x auf \mathbb{T}' ein ZEITSEGMENT VON X .

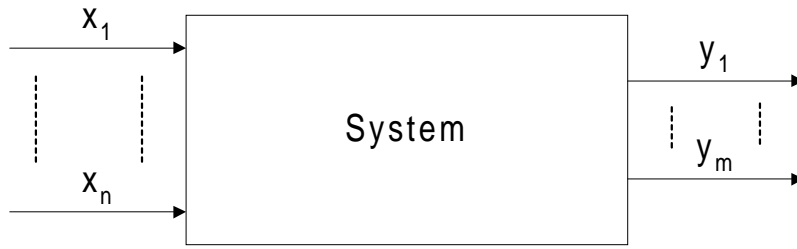


Abbildung 1.1: Systeme als Blackbox

3. Für $t \in \mathbb{T}$ definiert \circ_t

$$x = x_1 \circ_t x_2 = \begin{cases} x(\tau) = x_1(\tau) & , \tau < t \\ x(\tau) = x_2(\tau) & , \tau \geq t \end{cases}$$

das t -KONKATENATIONSPRODUKT von x_1 und x_2 .

4. Für zwei Signale x, y auf \mathbb{T} definiert

$$\begin{aligned} \langle x|y \rangle(t) &: \mathbb{T} \longrightarrow X \times Y \\ \langle x|y \rangle(t) &= (x(t), y(t)) \end{aligned}$$

das direkte Produkt von x und y .

5. $X^{\mathbb{T}}$ bezeichnet die Menge aller möglichen Signale x von \mathbb{T} nach X .

6. $\Xi \subseteq X^{\mathbb{T}}$ wird SIGNALRAUM genannt, falls sie bezüglich der t -Konkatenation abgeschlossen ist.

Im Allgemeinen wird X eine Teilmenge der reellen Zahlen \mathbb{R} bzw. \mathbb{R}^n sein. Im zweiten Fall sind die Definitionen komponentenweise zu lesen. Punkt 6 der Definition leitet zu einer ersten Definition des Begriffs System über. In der Realität werden nicht alle möglichen Signale beobachtbar sein. Vielmehr sollen diejenigen, die auftreten können, ausgezeichnet und als Signalmräume bezeichnet werden. Eine formale Bedingung hierfür ist die Abgeschlossenheit bezüglich der t -Konkatenation, wie sie in der Definition gefordert wird.

1.3 Relationale Systeme

Ein System wird als Relation zwischen zwei Signalen aus den zugehörigen Signalmräumen aufgefasst. Daraus ergeben sich verschiedene Eigenschaften von Systemen, die mit den im Zuge der Modellbildung notwendigen Grenzziehungen in Zusammenhang stehen. Schließlich ziehen wir den internen Zustand eines Systems in Betracht und geben eine Definition von Systemen als Funktion.

Definition 1.4 (Relationales System). Es seien \mathbb{T} eine Zeitskala, $\Xi \subseteq X^{\mathbb{T}}$ und $\Upsilon \subseteq Y^{\mathbb{T}}$ zwei Signalmräume und R_S eine zweistellige Relation auf Ξ und Υ .

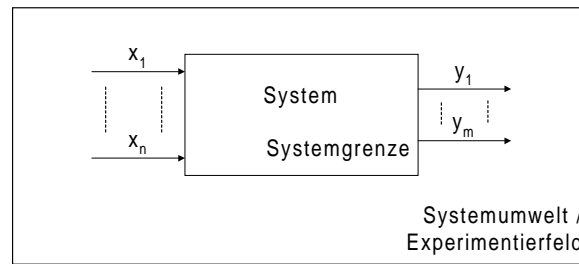


Abbildung 1.2: Systeme als Blackbox mit Experimentierfeld

1. Das Quadrupel $S = (R_S, \Xi, \Upsilon, \mathbb{T})$ heißt ein RELATIONALES SYSTEM.
2. Man bezeichnet $\Xi \times \Upsilon$ als REALISIERUNGSRAUM.
3. PHASENRÄUME sind die Mengen aus $\mathbb{T} \times \Xi$ und $\mathbb{T} \times \Upsilon$.

Mit der üblichen Bedeutung der Elemente aus $x \in \Xi$ und $y \in \Upsilon$ ist ein System eine Relation, die das EINGANGSSIGNAL x in das AUSGANGSSIGNAL y überführt. Der Einfachheit halber nehmen wir an, dass zwei Signale, die in Relation zueinander stehen, synchron sein sollen, d.h. $x(t)R_{Sy}(t') \Rightarrow t = t'$.

Die Relation enthält keine Information über die Struktur des Systems. Es handelt sich um eine BLACKBOX, wie sie in Abbildung 1.1 dargestellt ist. Bereits aus dieser Darstellung lassen sich für den Prozess der Modellierung wesentliche Informationen ableiten. Das System wird von außen mit der Information $x_1 \dots x_n$ gespeist, sie wird innerhalb des Systems transformiert und in Form von $y_1 \dots y_m$ in die Umwelt abgegeben. Daraus ergibt sich, dass im Zuge der Modellbildung eine doppelte Grenzziehung notwendig ist, wie sie in Abbildung 1.2 angedeutet wird:

1. **Systemgrenze:** Es muss festgelegt werden, welche Objekte sich innerhalb und außerhalb des Systems befinden.
2. **Systemumwelt:** Außerhalb des Systems muss eine Übereinkunft getroffen werden, welche Objekte das System beeinflussen, von ihm beeinflusst werden und ggf. die Signale steuern und beobachten; alles Übrige bleibt unberücksichtigt.

Für diese Grenzziehungen des Systems ist notwendig anzugeben, welches die Ein- und Ausgaben des Systems sind. In diesem Sinne muss ein SYSTEMZWECK definiert werden; es muss vereinbart werden, welche Funktion das System erfüllen soll. Diese wird mit Hilfe der WIRKUNGSVERKNÜPFUNGEN (RELATIONEN) DER SYSTEMELEMENTE bewerkstelligt, wie sie in der Definition 1.4 gegebenen Relation R_S festgelegt worden sind. Schließlich hat eine konkrete Systemdefinition so zu erfolgen, dass das System keine „unnötigen“ Elemente enthält. Das System soll NICHT TEILBAR sein: Das „Aufheben der Systemintegrität“ soll zum „Verlust der Systemidentität“ führen. Damit sind aus der formalen Definition 1.4 die Eigenschaften eines Systems gefolgert worden, die von Bossel in einem Lehrbuch über Modellbildung [Bos94] zur Definition² gemacht werden.

²siehe [Bos94] S. 16.

Diese Festlegungen liefern allerdings für die Praxis noch keine hinreichenden Hinweise und Anleitungen für die konkrete Festlegung einer Systemgrenze. Sie werden von Bossel gegeben:

1. „Systemgrenze dort ziehen, wo die Kopplung zur Umgebung sehr viel schwächer ist als die Binnenkopplung im System [...]“.
2. Systemgrenze dort ziehen, wo vorhandene Umweltkopplungen nicht funktionsrelevant sind. [...].
3. Systemgrenze dort ziehen, wo Umwelteinwirkungen nicht durch das System selbst bestimmt werden oder durch Rückkopplungen von Systemauswirkungen verändert werden können [...].“³

Hat man diese Grenze festgelegt, so muss eine zweite Grenze gezogen werden; nämlich diejenige zwischen der Systemumwelt und dem „Nichts“: Aus der Fülle möglicher Umwelteinflüsse müssen die relevanten selektiert werden. Für diese muss dann auch festgelegt werden, in welchem Wertebereich sie auf das System Einfluss nehmen können bzw. in welcher Genauigkeit sie vorliegen. Ebenso müssen aus den möglichen Auswirkungen des Systems auf die Umwelt diejenigen ausgewählt werden, die im Zuge der Modellbildung von Interesse sind. Gleichzeitig ist deren Wertebereich zu klären, so dass hier der potenzielle Realisierungsraum definiert werden muss. Alles Übrige in der Umwelt des Systems bleibt unberücksichtigt. In dieser Sichtweise werden alle Einflüsse der Systemumwelt vereinfachend zu Eingangssignalen des Systems zusammengefasst. Analoges gilt für die Ausgangssignale.

Die Systemumwelt wird stellenweise auch als EXPERIMENTIERFELD bezeichnet.⁴ Mit diesem Begriff ist auch eine andere Interpretation der Systemumwelt verbunden: Hier können Signale als „experimentelle Steuerung“ des Systems begriffen werden, die zu einer gewissen Ausgabe führen. Diese wird von einem Experimentator im Experimentierfeld aufgezeichnet, aufbereitet und mit anderen Versuchsanordnungen verglichen. So kann ein Experimentierfeld auch als Obersystem begriffen werden.

In der bisherigen Systemdarstellung ist es möglich, das selbe Eingangssignal mit verschiedenen Ausgangssignalen in Relation zu setzen. Der Grund hierfür liegt darin, dass der interne Zustand des Systems noch nicht in Betracht gezogen worden ist. Unter Zuhilfenahme der Ausgangsbedingungen, den STARTZUSTÄNDEN, kann die Systemrelation in eine Funktion überführt werden und so das System funktional beschrieben werden. Als typisches Beispiel gilt an dieser Stelle die in einer Differenzialgleichung gegebenen Kurvenschar als ein in einer Relation gegebenes System, das erst durch die Angabe des Anfangswertes zu einem funktionalen System wird. Dies leitet zum nächsten Abschnitt über, in dem dynamische Systeme eingeführt werden, die auch als Differenzialgleichungen darstellbar sind.

³Vgl. [Bos94] S. 18.

⁴Vergleiche hierzu [ZPK00] S. 27. Dort wird von einem „Experimental Frame“ gesprochen.

1.4 Dynamische Systeme

Es treten zwei wesentliche Aspekte zum Systembegriff hinzu: die Rolle der Zeitskala in Bezug auf die Dynamik des Systems und dessen innere Struktur.

Definition 1.5 (Dynamisches System). *Es seien eine Zeitskala \mathbb{T} , $\Xi \subseteq X^{\mathbb{T}}$ und $\Upsilon \subseteq Y^{\mathbb{T}}$ wie in 1.4 definiert. Ferner seien $t, t' \in \mathbb{T}$ mit $t < t'$, $z, \tilde{z} \in Z$, $x \in \Xi$ und $y \in \Upsilon$.*

Ein Tupel $S_{\text{dyn}} = (\mathbb{T}, Z, \Xi, \Upsilon, F, g)$ wird ein DYNAMISCHES SYSTEM genannt, falls gilt:

1. *Die Zustandsmenge Z ist eine beliebige Menge. Mit $z \in Z$ wird ein SYSTEMZUSTAND oder kurz ZUSTAND bezeichnet.*
2. *Die ÜBERFÜHRUNGSFUNKTION F ist eine (möglicherweise partielle) Abbildung*

$$F : \mathbb{T} \times \mathbb{T} \times Z \times \Xi \longrightarrow Z \text{ mit } F(t, t', z, x) = \tilde{z}.$$

3. *Die AUSGABEFUNKTION g ist eine (partielle) Abbildung*

$$g : \mathbb{T} \times Z \times \Xi \longrightarrow \Upsilon \text{ mit } g(t, z, x) = y.$$

Mit der Zustandsmenge Z wird erstmals der interne Zustand des Systems mit in die Begriffsbildung einbezogen. Er wird mit Hilfe der Überföhrungsfunktion manipuliert: Befindet sich ein System zum Zeitpunkt t im Zustand z und erhält das Eingangssignal x , so wird es vermöge der Überföhrungsfunktion F bis zum Zeitpunkt t' in den Zustand \tilde{z} überföhrt. Unter den gleichen Voraussetzungen antwortet das System aufgrund der Ausgabefunktion g zum Zeitpunkt t mit dem Ausgangssignal y . Schließlich lassen sich diese beiden Funktionen verknüpfen, so dass das zukünftige Systemverhalten ebenfalls aus den in der Definition gegebenen Gleichungen gefolgert werden kann.

Definition 1.6 (Systemfunktional). *Sei S_{dyn} ein dynamisches System wie in Definition 1.5. Für $t \leq t'$, Signale $x \in \Xi$, $y \in \Upsilon$ und einen Zustand $z \in Z$ nennt man die Abbildung*

$$G : \mathbb{T} \times \mathbb{T} \times Z \times \Xi \longrightarrow \Upsilon$$

$$G(t, t', z, x) = g(t', F(t, t', z, x(t)), x(t')) = y$$

das SYSTEMFUNKTIONAL von S_{dyn} .

In diesen Definitionen wird zwischen der Ausgabe- und der Überföhrungsfunktion unterschieden. In der Praxis heißt das für die Festlegung eines Systems, dass zwischen den VERHALTENSGRÖSSEN und den ZUSTANDSGRÖSSEN unterschieden werden muss. Jene machen sich in der Systemumwelt bemerkbar, stellen also für den Beobachter des Systems dessen Veränderungen dar. Dagegen bestimmen Zustandsgrößen den internen Zustand des Systems. Sie müssen nicht beobachtbar sein. Vielmehr beschreiben sie den Systemzustand vollständig. Dies spiegelt sich in der Definition der Ausgabefunktion wieder, in die der aktuelle Zustand und ein Eingangssignal als Information eingehen. Da ein System

vollständig durch die Zustandsgrößen beschrieben werden kann und eine Minimalitätsbedingung erfüllen soll (vgl. Abschnitt 1.3), wird eine bestimmte Anzahl von Zustandsgrößen benötigt. Daher ist es sinnvoll, von der DIMENSIONALITÄT DES SYSTEMS zu sprechen.

Die Zustandsgrößen bestimmen auch die Dynamik des Systems. Sie repräsentieren im aktuellen Zustand die vergangene Entwicklung. Die Zukunft des Systems, wie sie in der Überföhrungsfunktion dargestellt ist, bestimmen Zustandsgrößen dadurch, dass sie auf ihre eigene Veränderung Einfluss nehmen können. Es bestehen RÜCKKOPPLUNGEN im System. Diese föhren dazu, dass das Systemverhalten im Allgemeinen nicht vorhersehbar ist. Dies ist einer der wesentlichen Gründe, warum die Simulation von Systemen notwendig wird.

In der bis hierhin eingeföhrten funktionalen Sichtweise auf ein System stehen alle Zustandsgrößen gleichberechtigt nebeneinander. Eine interne Struktur ist nicht vorhanden. Bei komplexeren Systemen ist es auch für Experten schwierig, einen brauchbaren Systembegriff zu entwickeln, ohne konkrete Teilstrukturen zu definieren. Deshalb wird im folgenden Abschnitt ein Konzept zur Strukturierung vorgestellt.

1.5 Modulare Systeme

Systeme von diesem Typ zeichnen sich dadurch aus, dass die internen Zustandsgrößen zu Blöcken zusammengefasst werden. Die Größen innerhalb eines Blocks werden von außen über EINGANGSVARIABLEN angesprochen. Blöcke treten mit Hilfe von AUSGANGSVARIABLEN mit ihrer Umgebung in Kontakt. Interaktionen zwischen der Umwelt und dem System sowie Interaktionen zwischen einzelnen Blöcken müssen in der Definition ebenfalls berücksichtigt werden.

Definition 1.7 (Modulares System). Sei K eine Indexmenge. Ein MODULARES SYSTEM S_Σ wird definiert als

$$S_\Sigma = \left(\mathbb{T}, \Xi_\Sigma, \Upsilon_\Sigma, K, \{(M_k, I_k) | k \in K\}, \{J_k | k \in K \cup \{\Sigma\}\} \right),$$

wobei

$$M_k = (\mathbb{T}, Z, \Xi_k, \Upsilon_k, F, g), \quad k \in K$$

eine SYSTEMKOMPONENTE ist. Sie ist selbst ein dynamisches System. Mit $I_k \subseteq K \times \{\Sigma\}$ wird die Menge der Systemkomponenten bezeichnet, die auf k Einfluss nehmen.

$$J_k : \bigotimes_{i \in I_k} \Upsilon \Xi_i \longrightarrow \Xi \Upsilon_k$$

heißt die SCHNITTSTELLENABBILDUNG von k , wobei

$$\begin{aligned} \Upsilon \Xi_i &= \begin{cases} \Xi_i & , \text{ falls } i = \Sigma \\ \Upsilon_i & , \text{ falls } i \neq \Sigma \end{cases} \\ \Xi \Upsilon_k &= \begin{cases} \Upsilon_k & , \text{ falls } k = \Sigma \\ \Xi_k & , \text{ falls } k \neq \Sigma. \end{cases} \end{aligned} \tag{1.1}$$

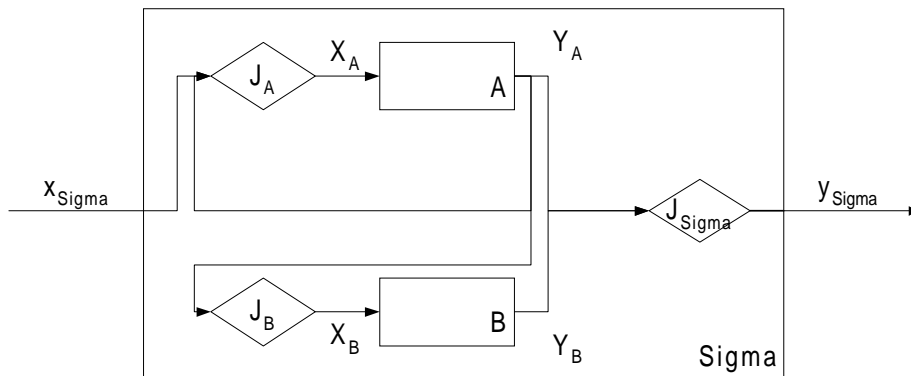


Abbildung 1.3: Ein modulares System mit zwei Komponenten

Mit I_k werden diejenigen Teilsysteme referenziert, die M_k beeinflussen. Die Schnittstellenabbildung ordnet jeder Systemkomponente k die sie beeinflussenden Größen zu. Die Fallunterscheidung (1.1) dient dazu, die Eingangs- und Ausgangsgrößen des Gesamtsystems von den einzelnen Teilkomponenten zu unterscheiden: Handelt es sich um das Gesamtsystem, ist es eine Schnittstelle zur Systemumwelt, sonst zu einer Komponente.

Am Beispiel von Abbildung 1.3 wird ein modulares System Sigma mit zwei Komponenten A und B gezeigt. Dabei symbolisieren die Rauten die Schnittstellenabbildungen J_k . So fasst J_A die Eingänge für A X_{Sigma} und Y_A zusammen. So sind z.B. die A beeinflussenden Komponenten A selbst sowie das Gesamtsystem Sigma .

1.6 Abbildungen von Systemen auf Systeme

Modellbildung besteht im Wesentlichen darin, von einem Systembegriff auf einen verwandten zu schließen. Daher ist sinnvoll, auf der Ebene der formalen Definitionen festzulegen, wie man verschiedene Systeme aufeinander abbilden kann. Bei komplexen Systembegriffen werden die formalen Definition unhandlich; die dahinter stehenden Gedanken sollten allerdings immer parat sein.

Im Folgenden seien S und S' zwei Systeme. Dabei soll S' das „kleinere“ sein in dem Sinne, wie er im Anschluss an Definition 1.9 eingeführt wird.

Definition 1.8 (Codierer, Decodierer). Seien S und S' Systeme, Ξ , Υ und Ξ' , Υ' die zugehörigen Signorräume.

1. Man nennt eine Abbildung $c : \Xi' \longrightarrow \Xi$ einen CODIERER.
2. Eine surjektive Abbildung $d : \Upsilon \longrightarrow \Upsilon'$ heißt DECODIERER.

Codierer c dienen dazu, den Eingang eines Systems S' in die „Sprache“ eines anderen Systems S zu übersetzen, während Decodierer d die Ausgabe von S wieder zurück nach S' übertragen, wie Abbildung 1.4 zeigt.

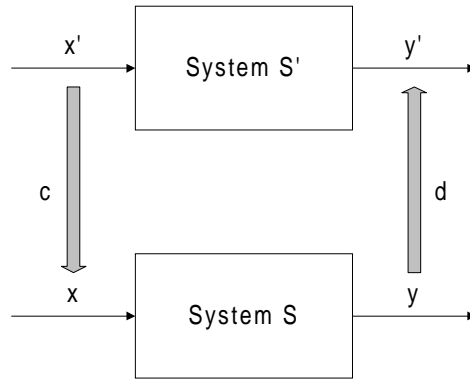


Abbildung 1.4: Codierer und Decodierer zweier Systeme

Definition 1.9 (Relationales System-Morphismus). *Es seien $S = (R, \Xi, \Upsilon, \mathbb{T})$ und $S' = (R', \Xi', \Upsilon', \mathbb{T})$ Systeme.⁵ Ein Paar (c, d) aus Codierer und Decodierer heißt RELATIONALES SYSTEM-MORPHISMUS, falls*

$$\forall (v', \xi') \in R' \exists (v, \xi) \in R : v = c(v') \wedge d(\xi) = \xi'.$$

In diesem Sinne ist das System S' „kleiner“ als das System S : Alle möglichen Relationen des kleinen Systems $(v', \xi') \in R'$ müssen sich mit Hilfe von Codierer und Decodierer auch im großen System wiederfinden. In diesem Zusammenhang bezeichnet man auch das System S als SIMULATIONSSYSTEM von S' , da es dessen vollständiges Verhalten simulieren kann.

Beim Übergang zur Definition auf dynamische Systeme, muss auch der interne Zustand des System in Betracht gezogen werden. Hierzu definiert man:

Definition 1.10 (Zustandsabbildung). *Es seien zwei dynamische Systeme $S = (\mathbb{T}, Z, \Xi, \Upsilon, F, g,)$ und $S' = (\mathbb{T}, Z', \Xi', \Upsilon', F', g',)$ mit den Zuständen Z und Z' gegeben. Dann heißt eine Abbildung $h : \bar{Z} \rightarrow Z'$ mit $\bar{Z} \subseteq Z$ ZUSTANDSABBILDUNG.*

Mit Hilfe der Zustandsabbildung wird geklärt, welche Zustände im „simulierbaren Bereich“ des Simulationssystems liegen.

Definition 1.11 (Dynamisches System-Morphismus). *Seien $S = (\mathbb{T}, Z, \Xi, \Upsilon, F, g,)$ und $S' = (\mathbb{T}, Z', \Xi', \Upsilon', F', g',)$ zwei dynamische Systeme. Dann heißt das Tripel Codierer, Zustandsabbildung und Decodierer (c, h, d) ein DYNAMISCHES SYSTEM-MORPHISMUS, falls $\forall z \in \bar{Z}, \xi' \in \Xi'$ und $t \leq t'$ gilt:*

$$F'(t, t', h(z), \xi') = h(F(t, t', z, c(\xi'))), \quad (1.2)$$

$$g'(t, h(z)) = d(g(t, z)). \quad (1.3)$$

Diese beiden Bedingungen sichern die Konsistenz der Überführungs- und der Ausgabefunktionen.

⁵Der Einfachheit halber wird angenommen, dass die Zeitskalen der Systeme identisch sind.

Um einen Morphismus-Begriff für modulare Systeme zu entwickeln, sind einige Vorarbeiten notwendig. Probleme entstehen hier durch die Abbildung der Kopplungen zwischen den einzelnen Blöcken. Insbesondere sollen Verbindungen im kleinen System auch im großen vorhanden sein.

Der Ausgangspunkt ist hier eine Abbildung $komp : K \longrightarrow K'$, die den Blöcken des großen Systems, entsprechende Blöcke im kleinen System zuordnet. Sie nennen wir BLOCK-ABBILDUNG. Für einen Block $k' \in K'$ im kleinen System bezeichnet $K_{k'} = komp^{-1}(k')$ das Urbild von k' : die Komponenten aus dem großen System, die auf die Komponente k' im kleinen System abgebildet werden. Damit kann die geforderte Strukturtreue, dass Verbindungen im „kleinen“ System sich auch im großen finden, formal formuliert werden:

1. Eingangs-Strukturtreue: $\forall k' \in K' \forall i' \in I'_{k'} \exists k \in K \exists i \in K_{i'} : i \in I_k$.
2. Ausgangs-Strukturtreue: $\forall k' \in K' \forall e' \in E'_{d'} \exists k \in K \exists e \in K_{e'} : i \in E_k$.

Die zu entwickelnde globale Zustandsabbildung lässt sich mit Hilfe von lokalen Zustandsabbildungen für die Teilblöcke der einzelnen Komponenten formulieren:

Definition 1.12 (Block-Zustands-Abbildung). Sei S ein modulare System mit K Blöcken. Eine Abbildung

$$h : \bigotimes_{k \in K} Z_k \longrightarrow \bigotimes_{k' \in K'} Z'_{k'} \quad , \text{wobei } h = (h_{d'_1}, \dots, h_{d'_n}),$$

nennen wir BLOCK-ZUSTANDS-ABBILDUNG. Dabei werden die einzelnen Komponenten der Abbildung lokal definiert als Zustandsabbildung auf die Komponente k' :

$$h_{k'} : \bigotimes_{k \in K_{k'}} Z_k \longrightarrow Z'_{k'}$$

Die Definition der Ein- und Ausgänge erfolgt derart, dass für sie deren lokale Blockzustandsabbildung herangezogen wird:

$$h_{I'_{d'}} : \bigotimes_{i \in \bigotimes_{i' \in I'_{d'}} Z_i} Z_i \longrightarrow \bigotimes_{i' \in I'_{d'}} Z'_{i'}$$

$$P_{i'} (h_{I'_{d'}}(\dots, z_i, \dots)) = h_{i'}(\dots, z_i, \dots) \quad .^6$$

Definition 1.13 (Modulares System-Morphismus). Seien S und S' zwei modulare Systeme. Dann nennt man ein Paar $(komp, h)$ aus einer Block- und einer Block-Zustands-Abbildung einen MODULARES SYSTEM-MORPHISMUS, falls

1. Zustandsbedingung: $F' (t, t', h_{I'_{d'}}(\dots, z_i, \dots), \xi) = h_{E'_{d'}} \left(\bigotimes_{d \in D_{d'}} F (t, t', z_{|I_d}, \xi) \right)$.
2. Ausgangsbedingung: $g'_{d'} \left(t, h_{I'_{d'}}(\dots, z_i, \dots) \right), \xi = \bigotimes_{d \in D_{d'}} g (t, z_{|I_d}, \xi)$.

⁶Dabei bezeichnet P_j die Projektion auf die j -te Komponente.

Die Zustandsbedingung besagt, dass die Zustandsüberföhrungsfunktion im kleinen System des Blocks d' , angewandt auf die Bilder der d' beeinflussenden GröÖen, sich ergibt als das Bild der Block-Zustandsabbildung der AusgangsgröÖen, die auf d' abgebildet werden, angewandt auf den zugehörigen Zustand im großen System.

Kurz gesagt, wird dieser Morphismus so gebildet, dass der Betrag der Eingänge dem Betrag der Ausgänge der Komponenten, die auf d' abgebildet werden, entspricht. Analog besagt die Ausgangsbedingung, dass der Ausgang im kleinen System dem im großen System entsprechen muss. Darüber hinaus müssen die Bedingungen zur Strukturtreue eingehalten werden.

Kapitel 2

Modelle, Simulation und Modellentwicklung

Modellbildung lebt vom Dualismus zwischen System und Modell. Im vorigen Kapitel sind verschiedene Systembegriffe eingeführt worden. Sie bilden den Ausgangspunkt für einige grundsätzliche Bemerkungen über Modelle, Simulation und Modellentwicklung.

2.1 Modelle

Am Anfang der Betrachtungen über Systeme hat das Objekt gestanden, das vom Subjekt erkannt wird. Darauf aufbauend sind verschiedene Systemdefinitionen aufgebaut worden.

2.1.1 Begriffsbildung

Für die Modellbildung ergibt sich daraus eine dreifache Unterscheidung:

1. **Systemobjekt:** Das Objekt, das wahrgenommen wird und modelliert werden soll. Oft meint man mit System auch das so genannte Systemobjekt.
2. **Systemmodell:** Im Grunde beschäftigt sich diese gesamte Arbeit damit, Systemmodelle auf verschiedenen Ebenen zu erstellen. Eine genauere Abgrenzung kann nur über das Herausarbeiten von verschiedenen Aspekten erfolgen:
 - (a) *Vorstellung:* Es besteht ein Unterschied zwischen dem, was ist, und dem, was man sich darunter vorstellt. Auf der „niedrigen“ Wahrnehmungsebene soll das Systemmodell vom Systemobjekt abgegrenzt werden.
 - (b) *Reflexion:* Auch wenn man sich darüber im Klaren ist, dass die Vorstellung von einem Systemobjekt nicht das Objekt selbst ist, hat man noch nicht den Bereich der wissenschaftlichen Reflexion betreten. Erst wenn versucht wird, das Objekt in Modellen für andere (intersubjektiv) nachprüfbar zu erklären oder zu beschreiben, soll von einem Systemmodell gesprochen werden.
 - (c) *Formalisierung:* Systemmodelle sollen auch nach dem Grad ihrer Formalisierung unterschieden werden. Dies geschieht vor allem in Hinblick darauf, dass am

Ende der Überlegungen stets ein auf einem Rechner simulierbares Modell steht. Es soll also zwischen dem wissenschaftlich reflektierten, dem zur Simulation formal aufbereiteten Systemmodell und dessen Realisierung als Simulationsmodell unterschieden werden.

Aufgrund dieser Abgrenzungen soll folgende Vereinbarung getroffen werden:

Vereinbarung 2.1 (Systemmodell). *Unter einem SYSTEMMODELL soll ein in der Wissenschaft entwickeltes, intersubjektiv nachprüfbares, mehr oder weniger formalisiertes, ein Systemobjekt erklärendes oder beschreibendes Modell verstanden werden.*

3. **Systemdefinitionen:** Die im vorigen Kapitel gegebenen Systemdefinitionen sind selbst Modelle der Struktur von Systemen. Sie dienen in der Systemtheorie als Raster, mit dem verschiedene Systeme beschrieben werden können.¹

Auch wenn sich in der Wissenschaft zu einem Systemobjekt ein bestimmtes Systemmodell etabliert hat, kann niemals von dessen Richtigkeit gesprochen werden. Vielmehr kann es nur als vorläufig, d.h. als mit dem gegenwärtigen Wissensstand nicht widerlegt gelten.² Dennoch ist es in diesem Fall müßig, das Objekt ohne konkreten Anlass vom Systemmodell und dessen Definition zu unterscheiden. Dann ist die Formulierung des Systems in systemtheoretischen Begriffen höchstens von theoretischem Interesse. Nur im umgekehrten Fall, im Zuge einer Modellbildung, ist die Unterscheidung wesentlich: Die allgemeinen Systemdefinitionen können als Leitfaden zur Bildung eines Systemmodells des Systemobjekts dienen.

Unter einem System versteht man stets einer der oben genannten Punkte. Es wird nur zwischen ihnen unterscheiden, wenn es in zuvor genanntem Sinne notwendig wird, die Begriffe genau zu klären. Damit können die Erkenntnisse aus den Systemdefinitionen aus Kapitel 1 — als ein Modell von Systemen — benutzt werden, um grundlegende Aspekte von Modellen einzuführen: Nach der Systemdefinition von Bossel in Kapitel 1.3 muss ein System einen Zweck erfüllen, eine Wirkungsstruktur haben und soll nicht teilbar sein. Überträgt man dies auf einen zu entwickelnden Modellbegriff, so ist die Forderung sinnvoll, für ein Modell zunächst einen Zweck festzulegen. Man sollte sich im Klaren darüber sein, wozu ein Modell erstellt wird. Ebenso kann die Forderung, dass ein System nicht teilbar sein soll, auf Modelle übertragen werden. Im Sinne des festgelegten Zwecks sollte das Modell schon aus ökonomischen Gründen möglichst einfach sein und keine unnötigen Komponenten enthalten. Da es sich bei einem Modell nur um ein Abbild der Realität handelt, kann folgende Vereinbarung getroffen werden:

Vereinbarung 2.2 (Modell). *Ein MODELL ist eine vereinfachende Abbildung eines Systems, das einem bestimmten Zweck dient und selbst möglichst einfach ist.*

In diesem Zusammenhang ist auch der Abschnitt über die Systemmorphismen 1.6 zu sehen. Betrachtet man ein System hier als eine Systemdefinition, liefert ein Morphismus eine Abbildung des Systems in eine passende „Modell-Sprache“.

¹Beispiele hierfür findet man in [Pic75].

²Auf den Aspekt der Falsifizierung kommen wir in 4.1 zurück.

2.1.2 Der Modellzweck

Zur weiteren Bestimmung des MODELLZWECKS werden die in Kapitel 1 eingeführten Systemdefinition genauer betrachtet.

Der Begriff der Systemumwelt veranschaulicht in diesem Zusammenhang, dass ein Modell stets in einem Kontext entsteht und in einem Umfeld eingesetzt wird. Für die Erstellung des Modells ist also darauf zu achten, dass es zur Verwendung in diesem Kontext geeignet ist; dies entspricht der Betrachtung von EINSATZMODELLEN im Rahmen des Software Engineerings, die im PFLICHTENHEFT dokumentiert werden.³ Wie bei der Unterscheidung zwischen der Systemumwelt und dem Experimentierfeld (vgl. Kapitel 1.3) kann auch hier der Gesamtkontext in der Definition eines Obersystems münden. Die Sichtweise ist entscheidend. Auf jeden Fall muss bei der Prüfung, ob es sich um ein gültiges Modell handelt, die Einsatzfähigkeit in seinem Kontext getestet werden. In diesem Sinne ist der Kontext auch Teil des Modells.

Die relationale Systembetrachtung mit ihrer Beschreibung des Ein- und Ausgabeverhaltens führt zunächst wieder zu einer Parallele zum Software Engineering. Dort ist die Erstellung des FUNKTIONSMODELLS Teil der Anforderungsdefinition. Es gibt an, welche Funktionen der Anwender der Software auslösen kann und wie deren Wirkungsweise ist. In der Sprache der Systeme aus dem vorigen Kapitel heißt dies, dass angegeben wird, mit welchen Ausgangssignalen das System Software auf die Eingangssignale des Benutzers antwortet. Im Zusammenhang mit der Modellbildung gilt es, den grundsätzlichen Dualismus zwischen System und Modell im Auge zu behalten.

Wendet man die relationale Sichtweise auf das Systemobjekt an, so beschreibt das FUNKTIONSMODELL DES SYSTEMS, ob und von welchen Signalen das Objekt gesteuert wird und welche Werte man als Ausgabe messen kann. Hier kann die Systemumwelt wiederum als Experimentierfeld interpretiert werden, in der ein Experimentator Systemsteuerung und Ausgabe überwacht. Für die Erstellung einer Anforderungsdefinition heißt dies insbesondere, dass man, ohne das System zu modellieren, sich einen Überblick über die Datenlage verschaffen sollte: Welche Daten stehen in welcher Güte zu welchem Zeitpunkt zur Verfügung?

Ebenso lässt sich die relationale Sichtweise auf das zu erstellende Modell anwenden: Es muss festgelegt werden, welche Ausgaben das Modell liefern soll und welche Eingaben gemacht werden müssen bzw. aufgrund der Datenlage gemacht werden können. Da diese Systemformulierungen für das Systemobjekt und das Modell im Allgemeinen im Sinne von Codierung und Decodierung gemäß Definition 1.8 nicht deckungsgleich sind, sprechen wir hier vom FUNKTIONSMODELL DES MODELLS.

Des Weiteren ist im Software Engineering im Funktionsmodell anzugeben, welche Funktionen der Benutzer aufrufen kann. Dies gilt sinngemäß für alle Modelle: es muss festgelegt werden, welche Algorithmen ein Modell oder dessen Steuerung durchführen können.

Im nächsten Schritt, der Betrachtung eines SYSTEMS ALS FUNKTION, sind die Anfangsbedingungen in Betracht zu ziehen. Übertragen auf Modelle heißt das, dass eine INITIATION stattfinden muss.

³Hierzu vergleiche man die gängige Literatur zum Software Engineering, etwa [Bal96].

Der Übergang zum Begriff des dynamischen Systems liefert eine in der Modellbildung prinzipielle Unterscheidung. Hier entscheidet sich, ob die interne Struktur des Systems mit in Betracht gezogen werden soll: Wird versucht, aus der BLACKBOX eine WHITEBOX zu machen? Dies ist gleichbedeutend mit der Frage, ob man STRUKTURTREUE zwischen Modell und System erreichen will oder sich mit VERHALTENZAUSSAGEN zufrieden gibt. Im letzteren Fall führt dies dazu, dass das Modell (streng genommen) nur für Daten gültig ist, die aus dem selben Zeitbereich stammen wie die zur Erstellung verwendeten Daten. Umgekehrt impliziert ein strukturtreues Modell, dass es auch für zukünftig anfallende Daten des Systems gültig ist, sofern sich das System nicht gewandelt hat.

Schließlich demonstriert der Übergang zum modularen Systembegriff, dass bei der Modellentwicklung zwischen Black- und Whitebox kombiniert werden kann und man dies oft auch muss. Für jede Komponente gilt dann das Gesagte. Insgesamt entsteht eine GREYBOX, und bei Gültigkeitsaussagen ist noch größere Vorsicht geboten. Jede Komponente kann für sich geeignet, aber deren Interaktion falsch modelliert worden sein. Für die Formulierung eines Modellzwecks folgt aus den zuletzt genannten Punkten, dass potenzielle Modellkomponenten und deren Abhängigkeiten frühzeitig benannt werden sollten. Ein weiterer Grund hierfür ist, dass man am Ende der Modellbildung vorgenommene Vereinfachungen benennen kann, die durch Ignorieren der potenzieller Modellkomponenten zustande gekommen sind.

2.2 Simulation

Modelle werden nicht als reiner Selbstzweck entwickelt. Einer ihrer Vorteile ist, dass sich deren Verhalten leichter studieren lässt als das des Systems. Umgangssprachlich bezeichnet man das Erzeugen von Modellverhalten als SIMULATION. Im Abschnitt 1.6 haben wir gesehen, wie man den Begriff eines SIMULATIONSMODELLS formal fassen kann. Hierzu ist das „größere“ System als Modell des kleineren aufzufassen, so dass Ersteres alle Ein- und Ausgaben von Letzterem simulieren kann. Durch die weite Verbreitung von Rechnern versteht man heute meist unter einem Simulationsmodell eine passende Implementation eines Modells auf einem Computer, und Simulation bedeutet das Ausführen der zugehörigen Berechnungen.

2.2.1 Simulationstypen

Man unterscheidet verschiedene Typen von Simulationen. Da hier die gesamte Theorie der Systeme zeitbezogen entwickelt wird, steht die ZEITORIENTIERTE SIMULATION, deren grundsätzliches Ablaufschema in Abbildung 2.1 links dargestellt ist, im Vordergrund. Sie wird von einer Simulationsuhr gesteuert, die zunächst auf Null gestellt wird. Nach einer Initialisierung werden solange Berechnungen durchgeführt, bis der festgelegte Zeitbereich, die SIMULATIONSDAUER, durchlaufen ist. Zur Initialisierung sind mindestens zwei Schritte nötig: Zunächst müssen die Startbedingungen des Modells festgelegt werden, zum anderen müssen die Parameter, die vor jeder Simulation eingestellt werden sollen, überprüft werden. Diese sollen SIMULATIONSPARAMETER genannt werden.

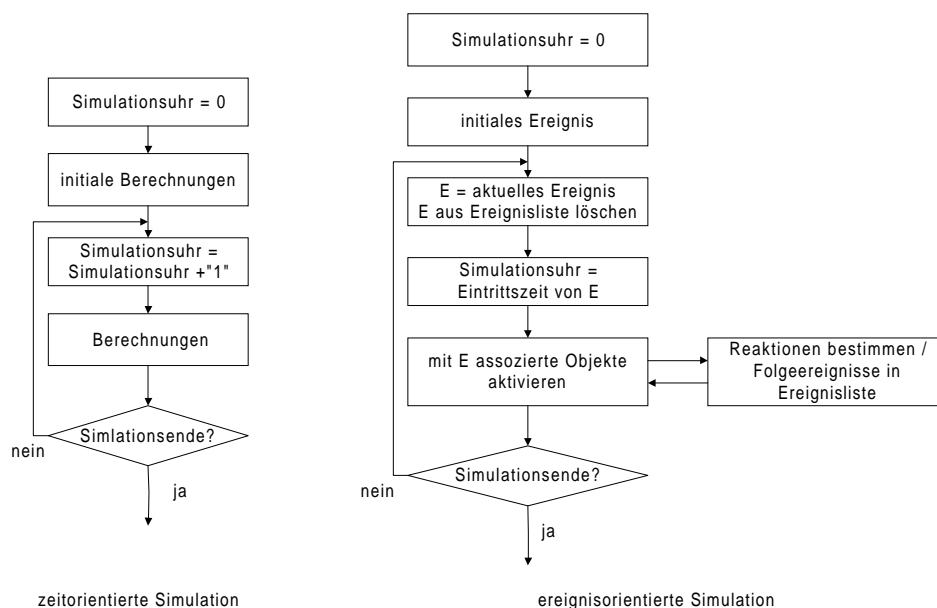


Abbildung 2.1: Schema der zeitorientierten und ereignisorientierten Simulation

4

In diesem Zusammenhang sind einige Bemerkungen zur Definition der Simulationsuhr notwendig: Zunächst ist die Frage zu beantworten, in welcher Zeiteinheit die Simulation stattfinden und wie grob das notwendig diskrete Zeitraster sein soll. Erst nach diesen Festlegungen ist der Befehl „Simulationsuhr = Simulationsuhr + 1“ eindeutig. Damit sind allerdings noch nicht alle Fragen in Zusammenhang mit der Zeitskala beantwortet. In die Systemdefinition ist die Zeitskala mit aufgenommen worden, insbesondere ist die Skala, in der das System beschrieben wird, eine definitorische Festlegung. Allerdings gibt es drei Punkte beim Übergang von der „System-Zeit“ zur „Simulations-Zeit“ zu bedenken:

1. In welchem Zeitbereich findet das tatsächliche Verhalten des Objekts statt?
2. Sind mehrere Skalen für uns von Interesse?
3. In welchen Zeitskalen liegen Beobachtungsdaten vor?

Diese Fragen gilt es, bei der Festlegung der Zeitskala für die Simulation im Auge zu behalten. Wir werden im Kapitel 5.2.3 auf die Modellierung der Zeit zurückkommen.

Der Begriff der Zeitskala ist in den meisten Fällen ein vom Beobachter dem System hinzugefügter Begriff. Kaum ein Systemobjekt wird sich verändern, nur weil eine gewisse Zeitspanne abgelaufen ist. Vielmehr entsteht Dynamik im Objekt eher ereignisorientiert. Dies hat zu EREIGNISORIENTIERTEN SIMULATIONEN geführt. Sie laufen grundsätzlich nach dem Schema rechts in Abbildung 2.1 ab: Dynamik entsteht dadurch, dass Objekte, die mit dem aktuellen Ereignis assoziiert sind, aktiviert werden, so neue Ereignisse entstehen und

⁴Die Abbildungen folgen [Sau99] S. 31 und 33.

nacheinander abgearbeitet werden. Zeit ist hier nur als ein Attribut der Ereignisse vorhanden. In Kapitel 3 wird eine Klasse von Modellen vorgestellt, die sowohl zur ereignis- als auch zeitorientierten Simulation geeignet ist.

2.2.2 Simulation zur Prognose

Gleichgültig um welchen der beiden vorgestellten Simulationstypen es sich handelt, ihnen ist gemeinsam, dass der Einfluss auf das Systemobjekt sowie dessen späteres Verhalten per Simulation nachgebildet wird. Abbildung 2.2 zeigt dies schematisch. Der zentrale Punkt ist, dass am Modell späteres Systemverhalten vorhergesagt wird, das Modell als PROGNOSE-MODELL eingesetzt wird. Es lassen sich SZENARIEN („in sich konsistente und plausible Parameterkonstellationen“⁵) durchspielen. Handelt es sich zusätzlich um ein Systemobjekt, das durch Eingreifen von außen beeinflusst werden kann, erhält man so Handlungsoptionen zu dessen Manipulation. In diesem Sinne können Modelle als ENTSCHEIDUNGSHILFE dienen.

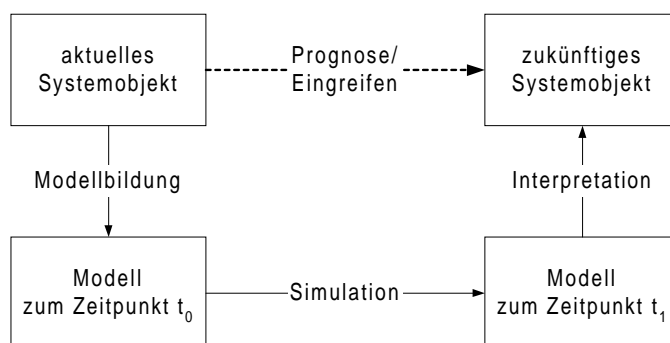


Abbildung 2.2: Simulation zur Prognose und Entscheidungshilfe

6

2.3 Modellentwicklung

Bisher stellt sich das Gesagte wie in Abbildung 2.3 dar: Ausgehend vom Systemobjekt wird mit Hilfe der systemtheoretischen Systemdefinitionen ein Systemmodell erstellt, das in ein Simulationsmodell überführt wird. Dabei muss mindestens in komplexeren Fällen das in den Fachwissenschaften vorhandene Systemmodell in eine für einen Rechner geeignete Form gebracht werden. Dies ist die Stelle, an der zwischen mehr oder weniger formalisierten Systemmodellen unterschieden wird, die Informatik in diesem Gesamtkontext ihren Platz hat und auch diese Arbeit thematisch anzusiedeln ist. Zusammen mit dem Fachwissenschaftler gilt es für den Informatiker, Entwicklungsschritte auf verschiedenen Ebenen durchzuführen.

⁵Vgl. [Bos94] S. 211.

⁶Die Abbildung lehnt sich an [Sau99] S. 9 an.

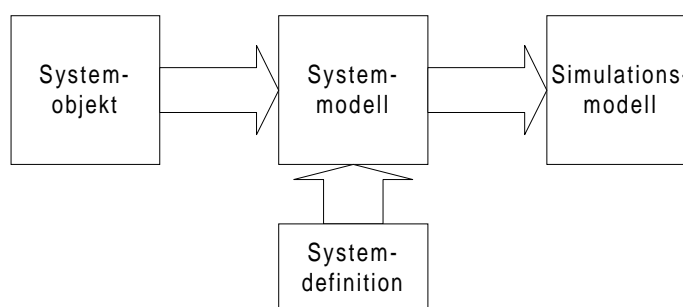


Abbildung 2.3: Der Übergang vom System zum Simulationsmodell

2.3.1 Ebenen eines Modells

Konzeptionell hilft bei diesem Vorgang wieder eine Parallele zum Software Engineering: Der STANDARDENTWURF, der in Abbildung 2.4 für die allgemeine Modellbildung formuliert ist.

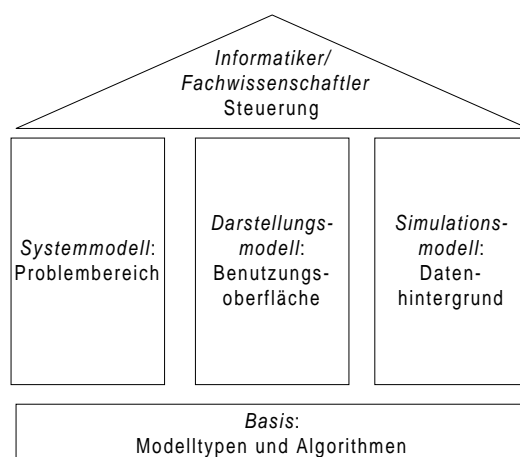


Abbildung 2.4: Der Standardentwurf übertragen auf die Modellbildung

Die Parallele zwischen Problembereich und Systemmodell ist offensichtlich. Die Organisation des Datenhintergrunds auf dem Rechner korrespondiert mit dem Simulationsmodell. Es stellt sich im Allgemeinen als eine konkrete, auf einem Rechner implementierte Struktur dar, die sowohl die Daten der Modellstruktur als auch die simulierten Ergebnisse verwaltet. Im Zusammenhang der allgemeinen Modellbildung steht das Darstellungsmodell oder die Benutzungsoberfläche im Software Engineering im Mittelpunkt. Hier wird eine zumeist graphische Darstellungsform gesucht, in der das Wissen aus dem (noch nicht formalisierten) Systemmodell in eine simulierbare Sprache für das Simulationsmodell überführt wird. In der Basis findet sich der „Werkzeugkasten“ für das System- und das Simulationsmodell. Sie bilden zusammen das Gesamtmodell, das vom Fachwissenschaftler und Informatiker verwaltet wird.

Die Darstellung ist dem Software Engineering entlehnt und ist beispielhaft in verschiedenen SIMULATIONSWERKZEUGEN implementiert. In ihnen verwendet man eine festgelegtes Darstellungsparadigma, das in einer Benutzungsoberfläche implementiert ist, um ein

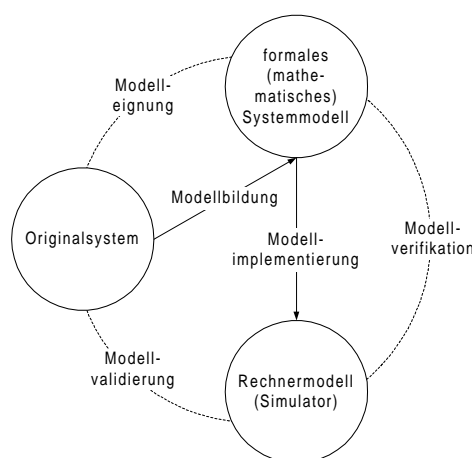


Abbildung 2.5: Das Grundmodell der Modellbildung

Systemmodell in ein Simulationsmodell zu übertragen. Die interne Datenverwaltung wird vom Simulationsprogramm übernommen. Der Standardentwurf hilft festzustellen, auf welchen Modellebenen an dem zu erstellenden Gesamtmodell gearbeitet werden muss.

Der Prozess des Modellierens wird nun genauer betrachtet, indem verschiedene Lebenszyklen als Entwicklungsmodelle vorgestellt werden. An dieser Stelle geht es nicht um richtige oder falsche Entwicklungsmodelle. Sie alle enthalten wesentliche Aussagen und fokussieren ausgewählte Aspekte. Daher werden diese Lebenszyklen ohne Wertung nebeneinander gestellt.

2.3.2 Grundmodell

Zunächst wird in Anlehnung an [Sau99] ein GRUNDMODELL angegeben. Die Abbildung 2.5 zeigt zwei grundlegende Tätigkeiten: In der Phase der Modellbildung findet der Übergang vom Originalsystem zum formalen (mathematischen) Systemmodell statt. Dieses wird durch die Modellimplementierung in ein Rechnermodell überführt, das die Simulationsergebnisse liefert. Die jeweiligen Revisionsprozesse zur Kontrolle der Schritte sind an den gestrichelten Linien angegeben: Bei der Modelleignung muss die Frage beantwortet werden, ob das Modell überhaupt die gewünschten Daten in akzeptabler Genauigkeit liefern kann. In der Modellverifikation wird überprüft, ob „das Rechnermodell wirklich exakt dem formalen Modell entspricht.“⁷ Der Schritt der MODELVALIDIERUNG überprüft die „inhaltliche Korrektheit“. Wir werden auf die Begriffe Validierung und Verifikation im zweiten Teil in Kapitel 6 zurückkommen. Über die hier gemachte Unterscheidung besteht allerdings Konsens in der Literatur: Verifikation bezieht sich darauf, ob der Prozess des Übergangs von einem Modell zum anderen korrekt stattgefunden hat. Validität bedeutet dagegen, dass das Resultat, das erstellte Modell, im Vergleich zum realen System, das Gewünschte leistet. Zur Unterscheidung dient das folgende Wortspiel:

| | |
|---------------|---|
| Validierung: | Haben wir das richtige Modell gebildet? |
| Verifikation: | Haben wir das Modell richtig gebildet? |

⁷Vgl. [Sau99] S. 8.

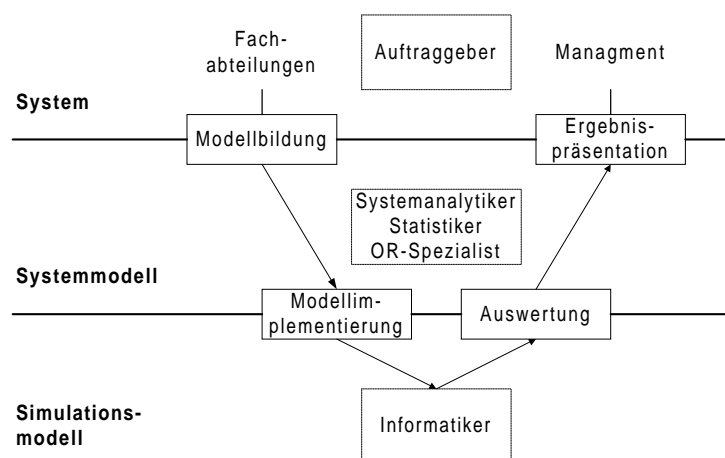


Abbildung 2.6: Das Ebenenmodell der Modellbildung

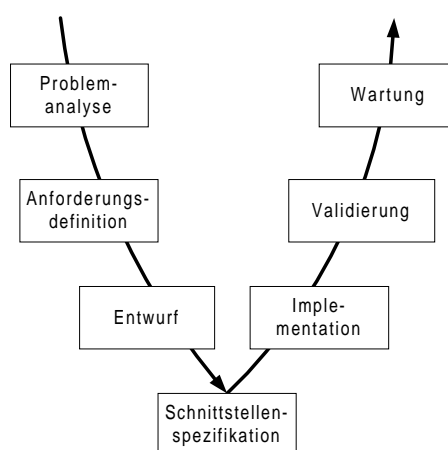


Abbildung 2.7: Software-Lebenszyklus: Das V-Modell

2.3.3 Ebenenmodell

Ausgangspunkt für das folgende Ebenenmodell ist die getroffene Unterscheidung zwischen Systemobjekt, Systemmodell und Simulationsmodell. Diese finden sich in der Abbildung 2.6 wieder, die an [Sau99] S. 15 angelehnt ist. Wie im Grundmodell ist der Übergang vom Realproblem zum Systemmodell die Phase der Modellbildung. Auf dieser Ebene arbeiten unter anderem Systemanalytiker. Zum Simulationsmodell kommt man durch dessen Implementierung durch den Informatiker. Die Phasen Modellverifikation und Modellvalidierung werden hier unter dem Aspekt konkretisiert, dass Simulationsergebnisse ausgewertet werden müssen und bei Eignung präsentiert werden können. Diese Phasen finden wiederum auf den höheren Ebenen statt.

2.3.4 Software-Lebenszyklen

Die Ähnlichkeit des Ebenenmodells mit dem bekannten V-MODELL-LEBENSZYKLUS im Software Engineering ist nicht zufällig. Die Parallele der beiden Modelle ist so zu zie-

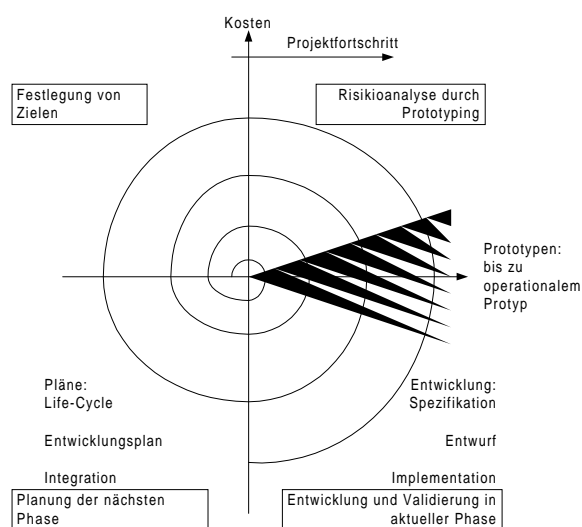


Abbildung 2.8: Software-Lebenszyklus: Das Spiralmodell

8

hen, dass Modellbildung und Modellimplementation im Software Engineering durch Anforderungsdefinition und Entwurf zu einem Systemmodell im Rahmen der entstandenen Dokumente führen und in der Implementation und Installation der Auswertung und Ergebnispräsentation entsprechen: Ob das Systemmodell des Entwurfs geeignet ist, zeigt sich in der Auswertung der Implementation, die das Systemmodell auf dem Rechner simuliert.

Idealtypisch sind die zuletzt vorgestellten Lebenszyklen Prozesse ohne Revisionen. Abschließend wird mit dem SPIRALMODELL ein Lebenszyklus vorgestellt, der diese offensichtliche Unzulänglichkeit korrigiert. Gemäß dem Spiralmodell 2.8 läuft der Prozess des Software Engineerings entlang der spiralförmig nach außen verlaufenden Linie. Die Achsen tragen die aufgelaufenen Kosten gegen den Projektfortschritt ab. Der Projektverlauf folgt jeweils den gleichen Schritten, die sich je nach Projektstand in unterschiedlichen Konkretisierungsstufen zeigen:

1. Planung der nächsten Phase
2. Festlegung von Zielen
3. Risikoanalyse durch Prototyping
4. Entwicklung und Validierung in der aktuellen Phase

Für die Modellbildung im Allgemeinen ist der Begriff des PROTOTYPING von Interesse. Grundidee des Prototyping ist, die gemachten Anforderungen in einem Prototyp zu realisieren und sie zusammen mit dem Auftraggeber zu überprüfen und zu konkretisieren, um ggf. später teure Revisionsprozesse zu verhindern.⁹ Dahinter steckt die Erkenntnis, dass die Güte eines Modells am besten am Modellverhalten zu überprüfen ist, selbst wenn das Modell noch nicht exakt das Gewünschte leistet.

⁸Die Abbildung an [PB93] S. 27 angelehnt.

⁹Zum Prototyping vergleiche man Lehrbücher wie [PB93] und [Bal96].

Kapitel 3

Netzmodelle

In Anlehnung an den Standardentwurf ist festgestellt worden, dass ein Modell prinzipiell aus den Komponenten Systemmodell, Simulationsmodell und Darstellungsmodell besteht. Grundlage eines Gesamtmodells ist das Basiswissen aus Methoden, Algorithmen und Modelltypen. Ferner zeigen die Lebenszyklen, dass Modellbildung ein Prozess ist, den man strukturieren kann.

Im Folgenden wird die Klasse der sog. NETZMODELLE behandelt, die man auf verschiedenen Ebenen der Modellbildung einsetzen kann. Diese folgen einem gemeinsamen Darstellungsparadigma. Aufgrund ihrer Syntax als bipartite Graphen sind sie ein Hilfsmittel bei der strukturellen Modellierung des Systems. Auf dieser statischen Struktur lassen sich mit Hilfe von Markierungen verschiedenen Bedeutungen und in der Folge unterschiedliche Dynamiken definieren, die entweder ereignis- oder zeitorientiert simuliert werden.

3.1 Strukturnetze

Sieht man von jedweder Bedeutung der Netze ab, so bleibt nur deren Struktur zu untersuchen.

Definition 3.1 (Netz). *Ein Tripel $N = (Z, E, F)$ heißt NETZ, falls gilt:*

1. Z und E sind disjunkte Mengen.
2. $F \subset (Z \times E) \cup (E \times Z)$ ist eine zweistellige Relation, die sog. FLUSSRELATION, von N .

Das Darstellungsmodell ist nach dieser Definition ein gerichteter bipartiter Graph mit zwei Sorten Z und E von Knoten. Im Graphen wechseln sich die beiden Sorten von Knoten ab. Diejenigen aus Z werden als ZUSTÄNDE und die aus E als EREIGNISSE bezeichnet. In der graphischen Darstellung werden Zustände als Kreise, Ereignisse als Quadrate oder Rechtecke und die Flussrelation als gerichtete Kanten zwischen den Knoten symbolisiert.

Unabhängig von der weiteren Konkretisierung werden diese Knoten stets so interpretiert, dass die Ereignisse das Modell von einem Zustand in den anderen überführen. Auf diesem

Abstraktionsniveau spricht man von STRUKTURNETZEN. Aus der Darstellung als bipartiter Graph ergeben sich die Begriffe VORGÄNGER, NACHFOLGER und UMGEBUNG in natürlicher Weise. Wir verzichten auf die einzelnen Definitionen.

Einzig die graphentheoretische Begriffe QUELLEN bzw. SENKEN als Knoten ohne Vorgänger bzw. Nachfolger sind von Interesse. Da mit Netzen Prozesse modelliert werden sollen, erscheint es sinnvoll, diese mit einem Ereignis beginnen und mit einem abschließenden Ereignis enden zu lassen. Diese Ereignisknoten stellen die Schnittstelle zur Umwelt dar. Quellenereignisse sind demnach exogene Ereignisse, die in einem Netz Dynamik von außen erzeugen. Senkenereignisse dagegen haben eher definitiven Charakter, sie geben die Ausgänge des Netzes zu seiner Umwelt an. Im Graphen wird man ggf. den Zustand der gesamten Systemumwelt in einem Quellen- oder Senkenknoten symbolisieren.

Schon als reine Struktur sind Netze ein geeignetes Hilfsmittel der Modellbildung. Denn aufgrund ihrer Darstellung als bipartite Graphen ergibt sich die Möglichkeit, diese Modelle syntaktisch korrekt zu konkretisieren, so dass sich schon hier Hinweise auf mögliche Modellierungsfehler finden. Die entscheidenden Begriffe sind in diesem Zusammenhang:

Regel 3.1 (Verfeinern).¹

1. In einem Netz N wird eine Stelle s durch das Netz M zum Netz N' verfeinert, wenn M anstelle von s so eingesetzt wird, dass für jeden Pfeil $x \rightarrow y$ von M nach N' (bzw. $x \leftarrow y$ von N' nach M) gilt:

- (a) x ist eine Stelle von M , und y ist eine Transition von N' .
- (b) In N gibt es einen Pfeil von $s \rightarrow y$ (bzw. $s \leftarrow y$).

2. In einem Netz N wird eine Transition t durch das Netz M zum Netz N' verfeinert, wenn M anstelle von t so eingesetzt wird, dass für jeden Pfeil $x \rightarrow y$ von M nach N' (bzw. $x \leftarrow y$ von N' nach M) gilt:

- (a) x ist eine Transition von M und y ist eine Stelle von N' .
- (b) In N gibt es einen Pfeil von $t \rightarrow y$ (bzw. $t \leftarrow y$).

Das eingesetzte Netz M muss an seinem Rand Knoten vom zum ersetzenden Typ haben und Kanten müssen entsprechend schon im ursprünglichen Netz vorgesehen gewesen sein. Durch das Verfeinern kann eine „grobe“ Modellkomponente konkretisiert werden.

Ferner kann durch Einbettungen ein Modell um bisher unberücksichtigte Komponenten erweitert werden.

Regel 3.2 (Einbetten). Ein Netz N' ist eingebettet in ein Netz N , wenn jede Stelle, jede Transition und jeder Pfeil von N' auch in N vorkommt. N' heißt dann auch AUSSCHNITT von N .

¹Vgl. [Rei85] S. 68.

Im Allgemeinen wird man Regel 3.2 derart anwenden, dass N ohne N' bereits ein „funktionsfähiges“ Netz ist, das um den Ausschnitt N' erweitert wird.

Bisher enthalten Netze nur durch Namen an den Knoten Information. In einem zweiten Schritt wird den Zustandsknoten ein quantitatives Attribut, eine MARKIERUNG, hinzugefügt.

Definition 3.2 (Markierung). Sei $N = (Z, E, F)$ ein Netz und $\mathbb{W} \in \{\{0, 1\}, \mathbb{N}_0, \mathbb{R}\}$. Eine Abbildung $M : Z \rightarrow \mathbb{W}$ heißt MARKIERUNG.

Netzmodelle können anhand der Wertebereiche der Markierungen unterschieden werden.

3.2 Bedingungs-Ereignisnetze und Stellen-Transitionsnetze

Es werden Netze mit dem Wertebereich $\{0, 1\}$ und solche mit Wertebereich in den natürlichen Zahlen zusammen vorgestellt.² Ist der Wertebereich des Netzes $\{0, 1\}$, so ist dies offensichtlich ein Spezialfall. Die Anzahl der Markierungen in den Zuständen können mit Punkten symbolisiert werden, und man spricht in diesem Zusammenhang von den MARKEN, die die Zustände tragen. Falls es sich um den Wertebereich $\{0, 1\}$ handelt, können Zustände als Bedingungen und Marken als Wahrheitswerte interpretiert werden.

Definition 3.3 (Stellen-Transitions-Netz). Ein 6-Tupel $N = (S, T, F, K, W, M)$ heißt STELLEN-TRANSITIONSNETZ (kurz: S/T-Netz), falls gilt:

1. (S, T, F) ist ein Netz aus STELLEN und TRANSITIONEN.
2. $K : S \rightarrow \mathbb{N}^\infty$ erklärt eine (möglicherweise unbeschränkte) KAPAZITÄT DER STELLEN.³
3. $W : F \rightarrow \mathbb{N}_0$ bestimmt zu jedem Pfeil des Netzes ein GEWICHT.
4. $M : S \rightarrow \mathbb{N}^\infty$ ist eine ANFANGSMARKIERUNG, die die Kapazitäten respektiert, d.h. $M(s) \leq K(s)$ für alle $s \in S$.

Handelt es sich bei einem Netz um ein S/T-Netz, werden die Zustände Z mit S für Stellen und die Ereignisse E mit T für Transitionen bezeichnet. Werden die Zustände als Bedingungen interpretiert, nennt man die Zustände B und die Ereignisse wie in der ursprünglichen Definition E und spricht von einem Bedingungs-Ereignis-Netz (kurz: B/E-Netz).

Die Kapazitäten der Stellen geben deren höchstmögliche Markenzahl an. Das Gewicht eines Pfeiles bestimmt, wie viele Marken im Falle eines zugehörigen Ereignisses entlang des Pfeiles bewegt werden.

Definition 3.4 (Aktivierung, unmittelbare Folgemarkierung). Sei $N = (S, T, F)$ ein S/T-Netz und M eine Markierung gemäß 3.2 mit Werten in \mathbb{N}_0^∞ , die die Kapazitäten der Stellen respektiert.

²Die Darstellung folgt wesentlich [Rei82] Kapitel 5.

³Dabei bezeichnet $\mathbb{N}^\infty := \mathbb{N} \cup \{\infty\}$, $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$ und $\mathbb{N}_0^\infty := \mathbb{N} \cup \{0\} \cup \{\infty\}$.

1. Eine Transition $t \in T$ heißt *M-aktiviert*, falls

(a) Für alle Vorgänger-Stellen s von t gilt: $M(s) \geq W(s, t)$.

(b) Für alle Nachfolger-Stellen s von t gilt: $M(s) \leq K(s) - W(t, s)$.

2. Eine *M-aktivierte* Transition $t \in T$ bestimmt eine UNMITTELBARE FOLGEMARKIERUNG M' VON M durch

$$M'(s) = \begin{cases} M(s) - W(s, t) & , s \text{ Vorgänger, aber kein Nachfolger von } t \\ M(s) + W(t, s) & , s \text{ Nachfolger, aber kein Vorgänger von } t \\ M(s) + W(t, s) - W(s, t) & , s \text{ Vorgänger und Nachfolger von } t \\ M(s) & , \text{sonst.} \end{cases} \quad (3.1)$$

Man sagt, t SCHALTET von M nach M' .

Eine *M-aktivierte* Transition ist derart zu interpretieren, dass durch das Eintreten des Ereignisses, verbunden mit den unmittelbaren Folgemarkierungen, die sich ergebenden Markenzahlen nicht den Wertebereich zwischen Null und der Kapazität verlassen.

Dynamik entsteht in diesen Netzen durch eingetretene Ereignisse aus aktivierten Transitionen, die gemäß Punkt 2 der Definition 3.4 schalten. Die nichtaktivierten Transitionen bleiben bei diesem Übergang passiv. In diesem Sinne kann man davon sprechen, dass die Transitionen ASYNCHRON schalten. B/E-Netze sind in dieser Begriffsbildung derart enthalten, dass Kapazitäten und Pfeilgewichte gleich 1 gesetzt werden.

Eine nahe liegende Erweiterung der Dynamik auf S/T-Netzen ergibt sich unmittelbar dadurch, dass eine „Abfolge“ $\omega = t_1 \dots t_n \in T^*$ von Ereignissen t_1, \dots, t_n angegeben wird.⁴ Ferner führt man für die Menge aller Markierungen auf einem Netz N die Abkürzung $\mathcal{M}(N)$ ein.

Definition 3.5 (Folgemarkierung). *Es sei N ein S/T-Netz mit Anfangsmarkierung M_0 und $\omega = t_1 \dots t_n \in T^*$ eine Folge von Ereignissen.*

1. Wir nennen $M'' \in \mathcal{M}(N)$ eine FOLGEMARKIERUNG von M_0 unter ω und schreiben $M_0[\omega > M'']$, wenn

(a) $\omega = \square$ („kein Ereignis ist eingetreten“) oder

(b) $\exists M' \in \mathcal{M}(N) : M_0[t_1 \dots t_{n-1} > M'][t_n > M''$.

2. Dann bezeichnet man ω als eine SCHALTFOLGE, spricht davon, dass ω unter M_0 AKTIVIERT und schreibt dafür $M_0[\omega >$.

3. Sind M_i die jeweiligen Markierungen, d.h. für alle $i = 1, \dots, n$ gilt $M_i = M_0 t_1 \dots t_i$, dann heißt M_0, \dots, M_n die ω zugeordnete Markierungsfolge.

⁴Dabei bezeichnet T^* die Menge aller Wörter aus dem Alphabet der Ereignisse, die durch deren Hintereinanderschreiben (Konkatenation) entstehen. Mit „ \square “ wird das leere Wort dargestellt.

Zwar sind nicht alle Permutationen von ω Schaltfolgen, handelt es sich aber um eine Schaltfolge, so erreicht sie die selbe Markierung. Insofern ist die Definition gerechtfertigt.⁵ Aufgrund dieser Bemerkung ist die Reihenfolge der Ereignisse in ω nicht entscheidend: Diese Reihenfolge gibt keine Auskunft darüber, ob die Ereignisse tatsächlich nacheinander eingetreten sind und in welchem kausalen Verhältnis sie zueinander stehen. Deshalb wird in der Literatur ein allgemeinerer Prozessbegriff eingeführt, der im Wesentlichen eine Abbildung von einem Kausalnetz⁶ auf das interessierende Netz darstellt. Im Zusammenhang mit dieser Arbeit ist die durch Schaltfolgen gegebene Dynamik ausreichend.⁷ Im Anhang B der Arbeit wird beispielhaft an der Dynamik von neuronalen Netzen erläutert, wie S/T- bzw. B/E-Netze im Zuge der Modellbildung eingesetzt werden kann.

Gleichgültig, wie Dynamik erzeugt wird, gemeinsam ist beiden Ansätzen, dass durch die vorgegebene Folge von Ereignissen, sei es in Schaltfolgen oder als Urbilder in der Prozessabbildung, ein diskreter Zeitbegriff induziert wird, der zur Simulation der Netze benutzt wird: Durch das „Abarbeiten“ der Ereignisfolge t_1, \dots, t_n oder das Durchlaufen des Kausalnetzes kann eine Simulationsuhr definiert werden. Dabei wird über den realen Zeitpunkt des Ereignisses höchstens über ein mitgeführtes Attribut „Eintrittszeit“ eine Aussage gemacht (vgl. Abschnitt 2.2). Beim Eintreten von Ereignissen springt die Simulationsuhr um einen Zeitpunkt weiter. Die Ereignisse schalten asynchron. Im folgenden Abschnitt wird der umgekehrte Weg gegangen: Die Dynamik wird über eine Simulationsuhr durch gleichzeitiges Eintreten aller Ereignisse gesteuert.

3.3 Level-Raten-Modelle

Die Ereignisknoten haben in den bisher betrachteten Modellen lediglich auslösenden Charakter. Wie viel von einer Vorgänger-Stelle zu einer Nachfolger-Stelle über eine Transition bewegt wird, ist abhängig von den Pfeilgewichten. Nunmehr betrachten wir Netze mit reellwertigen Markierungen und einer expliziten vorgegebenen Zeitskala $\mathbb{T} \in \{\mathbb{N}_0, \mathbb{R}_0^+\}$. Letzteres führt dazu, dass mit diesem Netztyp dynamische Systeme durch zeitorientierte Simulation nachgebildet werden. In diesem Fall geht man dazu über, in den Ereignisknoten zu entscheiden, wie viel entlang der ein- und ausgehenden Kanten fließt. In diesem Sinne haben diese Knoten eine Ventil-Funktion. Damit werden Netzmodelle mit der Theorie der Differenzial- und Differenzengleichungen verbunden und eröffnen systemtheoretische Zugänge zur Modellbildung.⁸ Falls mit Netzen dieser Semantik gearbeitet wird, bezeichnet man die Ereignisknoten als RATEN und die Zustandsknoten als LEVEL. Die Modelle werden dann als LEVEL-RATEN-MODELLE (kurz: L/R-Modelle) genannt und man führt als Darstellungsmodell das Level-Raten-Diagramm ein.

⁵Vgl. [Bau90] S. 86.

⁶Das ist ein kreisfreies Netz mit unverzweigten Stellen.

⁷Für genauere Darstellung von Prozessen vergleiche man etwa [Bau90] S. 108f und [Rei82] S. 39 f.

⁸Vgl. hierzu etwa [For72].

3.3.1 Begriffsbildung

Zunächst muss durch die explizite Zeitangabe der Markierungsbegriff zeitabhängig formuliert werden:

Definition 3.6 (zeitabhängige Markierung). Sei N ein Netz, das zeitorientiert simuliert werden soll, und \mathbb{T} die zugehörige Zeitskala. Eine Abbildung $M : Z \times \mathbb{T} \rightarrow \mathbb{R}$ heißt ZEITABHÄNGIGE MARKIERUNG.

Definition 3.7 (einfache Ratengleichung; Levelzustand). Sei $N = (L, R, F)$ ein Netz mit Zeitskala $\mathbb{T} \in \{\mathbb{N}_0, \mathbb{R}_0^+\}$. Ferner seien $t \in T$, $r \in R$ und $l \in L$.

1. Die Markierung $M(t, l)$ des Levels l zum Zeitpunkt t wird auch als dessen ZUSTAND zum Zeitpunkt t bezeichnet. Man schreibt auch kurz $l(t)$.
2. Mit $\mathcal{L}(r)$ werden die Level bezeichnet, deren Zustand die Rate r beeinflussen.
3. Eine Funktion $RGL_r(\bullet, t) : \bigotimes_{i \in \mathcal{L}(r)} l_i \rightarrow \mathbb{R}$ heißt EINFACHE RATENGLEICHUNG von r , falls sie in der Form

$$RGL_r(\bullet, t) = \frac{1}{T} f(\bullet), \quad T \in \mathbb{R}, \quad (3.2)$$

geschrieben werden kann.

Im Allgemeinen ist man am Wert der Markierungen der Level interessiert, so dass eine Unterscheidung zwischen dem Namen und der Markierung überflüssig ist. Die Werte der Ratengleichung einer Rate r werden mit $r(t, t + \Delta t)$ bezeichnet. Dabei deutet die Intervallschreibweise darauf hin, dass der Fluss von r im Intervall $[t, t + \Delta t)$ stattfindet. Daher muss eine Ratengleichung in obiger Form geschrieben werden können. Der Fluss wird in „Einheiten pro Zeit“ gemessen. Die Zeitkonstante $T \in \mathbb{R}$ wird im Allgemeinen (d.h. für $T > 1$) als VERZÖGERUNGSZEIT interpretiert. Genauer zur Bedeutung der Größe Δt findet sich in Abschnitt 3.3.3.

Zur endgültigen Definition von Level-Raten-Modellen muss noch deren Dynamik festgelegt werden. Bei S/T-Netzen ist sie durch aktivierte Transitionen bestimmt, insbesondere ist die Dynamik vom Zustand der umgebenden Stellen abhängig. Bei Level-Raten-Modellen soll die Quantifizierung der Dynamik komplett in den Raten stattfinden.

Definition 3.8 (einfaches Level-Raten-Modell). Ein Netz $LR = (L, R, F)$ heißt EINFACHES LEVEL-RATEN-MODELL mit Zeitskala \mathbb{T} , falls

1. jeder Rate $r \in R$ eine Ratengleichung gemäß Punkt 3 der Definition 3.7 zugeordnet ist,
2. jedem Level $l \in L$ ein Anfangszustand oder Anfangsmarkierung $l(0) = M(l, 0)$ zugeordnet wird,

3. die Level für $\Delta t \in \mathbb{R}^+$ ihren Folgezustand im Zeitpunkt $t + \Delta t$ gemäß der LEVELGLEICHUNG⁹

$$l(t + \Delta t) = l(t) + \Delta t \left\{ \sum_{r \in \text{pred}(l)} r(t, t + \Delta t) - \sum_{r \in \text{succ}(l)} r(t, t + \Delta t) \right\} \quad (3.3)$$

bestimmen.

Da in allen Ratengleichungen zu jedem Zeitpunkt bestimmt wird, wie groß der Fluss ist, spricht man in Level-Raten-Modellen davon, dass die Raten SYNCHRON SCHALTEN. Ansonsten enthält die Darstellung fast alle notwendigen Informationen zur Erzeugung eines L/R-Modells. Es fehlt lediglich die Möglichkeit der externen Steuerung: Bisher kann Dynamik in einem Level-Raten-Modell nur dadurch entstehen, dass die Level über ihre Zustände Flüsse verursachen. Es soll allerdings auch möglich sein, dass diese Dynamik nicht durch interne Regelung, sondern durch externes Steuern des Modells zustande kommt. Graphentheoretisch ist dies gleichbedeutend mit den am Anfang vom Abschnitt eingeführten Quellenereignissen. Da Level-Raten-Modelle nicht mehr ereignis-, sondern zeitgesteuert arbeiten, werden diesen Quellen-Ereignissen Quellen-Raten zugeordnet, die nichts anderes als die in Definition 1.3 eingeführten zeitabhängigen Funktionen, die Signale, sind.

Ein weiteres Manko der bisherigen Definition von L/R-Modellen ist deren mangelnde Praktikabilität. In realen Systemen werden Größen beobachtet, die sich als „Verknüpfung“ verschiedener Teilgrößen ergeben. Dies kann bisher nicht abgebildet werden. Daher erweitert man den Begriff des L/R-Modells um eine Menge von Konstanten und Parametern, die sich aus den Konstanten und den aktuellen Zuständen ergeben. Sie können auch in den Ratengleichungen verwendet werden.

Definition 3.9 (unabhängige Parameter). Sei P eine endliche Menge, die man als UNABHÄNGIGE PARAMETER bezeichnet. Ihnen wird eine Abbildung $\text{par} : P \rightarrow \mathbb{R}$ zugeordnet. Für $p \in P$ wird $\text{par}(p)$ der Wert des Parameters genannt.

Der Parameter wird mit seinem Wert identifiziert. Die unabhängigen Parameter, werden in einem konkreten Modell als Konstanten realisiert werden. Mit ihrer Hilfe können Zustandsgrößen eines Modells in eine äquivalente, aber ggf. mehr Information tragende Form gebracht werden:

Definition 3.10 (abhängige Parameter). Sei N ein einfaches L/R-Modell mit Parametern P und einer Menge von Signalen $S = \{s \mid s : \mathbb{T} \rightarrow \mathbb{R}\}$. Ferner seien $\tilde{P} \subset P$, $\tilde{L} \subset L$ und $\tilde{S} \subset S$ Teilmengen der Parameter, Level und Signale. Dann nennt man die Menge der Abbildungen

$$Pa = \left\{ pa(\bullet, t) \mid pa(\bullet, t) : \tilde{P} \times \tilde{L} \times \tilde{S} \rightarrow \mathbb{R} \right\}$$

die ABHÄNGIGEN PARAMETER von N .

⁹ $\text{pred}(l)$ und $\text{succ}(l)$ bezeichnen die Vorgänger und Nachfolger von l .

Damit kann der Begriff der Ratengleichung besser für die Praxis gefasst werden, indem eine Ratengleichung nicht als Abbildung aus den Level, sondern als Abbildung aus den abhängigen Parametern aufgefasst wird.

Definition 3.11 (allgemeine Ratengleichung). Sei $N = (L, R, F)$ ein Netz mit Zeitskala $\mathbb{T} \in \{\mathbb{N}_0, \mathbb{R}_0^+\}$. Ferner seien $t \in T, r \in R$ und $l \in L$.

1. Mit $\mathcal{P}(r)$ bezeichnet man die Menge der abhängigen Parameter und Level, die die Rate r beeinflussen.
2. Eine Funktion $RGL_r(\bullet, t) : \bigotimes_{i \in \mathcal{P}(r)} pa_i \longrightarrow \mathbb{R}$ heißt ALLGEMEINE RATENGLICHUNG, falls sie in der Form (3.2) geschrieben werden kann.

Mit diesen Begriffsbildungen kann ein ALLGEMEINES LEVEL-RATEN-MODELL definiert werden.

Definition 3.12 (allgemeines Level-Raten-Modell). $LR = (L, R, F, \Xi, P, Pa, \Upsilon, \mathbb{T})$ sei ein 8-Tupel. Es wird ALLGEMEINES LEVEL-RATEN-MODELL mit Zeitskala \mathbb{T} genannt, falls

1. (L, R, F) mit der Zeitskala \mathbb{T} ein einfaches L/R-Modell gemäß Definition 3.8 darstellt,
2. Ξ eine Menge von Signalen ist,
3. P eine Menge von Parametern bezeichnet,
4. Pa die zugehörigen abhängigen Parameter bestimmt,
5. $\Upsilon \subseteq L \times R \times Pa$ die Menge der Ausgänge der Modells darstellt,
6. die Raten gemäß Definition 3.11 der allgemeinen Ratengleichung synchron schalten und
7. die Level wie in Definition 3.8 schalten.

Diese Definition erfolgt mit der Festlegung der Menge Ausgänge auch formal parallel zu den Systemdefinitionen aus Kapitel 1. Insbesondere ist die Analogie zur Definition 1.5 von dynamischen Systemen erkennbar: Die Überföhrungsfunktion F ist für das Intervall $[t, t + \Delta t)$ in den Levelgleichungen formuliert und die Ausgabefunktion ist nur eine Festlegung, welche Größen Ausgaben sein sollen.

Die Parallele zu Kapitel 1 wird dadurch vervollständigt, dass auch der Begriff der L/R-Modelle modularisiert wird.

Definition 3.13 (modulares Level-Raten-Modell). Sei K eine Indexmenge, die die Teilmodelle referenziert. Dann wird ein MODULARES LEVEL-RATEN-MODELL LR_Σ definiert als

$$LR_\Sigma = \left(\mathbb{T}, \Xi_\Sigma, \Upsilon_\Sigma, F, K, \{(LR_k, I_k) \mid k \in K\}, \{J_k \mid k \in K \cup \{\Sigma\}\} \right),$$

wobei für $k \in K$ $LR_k = (L_k, R_k, F_k, \Xi_k, P_k, Pa_k, \Upsilon, \mathbb{T})$ ein allgemeines L/R-Modell mit den folgenden Eigenschaften ist:

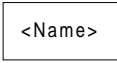
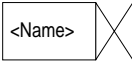


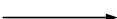
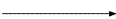


| Symbol | Bezeichnung | Symbol | Bezeichnung |
|---|----------------------|---|--------------------|
|  | Level |  | Rate |
|  | Signal |  | Quelle oder Senke |
|  | Flussrelation |  | "benutzt"-Relation |
|  | abhängiger Parameter |  | Parameter |

Abbildung 3.1: Die in Level-Raten-Diagramme verwendeten Symbole und deren Bedeutung

1. Ξ_Σ ist eine Menge von Signalen.
2. J_k ist eine Schnittstellenabbildung entsprechend Definition 1.7.
3. F ist eine globale Flussrelation zwischen den Komponenten.

In der Praxis ist bei der Definition der Schnittstellenabbildungen einige Vorsicht geboten: Zunächst besagen die Einschränkungen aus der allgemeinen Systemdefinition 1.7 hier nur, dass das Gesamtmodell LR_Σ von Umweltsignalen im bisherigen Sinne gesteuert wird, während die Teilmodelle ihre steuernden Signale aus den sie beeinflussenden Teilmodellen oder den Umweltsignalen erhalten. Die Eingangssignale in die Teilmodelle sind demnach im Allgemeinen Signale, die von anderen Komponenten erzeugt werden.

Die Schnittstellenabbildung induziert hier auch die in der Definition genannte globale Flussrelation. In der Praxis wird dies so geschehen, dass man gemäß den Regeln zum Verfeinern 3.1 und Einbetten 3.2 von Netzen sie schrittweise erweitert und sich so die Teilmodelle immer weiter konkretisieren.

Abschließend soll noch darauf hingewiesen werden, dass diese Modularisierung durch Konkretisierungen nicht nur auf der syntaktischen Ebene der graphischen Darstellungsmodelle, sondern auch auf der semantischen der Markierungen möglich ist. Dazu vergleiche man etwa die Arbeit [HJS83].

3.3.2 Level-Raten-Diagramme

Da diese Definitionen weit über die von Netzen hinausgehen und zusätzlich deren Bedeutung auch geändert worden ist, wird auch das verwendete Darstellungsmodell angepasst, das bisher nur aus bipartiten Graphen mit Kreisen als Stellen und Quadraten als Ereignissen besteht. Für L/R-Modelle werden stattdessen die Symbole aus Abbildung 3.1 verwendet: Raten üben eine Ventilfunktion aus und werden entsprechend symbolisiert. Rechtecke stellen Level dar, die den Fluss durch die Ventile aufnehmen. Über diese Grundkomponenten hinaus werden Symbole für Signale sowie abhängige und unabhängige Parameter eingeführt. Ferner werden die Quellen-Raten aus Quellen gespeist, während Senken-Raten

in Senken enden. Beide sind in der gleichen Symbolik in Level-Raten-Diagrammen integriert. Schließlich gibt eine *BENUTZT-RELATION* an, welche Größen zur Berechnung einer anderen benutzt werden, sofern dies nicht schon durch die Flussrelation gekennzeichnet ist.

3.3.3 Kontinuierliche und diskrete Modellierung

Je nach Zeitskala \mathbb{T} eines L/R-Modells ergibt sich dessen weitere Auslegung: Ist sie diskret, können L/R-Modelle in Differenzgleichungen, bei kontinuierlicher in Differenzialgleichungen umformuliert werden. In beiden Fällen hat die Operationalisierung über die Konstante Δt zu erfolgen. Im ersten Fall wird $\Delta t = 1$ gesetzt werden. Dann kann man auch eine Parallele zu den S/T-Netzen ziehen: Die Ratengleichung bestimmt die Gewichte der ein- und ausgehenden Kanten. Die Gleichungen müssen so formuliert werden, dass die Kapazitäten und der Wertebereich in den natürlichen Zahlen der Stellen beachtet werden. Der Unterschied zwischen L/R- und S/T-Netzen besteht in deren zeit- bzw. ereignisorientierten Simulation. Alle Raten schalten synchron und wirken so gleichzeitig auf die Level, die Ereignisse der Transitionen schalten asynchron. Nicht eingetretene Ereignisse müssen in L/R-Modellen durch Ratenwerte von Null nachgebildet werden. Somit entsprechen sich die Levelgleichungen gemäß Gleichung (3.3) in Definition 3.8 und die Folgemarkierungen aus Gleichung (3.1) in Definition 3.4 bei S/T-Netzen.

Im kontinuierlichen Fall handelt es sich bei L/R-Modellen um eine Diskretisierung von Differenzialgleichungen. Dies sieht man am schnellsten daran, dass die Levelgleichungen gerade die Integrationsvorschrift der *NUMMERISCHEN CAUCHY-EULER-INTEGRATION* sind. Daraus ergibt sich folgender Zusammenhang: Den Level entsprechen die Funktionen, und die Differenz aus Zu- und Abflüssen entspricht deren Ableitung. Die Integrationsschrittweite Δt ist von technischer Natur. Sie gibt an, auf welche Intervalllänge die Ableitungen als konstant angenommen werden. Insgesamt ergibt ein L/R-Modell in dieser Formulierung ein System aus Differenzialgleichungen, deren Anzahl gerade die der Level ist. Eine Inhomogenität der Differenzialgleichung kommt durch zeitabhängige Quellen-Raten, die Signale, zustande. Diese Parallele ermöglicht es, in einer Implementierung von L/R-Modellen auf andere (bessere) Integrationsverfahren als das Cauchy-Euler-Verfahren zurückzugreifen. Im Anhang A sind einige Details zur numerischen Integration zu finden, die die Grundidee der Verfahren und die vorgenommene Implementation darstellen. Ausführlicher wird die Integration von Differenzialgleichungen in [HrW87] oder [SB96] behandelt.

Teil II

Der Prozess der Operationalisierung

Thema des zweiten Teils der Arbeit ist DER PROZESS DER OPERATIONALISIERUNG. Die Darstellung unterteilt sich den Phasen der Modellbildung entsprechend in die drei Kapitel Analyse, Operationalisierung und Validierung.

DAS KOORDINATENSYSTEM DER MODELLBILDUNG wird am Anfang betrachtet. Es wird der wissenschaftliche Prozess analysiert, das Koordinatensystem begründet und schließlich an der Begriffsbildung der Modellbildungs-Lebenszyklen aus dem ersten Teil im Abschnitt 2.3 überprüft.

Das zweite Kapitel MODELLENTWICKLUNG stellt ein Brennglas auf die gesamte Arbeit dar. Es werden mit der Wissensakquisition und der Systemanalyse typische Analysetechniken eingeführt. Ferner wird die im Projekt vorgenommene Operationalisierung beschrieben. Dabei steht die Betrachtung von Modellgeneratoren im Zentrum. Speziell wird eine generische Struktur für modulare Level-Raten-Modelle vorgestellt, die vor dem Hintergrund der zu modellierenden Atmungsphysiologie entwickelt worden ist. Schließlich wird auf die Kalibrierung als Voraussetzung der Validierung eingegangen. Hier ist zunächst eine geeignete Begriffsbildung durch den Rückgriff auf relationale Systeme notwendig. Nach einer kurzen Darstellung der Standardmethoden der Kalibrierung wird ein Verfahren das auf Data Mining von Modelldaten beruht vorgestellt.

Der zweite Teil schließt mit einer Betrachtung zur VALIDIERUNG. Der Analyse bisheriger Ansätze in der Forschung folgt eine Terminologie im Rahmen des Koordinatensystems, die schließlich an Begriffen des Software Engineerings überprüft wird.

Kapitel 4

Das Koordinatensystem der Modellbildung

Schon in der Definition von Objekten (vgl. Kapitel 1.1) ist der Dualismus zwischen dem erkannten Objekt und dem erkennenden Subjekt genannt worden: Es besteht ein Unterschied zwischen dem, was ist, und dem, was wir daraus machen. Da unser Denken in Begriffen stattfindet, ist es sinnvoll zwischen zwei Arten von Objekten zu differenzieren:

1. begriffliche Objekte
2. dingliche Objekte

Wenn nichts anderes gesagt wird, gehen wir hier von dinglichen, greifbaren Objekten aus. Wir werden für die beiden Arten von Objekten und komplexere Strukturen das Koordinatensystem getrennt voneinander begründen. Zuvor sind allerdings einige Bemerkungen zum wissenschaftlichen Prozess notwendig.

4.1 Der wissenschaftliche Prozess

Der Prozess des Erkenntnisgewinns ist zyklisch. Ausgehend von noch nicht erklärten Phänomenen wird eine allgemeine Theorie entwickelt, aus der spezielle Vorhersagen gefolgert werden. Diese werden mit Hilfe von Experimenten, Beobachtungen oder Tests an der Realität überprüft. Abbildung 4.1 deutet dies schematisch an. Prinzipiell können Beobachtungen und Tests eine Theorie endgültig nur **FALSIFIZIEREN**: Widerspricht das beobachtete dem aus der Theorie abgeleiteten Verhalten, so muss (unter Ausschluss anderer Fehlerquellen) die entwickelte Theorie abgelehnt werden.¹

Umgekehrt gilt eine Theorie so lange als nicht widerlegt oder möglich, wie sie nicht falsifiziert werden konnte. Eine Theorie wird man als **BEWÄHRT**² oder (vorläufig) **GÜLTIG** achten, wenn zum einen Falsifikationsversuche an der Theorie und ihren Vorhersagen scheitern und zum anderen daraus abgeleitete abstraktere Theorien mit weiteren Vorhersagen nicht falsifiziert werden konnten. Auf der anderen Seite stellt eine falsifizierte abgeleitete

¹Vgl. hierzu etwa [Pop94] S. 8.

²Vgl. [Pop94] S. 8.

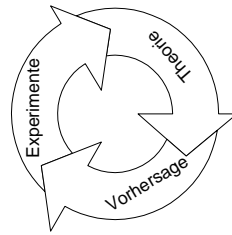


Abbildung 4.1: Der zyklische Prozess der Theoriebildung

Theorie einen Hinweis darauf dar, dass die ursprüngliche Theorie möglicherweise selbst falsch ist. Auf ihre Falschheit kann nicht mit Sicherheit geschlossen werden, da abgeleitete Theorien das Wissen aus anderen Theorien in Betracht ziehen, die ihrerseits bzw. deren Verknüpfung miteinander falsch sein kann. Damit hat der wissenschaftliche Prozess zwei Dimensionen:

1. **Lokal** hat sich eine Theorie in dem Dreiklang aus Theoriebildung, Vorhersage, und Experiment zu bewähren, wie es in Abbildung 4.1 angedeutet wird.
2. **Global** lassen sich aus verschiedenen Theorien weitere ableiten, die wie eine einzelne Theorie behandelt werden müssen.

Der globale Teil ist eine Abstraktion, wie Abbildung 4.2 schematisch darstellt. Die Aussagen aus Theorien der verschiedenen Bereichen werden zu einer abstrakteren umfassenden Theorie verbunden. Die untere BEOBACHTUNGSEBENE ist mit den Theorien T_1, \dots, T_n über die Vorhersagen (in der Abbildung als Pfeile symbolisiert) verbunden. Aus diesen Theorien wird eine abstraktere T_{neu} abgeleitet. Ein Ziel einer abstrakteren Theorie ist es, Vorhersagen zu machen, die aus den bisher bekannten Theorien nicht abgeleitet werden konnten, was in der Abbildung mit der Vorhersage rechts aus der neuen Theorie angedeutet wird. Lassen sich die neuen Vorhersagen nicht im Experiment reproduzieren oder in Tests bestätigen, so ist eine mögliche Fehlerquelle, dass eine der konkreten Theorien falsch war.

Offensichtlich entsprechen sich der prinzipielle Prozess der lokalen Theoriebildung gemäß 4.1 und das grundsätzliche Vorgehen der Modellbildung in Abbildung 2.5. Die Modellvalidierung erfolgt durch Experimente, die Theorie entspricht der Modellbildung und die Vorhersagen der Modellimplementation. Dieser Vergleich führt wieder zurück zu der eingangs vorgenommene Unterscheidung zwischen dinglichen und begrifflichen Objekten. Im Allgemeinen kann man davon ausgehen, dass Theorie der allgemeinere Begriff ist, dennoch wollen wir der Einfachheit halber den Modellbildungszyklus auf dingliche und den Theoriebildungszyklus auf begriffliche Objekte anwenden. Zunächst wenden wir uns den dinglichen Objekten zu.

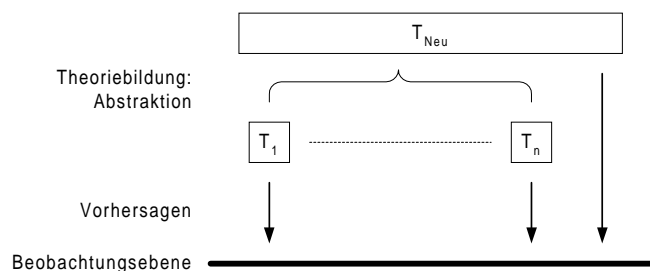


Abbildung 4.2: Abstraktion in der Theoriebildung

4.2 Begründung des Koordinatensystems

Aus dem ersten Teil der Arbeit und dem vorhergehenden Abschnitt ist klar geworden, dass drei Grundstrukturen zu beachten sind. Es sind dies einerseits die dinglichen Objekten, andererseits solche, die zunächst nur als Begriff existieren, und schließlich Objekte, die aus Teilobjekten zusammengesetzt sind und so komplexe Strukturen bilden.

4.2.1 Das Koordinatensystem für dingliche Objekte

Werden Dinge betrachtet, ist der Unterschied zwischen den Attributen und Eigenschaften offenkundig (vgl. Abschnitt 1.1). Eigenschaften von Objekten werden in unserem Verstand zu Attributen. Wenn schon auf dieser „niedrigen“ Ebene zwischen der Realität und unserer Wahrnehmung differenziert werden muss, bedarf die Unterscheidung zwischen System und Modell keiner weiteren Begründung.

Auf „Modellseite“ wird im Modellbildungs-Grundmodell (Abbildung 2.5) aus einem Systemmodell das Simulationsmodell implementiert. Da sich am Systemmodell nicht direkt das Modellverhalten ablesen lässt, sondern dies über das implementierte Simulationsmodell geschieht, spaltet sich die Modellebene in eine abstrakte und eine konkrete Ebene auf.

Auf Seiten des Systems ist der Unterschied zwischen einer abstrakten und einer konkreten Ebene ebenso offensichtlich. Ein System ist mehr als das, was man davon sieht. Daran haben auch die immer besser werdenden Beobachtungsmethoden nichts geändert. Sie haben das Problem nur auf eine andere Ebene gehoben. Vielmehr muss man immer wieder vom Verhalten auf die Struktur eines Systems schließen, falls man eine Whitebox-Modellierung im Auge hat, wie sie in Abschnitt 2.1.2 beschrieben worden ist.

Soll das Verhalten in einer Blackbox modelliert werden, bleibt die Unterscheidung zwischen Verhalten und Struktur weiter gültig, ist aber zunächst nicht von besonderem Interesse. Das erstellte Modell ist prinzipiell nur für Daten aus dem selben Zeitbereich gültig wie diejenigen Daten, mit denen das Modell erstellt worden ist. Außerdem besteht die Gefahr, dass Zusammenhänge zwischen Daten modelliert werden, die nur scheinbar miteinander korreliert sind. Insofern sollte man auch hier die Systemstruktur nicht völlig außer Acht lassen.

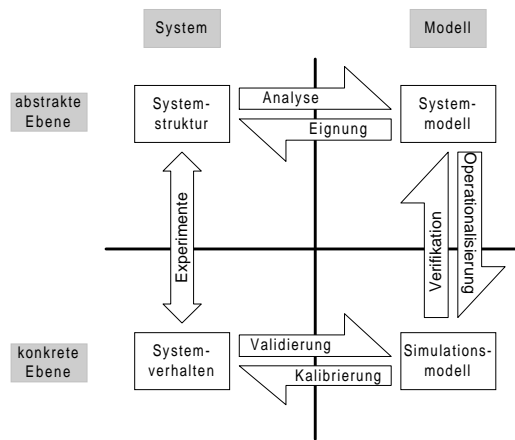


Abbildung 4.3: Das Koordinatensystem der Modellbildung als dessen Grundstruktur

Allerdings ist es gerade die Unterscheidung zwischen der Systemstruktur und dem Systemverhalten, die im Modellbildungs-Grundmodell aus Abbildung 2.5 nicht mit in Betracht gezogen worden ist. Hier wird stattdessen eine Grundstruktur gemäß Abbildung 4.3 vorgeschlagen: Sie entspricht dem Grundmodell mit dem Unterschied, dass auf Seiten des Systems zwischen dessen Verhalten und der Struktur unterschieden wird; durch Experimente erhält man aufgrund des Systemverhaltens Erkenntnisse über dessen Struktur. Das Koordinatensystem bildet zum einen die Realität besser ab, wie auch eine Betrachtung möglicher Fehlerquellen bei der Modellbildung zeigt: Erstens ist es möglich, dass das beobachtete Verhalten mit dem System nicht in Zusammenhang steht. Darüber hinaus besteht die Gefahr, die Zusammenhänge im System durch eine ungeeignete Begriffsbildung falsch zu interpretieren bzw. durch vorschnelle Identifikation von Begriff und Realität zu falschen Schlussfolgerungen zu kommen. Schließlich hat diese Grundstruktur den Vorzug der Symmetrie.

In der Abbildung 4.3 des Koordinatensystems sind die entsprechenden Revisionsprozesse mit aufgenommen worden. Der Analyse steht die grundsätzliche Modelleignung gegenüber, wie sie auch im Grundmodell eingeführt worden ist. Die Modellimplementation oder Operationalisierung wird mit der Modellverifikation überprüft. Im Grundmodell 2.5 fehlt zwischen dem Originalsystem und dem Rechnermodell ein Arbeitsschritt. Es wird nur die Modellvalidierung angegeben. Zur Überprüfung, ob ein Modell gültig ist, muss es allerdings zunächst an eine reale Situation des Systemobjekts angepasst — kalibriert — werden. Deshalb ist der Arbeitsschritt der Kalibrierung eingefügt worden. Der zugehörige Revisionsprozess der Validierung ist jedoch hier mit der Einschränkung zu lesen, dass nicht nur die Kalibrierung, sondern auch das ganze Modell auf dem Prüfstand steht: System und Modell lassen sich nur anhand des Verhaltens der beiden vergleichen: Das Verhalten des Modells wird durch Simulationen erzeugt, das an realen Daten des Systems geprüft werden muss.

4.2.2 Das Koordinatensystem für begriffliche Objekte

Der Aufsatz „Zur Validierung von Konstrukten“ hat die Idee des Koordinatensystems geliefert. Dort wird im sozialwissenschaftlichen Kontext zwischen den Konstrukten und den persönlichen Dispositionen (wie z.B. Parteiidentifikation), die Handlungen (wie z.B. Wahlentscheidungen) veranlassen, unterschieden. Ein Konstrukt stellt eine „Erfindung“ des Forschers dar, um eine Handlung zu erklären, während sie von latent vorhandenen persönlichen Dispositionen gesteuert wird. Aus diesen (begrifflichen) Konstrukten werden beobachtbare Indikatoren abgeleitet, die in empirischen Tests angewandt werden können. In diesem Zusammenhang ist nach zwei Arten von Validität zu fragen:³

1. **Indikatorenvalidität** fragt, „ob und inwieweit die ausgewählten Indikatoren die angezielte Dimension“ erfassen.
2. **Konzeptvalidität** fragt, „ob und inwieweit das theoretische Konstrukt eine valide Hypothese impliziert.“

Die folgende Abbildung 4.4 zeigt in Anlehnung an [Fal77] schematisch die dargestellten Zusammenhänge. Es stehen sich die Objekt- und die sprachlich-instrumentelle Ebene gegenüber. Auf Seiten des Objekts wird zwischen einer abstrakten latenten und einer manifesten Dimension unterschieden; beide sind über Indikatoren, die in Tests überprüft werden, verbunden.

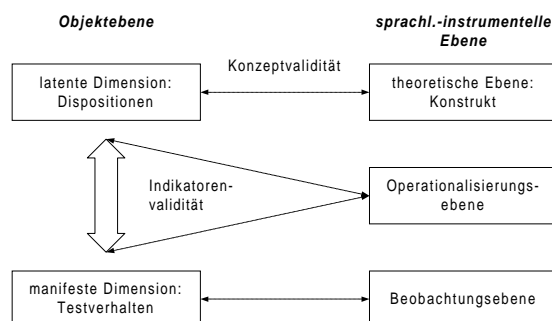


Abbildung 4.4: Das Verhältnis von Objekt und Konstrukt nach Falter

Die sprachlich-instrumentelle Ebene enthält auf der theoretischen Ebene die Konstrukte; dies sind die Begriffe aus einer „theoretischen Sprache“, welche die in der latenten Dimension zusammengefassten Dispositionen in der Objektenebene benennt. Operationalisiert werden die Konstrukte in der Beobachtungsebene. Dort werden Begriffe, sog. Indikatoren, entwickelt, die beobachtbares Verhalten beschreiben. Daraus ergibt sich sofort das Koordinatensystem für begriffliche Objekte in Abbildung 4.5. Dem Simulationsmodell entsprechen die aus dem Konstrukt hergeleiteten Indikatoren, die in Tests angewandt werden können. Der Systemstruktur entspricht die reale Disposition der Menschen. Bei dinglichen Objekten überprüft man in der Revisionsphase „Eignung“, ob man die Strukturen des Systems richtig erkannt hat; für begriffliche Objekte fragt die Eignung, ob es für das begriffliche Konstrukt überhaupt eine reale Entsprechung gibt.

³[Fal77] S. 371.

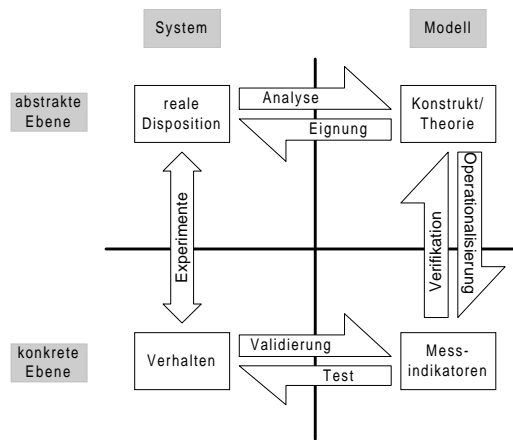


Abbildung 4.5: Das Koordinatensystem für begriffliche Objekte

4.2.3 Das Koordinatensystem bei komplexen Strukturen

Im Allgemeinen beschränkt sich der wissenschaftliche Prozess nicht auf einen Aspekt, so wie er in den beiden Abschnitten zuvor dargestellt worden ist. Vielmehr werden Theorien aus verschiedenen Bereichen auf einer abstrakteren Ebene miteinander verknüpft. Um sinnvoll formulieren zu können, nehmen wir zunächst an, dass diese „konkreten“ Theorien mit dem Verhalten, das sie erklären, identifiziert werden können. Lässt man diese Vereinfachung zu, so stellt sich der wissenschaftliche Prozess wieder wie in Abbildung 4.1 dar. Aus den verschiedenen Theorien (bzw. dem Verhalten, das sie erklären) wird eine abstraktere Theorie gewonnen; daraus wird potenzielles Verhalten abgeleitet, das in Experimenten mit der Realität verglichen wird. Damit lassen sich nunmehr die in den vergangenen Abschnitten angeführten Argumente entsprechend wiederholen. Der Systemstruktur entspricht die reale Verknüpfung der durch die konkreteren Theorien beschriebenen Systeme. Das Systemmodell ist die neue Theorie. Ihre und die bisherigen Vorhersagen entsprechen dem Simulationsmodell, das wiederum an der Realität überprüft werden kann. Insgesamt entspricht dies der Abbildung 4.2. Dabei werden die konkreten Theorien $T_1 \dots T_n$ mit ihren Vorhersagen und der Beobachtungsebene im III. Quadranten zusammengefasst. Die neue Theorie T_{Neu} wird im II. und I. Quadranten in einen Objekt- und Konstruktbereich getrennt und die neu gewonnenen Vorhersagen werden im IV. Quadranten mit der Beobachtungsebene konfrontiert.

Gemäß dem Falsifikationsprinzip aus Kapitel 4.1 ist die Interpretation eines misslungenen Experiments interessant. Hier sind grundsätzlich zwei Fälle zu unterscheiden:

1. Die „konkreten“ Theorien sind richtig: Hier ist im Zuge des Abstraktionsprozesses, der Operationalisierung in neue Vorhersagen oder bei der Durchführung der Experimente ein Fehler unterlaufen.
2. Eine der „konkreten“ ist falsch: Hier hat man möglicherweise alles richtig gemacht, sich aber auf falsche Vorergebnisse verlassen.

Welcher der beiden Fälle eingetreten ist, muss im Einzelfall geprüft werden. Falls das Experiment glückt, hat man eine neue nicht widerlegte Theorie und auch einen weiteren Hinweis darauf, dass sich die konkreten Theorien bewährt haben.

4.3 Das Koordinatensystem im Software Engineering

Ein erster Test, ob die Begriffsbildung des Koordinatensystems gelungen ist, findet durch eine Überprüfung anhand der Software-Lebenszyklen, wie sie in Abschnitt 2.3 eingeführt worden sind, statt.

4.3.1 Vergleich mit den Ebenenmodellen

Die V-Modelle stellen Arbeitsschritte jeweils auf verschiedenen Ebenen gegenüber: So laufen beispielsweise in Abbildung 2.6 die Modellbildung und die Ergebnispräsentation oder in 2.7 die Anforderungsdefinition und die Validierung auf einer Ebene ab. Beim zuerst genannten allgemeinen Ebenenmodell wird darüber hinaus besonderen Wert darauf gelegt, dass auf unterschiedlichen Ebenen verschiedene Personen in den Arbeitsprozess involviert sind. Die Arbeitsschritte enden mit unterschiedlichen Modellen: Der Systemanalytiker liefert Systemmodelle und der Informatiker darauf aufbauend Simulationsmodelle. Der Transfer der Lebenszyklen des Software Engineerings der V-Modelle in das Koordinatensystem hat daher folgendermaßen zu geschehen:

1. **Experimente.** Diese Phase liegt nur indirekt vor. Auf Seiten des Auftraggebers muss zwischen den Fachabteilungen und dem Management Konsens darüber hergestellt werden, dass zum einen Notwendigkeit für den kommenden Modellierungsprozess besteht und zum anderen die Mittel bereit gestellt werden.
2. **Analyse/Eignung.** Während die Analyse dem Arbeitsprozess der Modellierung entspricht, an dessen Ende ein Systemmodell steht, ist deren Revision, die Eignung, hier (zunächst) nur ein konzeptioneller Schritt und findet sich im V-Modell erst später.
3. **Operationalisierung/Verifikation.** Gemäß dem Ebenenmodell ist in dieser Phase der Informatiker beschäftigt. Er implementiert das Modell und führt zusammen mit den Systemanalytikern die Auswertungen durch. Diese finden zunächst unter dem Aspekt der Korrektheit der Implementation statt.
4. **Test/Validierung.** Aufgabe dieser Phase ist es, das Simulationsmodell mit der Realität zu konfrontieren. Dies geschieht in den V-Modellen durch die Ergebnispräsentation. Hier wird schließlich entschieden, ob ein gültiges Modell entwickelt worden ist. Falls in der Analyse-Phase an der Aufgabenstellung vorbei modelliert worden ist, müsste von hier ein wahrscheinlich aufwändiger „Rückschritt“ in diese Phase stattfinden: Das Modell wäre ungeeignet. So wird der zunächst konzeptionelle Schritt der Modelleignung von hier aus mit Leben gefüllt.

Treten Fehler beim zuletzt genannten Revisions Schritt auf, so ist deren Beseitigung meist sehr aufwändig und damit kostenintensiv. Dieser Mangel hat zur Entwicklung weiterer Lebenszyklen geführt, die mit Hilfe von Prototyping schon früher Revisions Schritte vorsehen.

4.3.2 Prototyping

Es wird das Koordinatensystem mit dem für das Prototyping typischen Spiralmodell aus Abbildung 2.8 verglichen. Die Idee des Prototyping besteht darin, die aufwändigen Revisionsprozesse durch frühzeitige Entwicklung von Prototypen, die auf etwaige Risiken aufmerksam machen, zu verhindern. Daraus ergibt sich für die Übertragung des Koordinatensystems, dass an die Stelle eines theoretischen Konstrukts — dem Systemmodell — ein Prototyp tritt, der weiterentwickelt und validiert werden muss. Am Ende eines jeden Zyklus steht ein „Simulationsmodell“, das in der Planungsphase in Angriff genommen worden war. Dabei handelt es sich um eine systematische Operationalisierung eines Prototypen. Kurzum: Das Koordinatensystem lässt sich lokal für jeden Zyklus im Spiralmodell anwenden. Die Dimensionen Kosten und Projektfortschritt werden dann durch die der Abstraktion/Konkretion und System/Modell ersetzt.

Weitere Durchläufe im Spiralmodell erfolgen wie im Abschnitt 4.2.3 über das Koordinatensystem bei komplexeren Strukturen beschrieben: Die bis zum aktuellen Durchlauf entwickelten Modelle werden als vorläufig validiert angenommen. Dann wird nicht mehr zwischen ihnen und den Strukturen und Abläufen, die sie abbilden, unterschieden, und sie gehen als „Teil der Realität“ in die Planung der nächsten Phase ein.

Kapitel 5

Modellentwicklung

Ausgehend von den grundlegenden Analysetechniken der Wissensakquisition und der objektorientierten Systemanalyse wird eine geeignete Operationalisierung für modulare dynamische Systeme, die in modularen L/R-Modellen dargestellt sind, demonstriert. Dabei wird auch das Zusammenspiel von Prototypen und Modellgeneratoren eingegangen. Das Kapitel schließt damit, dass nach einer geeigneten Begriffsbildung eine Methode der Kalibrierung durch Data Mining vorgestellt wird.

5.1 Analyse in der Modellentwicklung

Die Theorie dieser Arbeit wird im Hinblick darauf entwickelt, komplexe dynamische Systeme in ein simulierbares Modell zu überführen und dieses zu implementieren. Dabei müssen in der Analyse die Ebenen des Modellkontextes, des Problembereiches sowie der softwaretechnischen Realisierung bedacht werden.

Die Analyse des Kontextes führt auf die Frage nach dem Modellzweck und den für eine Anforderungsdefinition im Software Engineering notwendigen Arbeitsschritten und Dokumenten, wie sie Abschnitt 2.1.2 angesprochen worden sind und in jedem Lehrbuch zum Software Engineering (wie etwa [Bal96]) zu finden sind. Hier sollen der Problembereich und die softwaretechnische Ebene betrachtet werden: Die üblichen Analysekonzepte sind hier die WISSENSAKQUISITION und SYSTEMANALYSE. Beides sind eigene Forschungsgebiete. Eine umfassende Würdigung ist an dieser Stelle nicht möglich. Vielmehr soll klar werden, dass es diese Techniken sind, die zur Systematisierung des Übergangs vom System zum Modell genutzt werden.

5.1.1 Wissensakquisition

Die Modellierung komplexer dynamischer Systeme ist mit der Entwicklung eines Expertensystems vergleichbar, da der Experte im Allgemeinen nicht in der Lage ist, sein Wissen in ein simulierbares Modell zu überführen. Probleme bereiten die Kenntnisse der Programmierung, die Anwendung der Simulationssoftware oder die Formalisierung des Wissens. Dies ist im Zusammenhang von Expertensystemen Aufgabe des Wissensingenieur, der in diesem Kontext als MODELLINGENIEUR bezeichnet werden soll. Wieder lässt sich die Si-

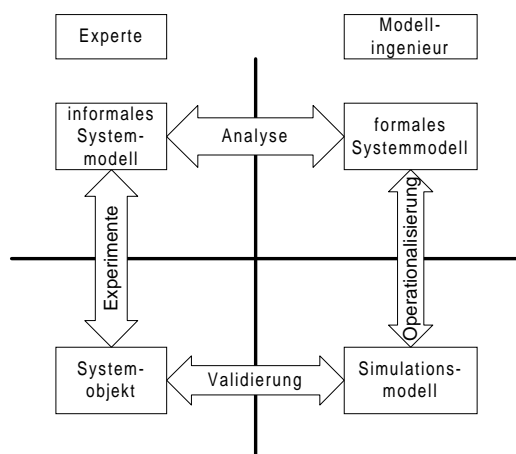


Abbildung 5.1: Expertensysteme und dynamische Systeme

tuation mit Hilfe des Koordinatensystems beschreiben, wie es Abbildung 5.1 zeigt. Auf der Systemseite steht der Experte. Aus der jeweiligen Fachwissenschaft oder dem Fachgebiet hat er ein Systemmodell des Systemobjekts. Aufgabe der Analyse, die hier zur Wissensakquisition wird, ist es, dieses Wissen an den Modellingenieur in einer Form zu übermitteln, aus der dieser (in Zusammenarbeit mit dem Experten) ein Simulationsmodell erstellen kann.

An dieser Stelle soll exemplarisch auf zwei verbreitete Ansätze und die jeweiligen Techniken aufmerksam gemacht werden.

prototypische Akquise. Nach [BBB⁺83] erfolgt der Wissenserwerb zyklisch in 5 Phasen gemäß Abbildung 5.2:

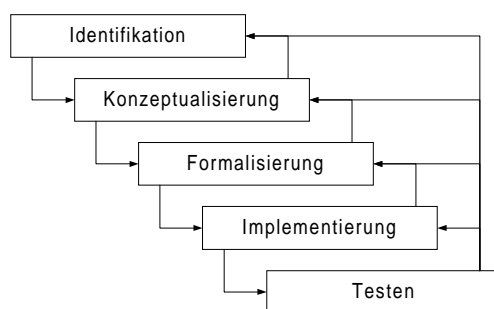


Abbildung 5.2: Der Zyklus der prototypischen Wissensakquisition

Zunächst wird das Systemobjekt eingegrenzt, der Problembereich wird identifiziert und analysiert. Ist dies geschehen, müssen Konzepte entwickelt werden, wie das Wissen über das Systemobjekt strukturiert werden kann. Diese Strukturen müssen dann auch tatsächlich in ein formales Systemmodell gebracht werden. Schließlich wird dies implementiert und getestet. Die vorgesehenen Revisionszyklen gehen hier entweder jeweils eine Stufe zurück oder man kehrt vom Testen in eine der vorhergehenden Phasen zurück.

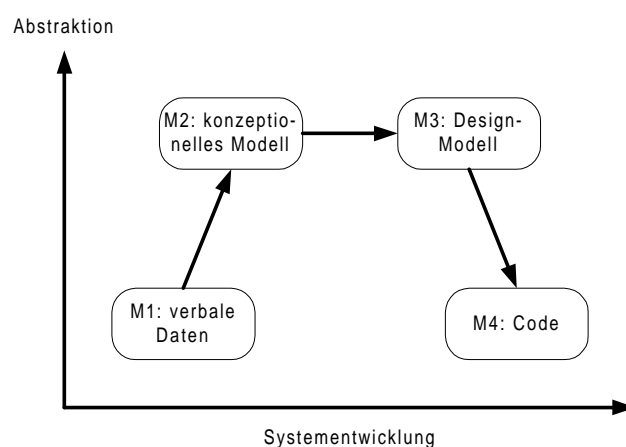


Abbildung 5.3: Stadien der Modellierung in KADS

modellorientierte Akquise. Auf der Basis des (zunächst möglichst vollständig) erhobenen Wissens wird vor dessen Design und Implementierung ein abstraktes, implementationsunabhängiges Modell der Expertise durch Analyse und Interpretation des Wissens gewonnen.¹ Dabei ist es eine Möglichkeit, Modellierungsansätze zu benutzen, „die eine generelle, anforderungsspezifische Neuentwicklung oder Adaption von nicht-operationalen Modellen erlauben, wie KADS (Knowledge Acquisition and Documentation System).“ In diesem System werden gemäß Abbildung 5.3 nacheinander vier Stadien der Modellierung durchlaufen: „Ausgehend von verbalen Daten (M1) wird ein konzeptionelles Modell (M2) aufgebaut, das über ein Design-Modell (M3) in eine ablauffähige Wissensbasis (M4) verwandelt wird.“² Innerhalb dieser Prozessmodelle lassen sich methodisch drei Techniken unterscheiden: (vgl. [Pup88]):

1. indirekter Wissenserwerb

In Befragungen erhält der Wissensingenieur vom Experten sein Wissen.

2. direkter Wissenserwerb

Der Experte formalisiert sein Wissen selbst.

3. automatischer Wissenserwerb

Aus vorhandenen Falldaten wird automatisch Wissen extrahiert.

Gleichgültig, welche Techniken angewandt werden, können allgemeine Anforderungen an Methoden der Wissensakquisition formuliert werden:³

1. Die Methoden sollten einfach zu handhaben sein.
2. Die Vorgehensweise sollte in Etappen nachvollziehbar sein.
3. Das Ziel sollte sein, das Expertenwissen vollständig hervorzuholen.

¹Vgl. [Uth94] S. 54.

²[KL90]S. 75 und S. 66.

³Vgl. [KL90] S. 15 f, der [Fre85] S. 152 und [GC86] S. 2 zitiert.

4. Die Vorgehensweise sollte messbare Meilensteine beinhalten.
5. Repräsentation von wichtigen Konzepten des Experten soll in aufgabenspezifischen Formalismen erfolgen.
6. Die Repräsentationsformalismen sollen deklarativer Art sein.
7. Die Repräsentation soll auf der Abstraktionsebene des Experten stattfinden.

Gerade die drei zuletzt genannten Prinzipien aus [GC86] werden sich für die Modellierung komplexer dynamischer Systeme als wesentlich erweisen.

5.1.2 Systemanalyse

Während im letzten Abschnitt der Fokus auf Methoden aus dem Bereich der Expertensysteme gelegen hat, stehen hier die im Software Engineering entwickelte Technik der objektorientierten Systemanalyse im Mittelpunkt. Wieder werden die Zusammenhänge nur kurz beschrieben. Als Darstellungsmodell wird UML benutzt.

Die objektorientierte Systemanalyse erlaubt es dem Systemanalytiker, „Anforderungen an ein neues Softwaresystem problemnah zu modellieren.“⁴ Ausgehend von der bekannten Unterscheidung zwischen Objekt und Klasse differenziert man zwischen der statischen und dynamischen Modellierung durch Attribute und Operationen. In dieser Arbeit werden nur die statischen Konzepte benötigt, auf die sich hier beschränkt wird:

Assoziation „modelliert Beziehungen zwischen Objekten gleichrangiger Klassen.“ Diesen Relationen gibt man im Allgemeinen Namen. Assoziationen haben Kardinalitäten. Sie geben an, „mit wie vielen anderen Objekten ein Objekt einer bestimmten Klasse in einer konkreten Beziehung stehen kann bzw. stehen muss.“ Dabei werden die Symbole aus der folgenden Tabelle 5.1 in den Diagrammen verwendet:

| Symbol | Bedeutung |
|--------|--------------------------|
| 1 | = 1 |
| * | ≥ 0 |
| 0..1 | 0 oder 1 |
| m..n | $[m, n] \cap \mathbb{N}$ |

Tabelle 5.1: Kardinalitäten bei Assoziationen in UML-Diagrammen

Schließlich können an eine Assoziation Restriktionen geknüpft sein, so dass jene nur erlaubt wird, wenn die Nebenbedingungen erfüllt sind. In UML-Notation wird eine Assoziation als Verbindungslinie zwischen zwei Objekten dargestellt, an deren Rand jeweils die Kardinalitäten stehen und die Restriktion in Klammern an die Linie geschrieben wird, wie es in Abbildung 5.4 angedeutet ist: Dort ist keines oder ein Objekt der „Klasse 1“ mit 0 bis n Objekten der „Klasse 2“ assoziiert.

⁴Vgl. [Bal96]. S. 154 f. An dieser Stelle finden sich auch die übrigen Zitate zur Systemanalyse.

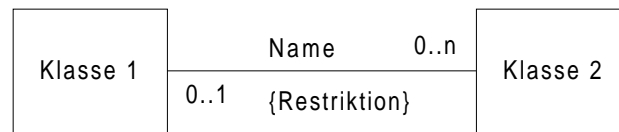


Abbildung 5.4: Die Assoziation zweier Klassen

Aggregation ist „eine gerichtete Assoziation zwischen Objekten. Sie stellt einen Sonderfall der Assoziation dar und liegt vor, wenn zwischen den Objekten der beteiligten Klassen (kurz: den beteiligten Klassen) eine Rangordnung gilt, die sich durch ‚ist ein Teil von‘ bzw. ‚besteht aus‘ beschreiben lässt.“ In UML-Notation wird die Aggregation durch eine Raute an der Verbindungslinie der Aggregationsklasse zur Teilklasse symbolisiert (vgl. Abbildung 5.5).

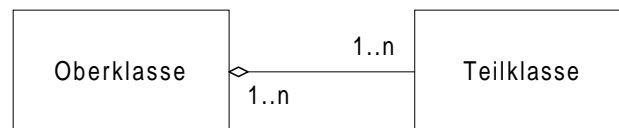


Abbildung 5.5: Die Aggregation zweier Klassen

Vererbung beschreibt die Beziehung zwischen einer allgemeinen Klasse (Basisklasse) und einer spezialisierten Klasse. Zweck der Vererbung ist es, zum einen die Relationen zwischen Objekten zu verallgemeinern bzw. zu spezialisieren und zum anderen Eigenschaften und Funktionen zusammenzufassen, so dass dann auch (und gerade) mit Hilfe von abstrakten, nicht einsetzbaren Klassen polymorphe Methoden zur Verfügung stehen. Ein Dreieck in der Verbindung zwischen der Basisklasse und der spezialisierten Klasse symbolisiert in UML-Notation die Vererbung (vgl. Abbildung 5.6).

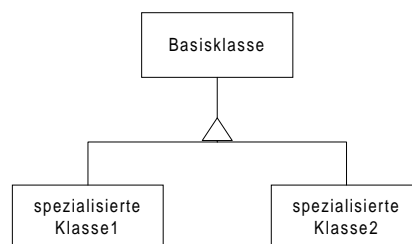


Abbildung 5.6: Die Vererbung von Klassen

5.1.3 Einordnung der Verfahren

Eine notwendige Vorarbeit, um mit der Wissensakquisition und der Systemanalyse den Kern des Systems herausarbeiten zu können, besteht zunächst darin dessen Rahmen abzustecken. Hierzu ist auf die Definition eines Modellzwecks mit zu erstellender Anforderungsdefinition zu verwiesen worden. Im Kontext des Koordinatensystems ist dies so zu verstehen, dass zunächst festgelegt werden muss, was überhaupt mit in den III. Quadranten aufgenommen wird. In der Sprache der Systemdefinition heißt dies, die Grundmenge

der potenziell zu betrachtenden Objekte U festzulegen (vgl. Kapitel 1.1). Alles Übrige bleibt außen vor.

Die Zyklen der Wissensakquisition beginnen ebenfalls mit einer entsprechenden Phase. In der prototypischen Akquise wird das Problem identifiziert. Durch die Konzeptionalisierung wird das identifizierte System auf einer abstrakteren Ebene betrachtet, von der aus es in einem Analyseprozess formalisiert wird. Implementierung und Testen finden sich direkt im Koordinatensystem. Somit passt auch dieser Lebenszyklus in das vorgestellte Schema. Im modellbasierten Ansatz nach Abbildung 5.3 ist die Parallele noch offensichtlicher. Ausgehend von den verbalen Daten im III. Quadranten kann der modellbasierte Ansatz direkt ins Koordinatensystem eingepasst werden.

Unabhängig vom zugrunde liegenden Zyklus der Akquise wird man für die Modellierung komplexer dynamischer Systeme den Wissenserwerb indirekt durchführen. Weder lässt er sich automatisieren, noch ist dem Experten selbst die Implementierung zuzumuten. Dies führt zurück auf die Frage des Wissenstransfers und den zugehörigen Repräsentationsformalismen. Letztere sind bei der Formulierung der Anforderungen an Akquisemethoden betont worden und spielen auch in der Modellbildung eine wichtige Rolle: Hier wird deutlich, weshalb in Kapitel 2.3.1 der Standardentwurf in der Modellbildung derart formuliert worden ist, dass das Darstellungsmodell im Mittelpunkt steht. Es dient als „Brücke“ zwischen dem nicht formalisierten Systemmodell des Experten und dessen Darstellung in einer simulierbaren Sprache. Der Wissenstransfer wird dadurch erleichtert, dass sich der Modellingenieur intensiv mit dem System befasst und dadurch ein Grundverständnis für das System entwickelt.

Im Zusammenhang mit komplexen dynamischen Systemen führt diese Herangehensweise direkt zu den Level-Raten-Modellen, wie sie in 3.3 behandelt worden sind. Sie besitzen mit Level-Raten-Diagrammen Darstellungsmodelle, die leicht verständlich und somit geeignet sind, den Experten sein Wissen darstellen zu lassen. Dann hat er „nur“ noch die nötigen Ratengleichungen einzugeben. Daraus kann ein Simulationsmodell erzeugt werden.

Liegt der Fokus in der Wissensakquise eindeutig auf der Systemseite, dient die Systemanalyse dazu, „die Anforderungen an ein neues Softwaresystem problemnah zu modellieren.“ Im konkreten Fall hängt die Entscheidung über die Auswahl der Verfahren von den Kenntnissen und Vorlieben der beteiligten Personen sowie der Problemstellung ab.

5.2 Operationalisierung in der Modellentwicklung

Dieser Schritt im Prozess der Modellbildung beschäftigt sich damit, aus einem formalen oder unformalen Systemmodell ein Simulationsmodell zu erstellen. Hier hat sich ein Zusammenspiel aus Prototypen und problemspezifischen Modellgeneratoren als günstig erwiesen. Im ersten Abschnitt wird darüber ein kurzer Überblick gegeben. Davon ausgehend wird im zweiten Abschnitt eine generische Struktur entwickelt, wie sie im Zusammenhang mit der Atmungsphysiologie und modularen L/R-Modellen benötigt wird. Schließlich wird auf einen speziellen Aspekt, die Operationalisierung der Zeit, eingegangen.

5.2.1 Prototypen und Modellgeneratoren

Der Zweck von Modellgeneratoren liegt auf 3 Ebenen.⁵ Zunächst soll er die Entwurfsideen für das zu entwickelnde Modell durch einen geeigneten Aufbau und Interaktionsmöglichkeiten unterstützen. Dieser Anspruch ist leichter formuliert als realisiert, da man oft in der Entwicklung eines Modells nicht weiß, welche Modelltypen und Konzepte überhaupt zum Einsatz kommen werden. Zum Realisieren der Entwurfsideen gehört zweitens, dass der Generator ein simulierbares Modell zu liefern hat, mit dem dann — drittens — durch geeignete Interaktion und „Spielen“ an den Parametern erste Überprüfungen der Modellbildung und weitere Untersuchungen durchgeführt werden können.

Modelle werden nur selten als reiner Selbstzweck erzeugt; sie entstehen vielmehr in einem größeren Zusammenhang, in dem sie auch produktiv eingesetzt werden sollen. Vor diesem Hintergrund sind kommerzielle Simulationswerkzeuge zu bewerten: Hier zeigt ein Vergleich, „dass allenfalls kommerzielle Systeme im deutlich gehobenen Preisbereich diese Anforderungen weitestgehend erfüllen. Deren Nachteil liegt, außer im hohen Preis, häufig aber auch in mangelnder Flexibilität und unspezifischem Leistungsspektrum, was andererseits einen erhöhten Einarbeitungsaufwand bedingt.“⁶ Insbesondere ist zu klären, ob und wie das Simulationsmodell mit den übrigen Systemkomponenten interagieren kann.

Somit ist es eine gangbare Alternative, für einen eingegrenzten Anwendungsbereich einen Modellgenerator zu entwerfen. So hat der Autor im Rahmen seiner Diplomarbeit [No198] Erfahrung bei der Entwicklung eines Generators für Level-Raten-Modelle gesammelt, der speziell für die Anwendung in der Sportinformatik entworfen worden war. Auch aus dieser Erfahrung ergeben sich die Vorteile einer Eigenentwicklung:

1. Der Aufwand zur Einarbeitung minimiert sich, solange der Entwickler auch zu den Anwendern gehört.
2. Der Leistungsumfang des Programms entspricht genau den spezifischen Anforderungen.
3. Falls im Zuge der Modellentwicklung weitere Features benötigt werden, können diese entworfen und integriert werden.
4. Schließlich ist die Lösung preiswert.

Dem stehen natürlich auch Nachteile gegenüber:

1. Die Eigenentwicklung kostet Zeit und ist als Neuentwicklung fehleranfällig.
2. Die wahrscheinlich notwendigen Entwicklungen neuer Features erschwert einen „geschlossenen“ Entwurf. Die Gefahr, Lösungen „mit heißer Nadel zu stricken“, ist groß.

Aus diesen Erfahrungen heraus ist bei der Modellierung der Atmungsphysiologie im EWS-Projekt (vgl. Teil 3) einen anderen Weg gegangen. Er ist zunächst aus praktischen Gründen beschränkt worden, vereinigt aber in sich die jeweiligen Vorteile der Nutzung vorhandener Programme und der Eigenentwicklung:

⁵Vgl. [PLG02].

⁶[PLG02] MI 4.2.

1. Erstellung eines Prototypen des zu erstellenden Modells mit Hilfe eines Freeware-Modellgenerators.
2. Programmierung eines funktionalen Prototypes zur Weiterentwicklung des Modells, der beteiligten Algorithmen und des gesamten Softwaresystems auf dieser Grundlage.
3. Entwicklung einer generischen Struktur, die Weiterentwicklungen ermöglicht.

Diese drei Phasen sollen genauer betrachtet werden:

Erster Prototyp

In den Abschnitten über den Standardentwurf 2.3.1 und über die Wissensakquise 5.1.1 ist dargestellt worden, dass das Darstellungsmodell das zentrale Mittel ist, den Wissenstransfer vom unstrukturierten Systemmodell des Experten zum formalisierten Systemmodell des Modellingenieurs zu leisten. Infolgedessen ist gerade in der Anfangsphase eines Projektes eine graphische Unterstützung des Prozesses wichtig: Der Modellingenieur kennt die Hintergründe des Systems noch nicht in ausreichendem Maße; der Experte ist nicht mit den Formalismen der „Simulationsprache“ vertraut, so dass eine geeignete graphische Repräsentation zum „Dolmetschen“ dient. Andererseits wäre eine geeignete Eigenentwicklung nicht möglich, da alle notwendigen Konzepte noch nicht bekannt sind und außerdem ein längerer Vorlauf notwendig wäre. Daher ist die Nutzung eines Freeware-Modellgenerators zur Erstellung eines ersten Prototypen ein geeigneter Einstieg in ein Projekt. Mit diesem Prototypen ist dann eine Grundlage gelegt, auf der die an der Modellierung Beteiligten sich auf den jeweils anderen Partner und dessen Sichtweise einstellen können.

Funktionaler Prototyp

Diese Phase erscheint nur in größeren Projekten notwendig, in denen das zu erstellende Modell ein Subsystem darstellt und Zeitdruck herrscht. In eng bemessenen Zeitplänen, muss de facto zu jedem Zeitpunkt ein zumindest in großen Teilen funktionsfähiger Prototyp zur Verfügung stehen, damit die Arbeit stets auf den folgenden drei Ebenen weitergeführt kann:

1. Die Interaktionen der Systemkomponenten und auch die übrigen Systemkomponenten müssen (weiter-)entwickelt werden.
2. Der Aufbau des Modells muss geprüft und entwickelt werden.
3. Die Algorithmen des Modells, insb. zur Kalibrierung, müssen entwickelt und getestet werden.

Modellgenerator

Die Anforderungsdefinition hat in der Analysephase die Ziele der Modellbildung definiert. Die Prototypen haben eine „Kommunikationsbasis“ geschaffen, einen Eindruck von der kommenden Struktur des Modells geliefert und somit einen wichtigen Baustein zur Wissensakquisition geliefert: Vom Prototypen ausgehend kann der Modellingenieur sich in

den Problembereich einarbeiten⁷ und mit dem Experten eine Systemanalyse beginnen, die schließlich in eine generische Struktur zur Weiterentwicklung münden kann.

5.2.2 Eine komponentenorientierte generische Struktur für Level-Raten-Modelle

Ausgangspunkte sind die Definitionen 1.7 und 3.13 von modularen dynamischen Systemen und modularen Level-Raten-Modellen. Sie werden in eine generische Struktur übertragen, so dass sich modulare dynamische Systeme über modulare L/R-Modelle simulieren lassen. Anschließend wird auf die Struktur einer Modellkomponente, die als Klasse `TPMComponent` implementiert ist, eingegangen. Der Abschnitt endet mit der Beschreibung des Gesamtmodells als Klasse `TPMModel`.

Der grundsätzliche Aufbau

Ein modulares dynamisches System setzt sich aus miteinander in Wechselwirkung stehenden kleineren dynamischen Systemen, den Teilsystemen, zusammen, die dann zu Modellkomponenten werden. Sie sind im Gesamtsystem aggregiert, wie die folgende Abbildung 5.7 zeigt:

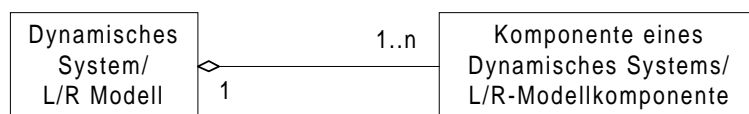


Abbildung 5.7: Verhältnis von Gesamtsystem zu Teilsystemen

Im Extremfall, in dem das Gesamtsystem nur aus einer Teilkomponente besteht, muss die Komponente die gesamte Funktionalität eines Modells erfüllen können. Das Gesamtmodell ist strukturell die Summe der Teilmodelle. Es dient dazu, die Prozesse in den Teilmodellen zu koordinieren. Daher fasst man das Gesamtmodell selbst als eine Modellkomponente auf, die über Vererbung und Polymorphie nach dem Prinzip der Fortpflanzung von Operationen⁸ die nötigen Aktionen in den Komponenten steuert.

Hierzu zeigt Abbildung 5.8 die softwaretechnische Modellierung. Von `TPMComponent` werden die Klassen `TPMModel` und `TPMProblemComponent` abgeleitet. Alle Operationen in der Basisklasse werden als virtuell deklariert, so dass sie in der Klasse `TPMModel` überschrieben werden können. Dort werden sie dahingehend umgewandelt, dass in einer Schleife die jeweiligen Operationen der im Modell aggregierten Komponenten aufgerufen werden.

Die Schnittstellenabbildung aus den Definitionen für modulare Systeme und L/R-Modelle könnte man in der Klasse für das Gesamtmodell realisieren, da jene zwischen den einzelnen Komponenten stattfindet. Allerdings ist es einfacher, nicht nur spezielle Werte einer

⁷Hier gelten natürlich die grundsätzlichen Bedenken gegenüber Prototypen weiter. Außerdem ist hier aber auch deren Nutzen insofern zu erwähnen, als sie das Verständnis des Experten für die Simulation steigern und umgekehrt der Modellingenieur durch die Simulationsergebnisse sein Verständnis für Effekte im System vertiefen kann.

⁸Vgl. [Rum93] S. 74.

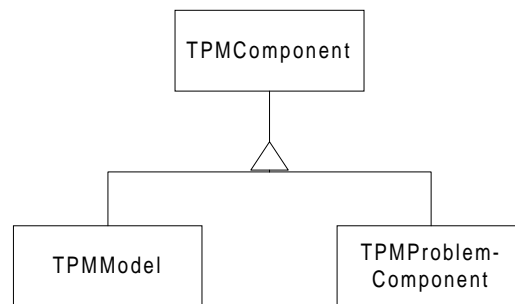


Abbildung 5.8: Verhältnis von Gesamtmodell, Komponenten und Problemkomponenten

anderen Komponente zur Verfügung zu stellen, sondern die sich gegenseitig benutzenden Komponenten über Zeiger miteinander zu assoziieren, so dass die Schnittstellenabbildung zu einer „benutzt“-Relation zwischen den Komponenten wird.

Die zeitorientierte Simulation ist der grundlegende Algorithmus der Modelle dynamischer Systeme. Über weiter notwendige Algorithmen kann unabhängig von der Problemstruktur keine Aussage gemacht werden. Daher ist es sinnvoll, das Modell und dessen Steuerung zu trennen, so dass das Modell als einzig „problemorientierten“ Algorithmus die Simulation enthält.⁹ In Anlehnung an die Terminologie von Expertensystemen soll diese Komponente `EXPERTENKLASSE` genannt werden. Sie ist in jedem Fall für jedes neu zu entwickelnde Modell ebenfalls neu zu konzipieren, da damit auch neue Probleme gelöst werden müssen. Kurzum: Instanzen des steuernden Experten, `TPMExpert`, und des Modells von der Klasse `TPMMoel` sind assoziiert, wie Abbildung 5.9 zeigt. Da aber das Modell auf einer generischen Struktur basiert, ist a priori nicht klar, ob der implementierte Experte das Modell richtig steuern kann. Hierfür hat der Entwickler Sorge zu tragen.

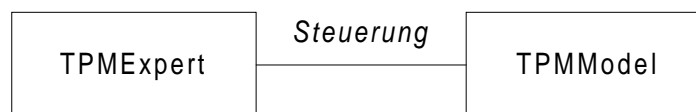


Abbildung 5.9: Modellsteuerung durch Expertenklasse

Von der eben genannten Steuerung ist die Steuerung eines Modells durch die Eingangssignale zu unterscheiden. Aus der Sichtweise des Experten sind die Eingangssignale Teil des Modells, die bei dessen Benutzung gehandhabt werden müssen. So sind auch bei den formalen Definitionen von modularen dynamischen Systemen und L/R-Modelle die Eingangssignale Teile der selben. Daher ist es sinnvoll, unabhängig von der weiteren Problemstruktur, eine Klasse einzuführen, die die Umwelteinflüsse auf das Modell darstellt, `TEnvironment`, während die übrigen Probleme im „Inneren“, `TInternal` gelöst werden, wie die folgende Abbildung 5.10 andeutet:

⁹Diese Darstellung ist offensichtlich objektorientiert motiviert. Mit nicht problemorientierten Algorithmen wird etwa das Laden und Speichern verstanden.

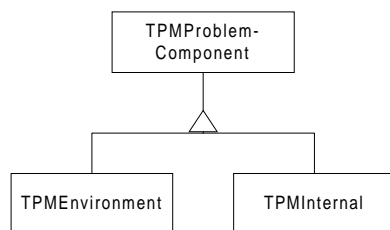


Abbildung 5.10: Modellierung der Systemumwelt

Die Klasse TPMComponent

Die Darstellung der Funktionalität einer Modellkomponente in einem softwaretechnischen Entwurf wird in einen Abschnitt über die statischen und einen über die dynamischen Aspekte unterteilt.

Statisches Modell: Attribute. Gemäß Definition 3.13 besteht eine Komponente eines L/R-Modells aus einer Menge von Raten, Level mit ihren Startwerten, unabhängigen und abhängigen Parametern. Ihr Verhältnis zur Komponente ist die Aggregation (vgl. Abschnitt 5.1.2), wie Abbildung 5.11 zeigt:

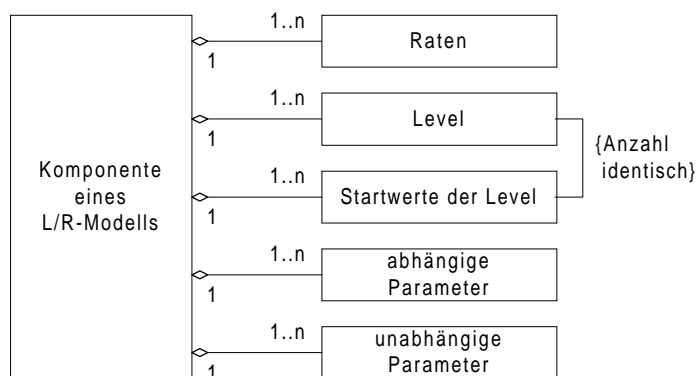


Abbildung 5.11: Modellierung einer Komponente eines L/R-Modells

Die Parametermengen sind unabhängig von den Level und den Raten. Zwar sind Parameter wie Verzögerungen der Raten oder Kapazitäten der Level einer Größe zuzuordnen, doch ist dies nicht immer eindeutig, so dass die hier vorgenommene Modellierung flexibler erscheint. Über den Namen der Level bzw. der Rate soll der Wert des Zustandes bzw. des aktuellen Flusses abgefragt werden können. Daher sind die Startwerte und die Level lediglich assoziiert und die Startwerte nicht in die Level integriert. Damit ist allerdings auf die Restriktion zu achten, dass die Anzahl und die Namen der Startwerte mit denen der Level übereinstimmt.

Dies ist der Kern der statischen Struktur einer Komponente eines L/R-Modells. In der Realisierung sind noch etliche Attribute zu implementieren, die allerdings nicht den inhaltlichen Kern betreffen und zum Teil auch in direktem Zusammenhang mit dem zugrunde liegenden Problem der Atmungsphysiologie, das im dritten Teil behandelt wird, stehen.

Beispielhaft wird auf zu modellierende Zeitskalen und der damit verbundenen Verwaltung der simulierten Werte hingewiesen: Für eine Auswertung ist es unerlässlich, den Verlauf einer Simulation zu speichern, da erstens für eine Auswertung möglicherweise Daten auf verschiedenen Zeitskalen interessant sind¹⁰ und zweitens graphische Darstellungen den Verlauf von Simulationsergebnissen reproduzieren sollen. Daher erhalten alle Modellkomponenten noch Listen, in denen die Simulationswerte gespeichert werden. Dies führt auf die Betrachtung der Dynamik eines L/R-Modells.

Dynamisches Modell: Operationen. Der einzige problemorientierte Algorithmus ist die Simulation. Dies setzt natürlich voraus, dass die Modellkomponente erzeugt und richtig initialisiert worden ist. Außerdem muss festgelegt werden, welche anderen Komponenten im Sinne der Schnittstellenabbildung sie beeinflussen. Ferner werden entsprechend dem allgemeinen Schema der zeitorientierten Simulation aus Abschnitt 2.2.1 Methoden zu deren Initialisierung sowie Methoden zum Berechnen der einzelnen Größen im Modell benötigt. Insgesamt sind zur Realisierung einer konkreten Modellkomponente nur die virtuellen Methoden aus Tabelle 5.2 der Basisklasse `TPMComponent` zu überschreiben.

| Zusammenhang | Methode |
|----------------------------|--|
| Erzeugen | <code>build</code> <code>InitModel_ConnectComponents</code> |
| Simulationsinitialisierung | <code>SetDefaultStartvalues</code> <code>InitSimulation_SetParam</code> |
| Simulationsdurchführung | <code>CalculateRates</code> <code>FlowAddition</code> |

Tabelle 5.2: virtuelle Methoden der Klasse `TPMComponent` zur Konkretisierung von Problemkomponenten

Somit erweist sich diese Struktur als generisch. Sind neue Komponenten zu entwickeln, leitet man sie von der Grundkomponente ab und muss nur diese problemorientierten Funktionen überschreiben. Jeglicher Verwaltungsaufwand ist schon in der Grundkomponente und deren Zusammenspiel mit der Komponente `TPMModel`, die das gesamte Modell darstellt, realisiert. Im Anhang E wird am Beispiel der Komponente `TPMVolCompartment` das Vorgehen genau illustriert.

Die Klasse `TPMModel`

Zwar ist das Gesamtmodell strukturell die Summe seiner Teilmodelle, doch bis zum simulationsfähigen Programm ist noch einiges an Verwaltungsaufwand zu leisten. Die wichtigsten Punkte sind:

Komponenten verwalten. Aufrufe an das Gesamtmodell werden im Sinne der Fortpflanzung der Operationen in einer Schleife an die einzelnen Komponenten weitergegeben. Die Komponenten sind im Modell in einem entsprechenden Feld enthalten und können

¹⁰Vgl. 1.2 für die Begriffsbildung im System, 2.2 für die Simulation, 5.2.3 für die allgemeinen Aspekte der Modellierung und 8.2 für den speziellen Fall der Atmungsphysiologie.

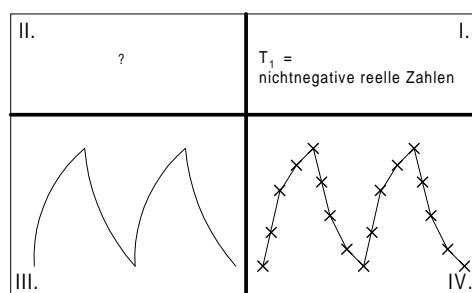


Abbildung 5.12: Das Koordinatensystem für die Modellierung der Zeit

über ihren Namen, einen Index oder einen Parameter, den sie enthalten, angesprochen werden.

Modell erzeugen. Die `build`-Routinen der Komponenten bearbeiten den jeweiligen Problembereich. Zuvor werden die Komponenten in der `build`-Routine des Modells erzeugt. Dort werden auch die Parameter, die sich auf das Gesamtmodell beziehen, gesetzt. Dies sind neben der von der jeweiligen Anwendung abhängigen Parameter die Integrations-schrittweite Δt für die Simulation durch numerische Integration (vgl. Abschnitt 3.3.3), die Simulationsdauer und die Speicherfrequenz, auf die im folgenden Abschnitt eingegangen wird.

Simulation. Der Kern der Simulation ist die numerische Integration. Hierzu ist ein Runge-Kutta-Verfahren implementiert worden, das als Objekt im Modell enthalten ist und dessen Grundlagen im Anhang A dargestellt ist. Dieses benötigt eine Funktion, die aus den Zuständen der Level über deren Ableitung — der Differenz aus Zu und Abflüssen (vgl. Kapitel 3.3.3) — die neuen Zustände bestimmt. Hierzu ist im Modell die Methode

```
NewRates(states : array of double;
          var newstates : array of double)
```

implementiert, die vom konkreten Modell unabhängig ist und nur die Funktionen `CalculateRates` und `FlowAddition` der Komponenten aufruft. Schließlich wird im Modell der gesamte Ablauf der Simulation in der Routine `Simulate` durchgeführt.

5.2.3 Die Modellierung der Zeit

Der Begriff der Zeit ist uns in dieser Arbeit schon mehrfach begegnet. Bei der Definition eines dynamischen Systems gemäß 1.5 ist eine Zeitskala explizit anzugeben. Man unterscheidet dort zwischen diskreten und kontinuierlichen Systemen. Wie gewohnt steht der kontinuierliche Fall im Mittelpunkt: Zur Modellierung dynamischer Systeme stehen wiederum kontinuierliche und diskrete Methoden zur Verfügung. Es sind dies kontinuierliche Differenzialgleichungen bzw. diskrete L/R-Modelle. Diese Modelle können wiederum kontinuierlich (z.B. durch symbolisches Rechnen in Mathematica) oder diskret durch numerische Integration simuliert werden.

Für den Fall, dass für das dynamische System eine eindeutige Zeitskala definiert werden kann, soll die Situation wieder am Koordinatensystem der Modellbildung veranschaulicht werden: Der Ausgangspunkt der Betrachtungen liegt in Abbildung 5.12 im III. Quadranten. Der Prozess des dynamischen Systems findet zu jedem Zeitpunkt statt. Erst durch Beobachtungen aus der Systemumwelt entsteht im dritten Quadranten eine BEOBACHTUNGSSKALA. Wie die Prozesse im Inneren des Systems (im II. Quadrant angedeutet) ablaufen, ist nicht klar. Diese sind sogar im Allgemeinen von jedweder „Uhr“ unabhängig.

Im einfachsten Fall sind die Beobachtungsskala und die MODELLIERUNGSSKALA im zu erstellenden Systemmodell identisch. Allerdings können sich beide auch unterscheiden. In diesem Zusammenhang wird zwischen der „Realzeit, der beobachtete und gemessene Zeitablauf im realen System“ und der „Modellzeit, der Nachbildung der Realzeit im abstrakten Modell“¹¹ unterschieden.

Des Weiteren gilt es, die Zeitskala im Systemmodell von der in der Simulation abzugrenzen. Denn wie oben beschrieben worden ist, kann ein kontinuierliches Systemmodell diskret simuliert werden. Die Unterscheidung zwischen der Systemmodellzeit und der Simulationsmodellzeit führt zunächst zurück auf die Ebenen eines Modells, wie sie in Abschnitt 2.3.1 aus dem Standardentwurf entwickelt worden sind. Der Problembereich „enthält“ die Systemmodellzeit, und der Datenhintergrund korrespondiert mit der Simulationsmodellzeit. Doch gerade im Fall einer diskreten Simulation führt der Standardentwurf zu einer weiteren Unterscheidung: Zur Berechnung wird ein diskretes Zeitraster benutzt. Im Allgemeinen wird man nicht jeden simulierten Wert speichern, sondern nur auf einen bestimmten Anteil, der als SPEICHERFREQUENZ bezeichnet werden soll. Bei der Visualisierung der Simulationsergebnisse werden ggf. nicht alle gespeicherten Werte dargestellt, sondern z.B. jeder zweite. Folglich müssen bei der Operationalisierung vom System- zum Simulationsmodell gemäß dem Standardentwurf die drei Zeitebenen SYSTEMMODELL-, SIMULATIONSMODELL- UND DARSTELLUNGSMODELLZEIT unterschieden werden.

Bei der Modellierung der Zeit ist auch der Fall zu beachten, dass im selben Modell ggf. verschiedene Zeitskalen von Interesse sind. Im Systemmodell will man Aussagen für jeden Zeitpunkt und z.B. Minuten treffen. Dann gilt zunächst für jede einzelne Skala das zuvor Gesagte. Zusätzlich muss nun beachtet werden, dass die Simulation nur auf der feinsten Skala stattfindet und die Simulationsdaten auf eine höhere Skala aggregiert werden. Dies erfolgt entweder mit jedem simulierten Wert, sozusagen online, oder nach der Simulation — offline — auf den gespeicherten Werten, was zu zusätzlichen Verzerrungen führen kann.

Schließlich sei noch an eine weitere Zeitebene erinnert. Die „Computerzeit ist die für eine Simulation des Modells auf einem Rechner benötigte Zeit.“¹² Diese stellt zwar mit dem Einzug der schneller werdenden Rechner ein geringer werdendes Problem dar, muss aber gerade im Zusammenhang mit in Echtzeit arbeitenden, komplexen Modellen beachtet werden.

¹¹Vgl. [Uth94] S. 22.

¹²Vgl. [Uth94] S. 22.

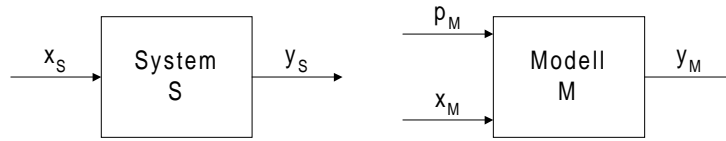


Abbildung 5.13: Vergleich von System und Modell zur Begriffsbildung der Kalibrierung

5.3 Kalibrierung

Ist ein System analysiert und zu einem Simulationsmodell operationalisiert worden, muss sowohl das Modell validiert als auch die Operationalisierung verifiziert werden. Hierzu wird man die Ergebnisse des Modells in „geeigneter Weise“ mit den Daten des Systems vergleichen. Dies setzt voraus, dass das Modell in ebenso „geeigneter Weise“ an die Daten des Systemfalls angepasst oder KALIBRIERT ist. Damit beschäftigt sich dieser Abschnitt: Zunächst wird eine geeignete Begriffsbildung eingeführt, die sich an Prognosemodellen (vgl. Abschnitt 2.2.2) orientiert. Dann wird auf Standardmethoden eingegangen und schließlich wird gezeigt, wie Konzepte des Data Minings zum Kalibrieren verwendet werden können.

5.3.1 Begriffsbildung zur Kalibrierung

Ausgangspunkt ist die Unterscheidung zwischen dem Funktionsmodell für das System und dem für das Modell aus Abschnitt 2.1.2. Abbildung 5.13 deutet dies nochmals an: Zunächst wird das Modell (rechts) von Parametern p_M und Signalen x_M gesteuert und liefert Simulationsergebnisse y_M . Eigentlich handelt es sich bei den Parametern um Bestandteile des Modells. Für das Folgende ist es aber günstig, sie als konstante Signale, die das Modell steuern, zu interpretieren. Das Systemobjekt wird von den Signalen x_S gesteuert und liefert die Messwerte y_S . Um die Kalibrierungsaufgabe vollständig formulieren zu können, werden die Parameter des Modells p_M nach ihrer Gültigkeitsdauer differenziert. Gültigkeitsdauer ist hier wieder ein Begriff mit zwei Dimensionen. Erstens ist zu prüfen, ob der entsprechende Parameter im System sich in dem zugehörigen Zeitraum ändern kann, und zweitens, ob dieser in dem betrachteten Zeitintervall gemessen werden kann. Dementsprechend können sie im Modell eingesetzt werden.

Vereinbarung 5.1 (Modellparameter). *Wir nennen Modellparameter — Eingangsgrößen in ein Modell —*

1. SIMULATIONSPARAMETER, wenn sie sich von Simulation zu Simulation ändern können,
2. KALIBRIERUNGSPARAMETER, wenn sie sich von Kalibrierung zu Kalibrierung ändern können,
3. SITZUNGSPARAMETER, wenn sie sich nur von Arbeitssitzung zu Arbeitssitzung ändern können und
4. MODELLKONSTANTEN, wenn sie nicht geändert werden.

Die Signale des Modells x_M sind Operationalisierungen der Signale x_S , die das in Abbildung 5.13 links angedeutete Systemobjekt steuern. Die Systemmesswerte y_S können nunmehr in der Modellbildung verschiedene Rollen spielen. Handelt es sich um Werte, die ständig messbar sind, können sie bedenkenlos als Eingangsgrößen des Modells verwendet werden. Sind sie nur zu gewissen Zeitpunkten verfügbar und ist das Modell darauf ausgerichtet, diese Werte zwischen den Zeitpunkten zu prognostizieren, bieten sie sich als Kalibrierungszielwerte an:

Vereinbarung 5.2 (Kalibrierungszielwerte). *Unter den KALIBRIERUNGSZIELWERTEN sollen die Werte verstanden werden, die dazu dienen, die Ausgabewerte einer Simulation die des Systemobjekts anzupassen. Zur Unterscheidung wird von SIMULIERTEN und GEMESSENEN KALIBRIERUNGSZIELWERTEN gesprochen.*

Mit diesen Begriffsbildungen kann die Kalibrierungsaufgabe für Prognosemodelle (vgl. 2.2.2) formuliert werden.

Vereinbarung 5.3 (Kalibrierungsaufgabe). *Unter der KALIBRIERUNGSAUFGABE versteht man, Kalibrierungsparameter auszuwählen und so einzustellen, dass die Abweichung von gemessenen und simulierten Kalibrierungszielwerten eine vorgegebene Toleranzschwelle nach einem festgelegten Maß nicht überschreitet und das Modell im Folgenden gültige Prognosen liefert.*

Der erste Teil der Kalibrierungsaufgabe besteht darin, Kalibrierungsparameter auszuwählen. Dies ist meist nicht eindeutig. Im Allgemeinen wird man mit verschiedenen Parametern zunächst zu gleichen Annäherungen kommen. Erst durch die Güte der Simulationsprognosen wird man die Güte der Auswahl der Kalibrierungsparameter abschätzen können. Im Zweifelsfall und für die Vorauswahl der potenziellen Kalibrierungsparameter ist der Experte zu Rate zu ziehen.

5.3.2 Standardmethoden der Kalibrierung

Sind einmal die Kalibrierungsparameter festgelegt, so lässt sich die Kalibrierungsaufgabe in eine Optimierungsaufgabe umformulieren. Hierzu betrachten wir einen Fall des Systems $S(x_S, y_S)$ aus Ein- und Ausgangssignalen. Ferner seien $y_S^{(K)}$ die gemessenen Kalibrierungszielwerte des Systems, und x_M bezeichne die für das Modell M zu x_S passende Parametrisierung der Eingangssignale. Schließlich seien $p_K \subset p_M$ die Kalibrierungsparameter des Modells. In der Fragestellung der Kalibrierung wird das Modell nicht als Funktion des Eingangssignals, sondern bei gegebenem Eingangssignal als Funktion der Kalibrierungsparameter betrachtet. Dann lautet die Kalibrierungsaufgabe, die Größe

$$\|M_{x_M}(p_K) - y_S^{(K)}\| \quad (5.1)$$

zu minimieren. Dabei bezeichnet $\|\cdot\|$ eine geeignete Norm.

Zur Behandlung des Kalibrierungs- als Optimierungsproblems kann auf die gängige Literatur verwiesen werden. Hier soll lediglich auf das Gemeinsame der Standardverfahren kurz eingegangen werden. Unabhängig davon, welches Verfahren eingesetzt wird, beruhen sie

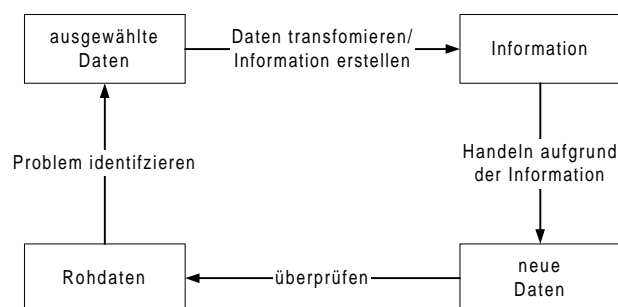


Abbildung 5.14: Ein Data Mining-Zyklus

alle im Wesentlichen darauf, dass von einem Startpunkt ausgehend immer besser werdende Lösungen gesucht werden. Um die Güte einer Lösung zu bestimmen, muss die zu minimierende Funktion ausgewertet werden. In unserem Zusammenhang geschieht dies dadurch, dass eine Simulation durchgeführt wird. Daraus ergibt sich, dass ein solches Verfahren Zeit braucht, weil stets Simulationen nötig sind und daher die Laufzeit verbessert werden kann, indem ein besserer Startpunkt gewählt wird. Wie man diesen mit Hilfe von Data Mining-Methoden finden und so im günstigsten Fall gar keine Iterationen nötig werden, zeigt der nächste Abschnitt.

5.3.3 Kalibrierung durch Data Mining

Im Zuge der Leistungssteigerung von Rechnern haben sich auch die technischen Möglichkeiten der Auswertung von Daten verbessert. Damit hat sich die Forschungsrichtung Data Mining etabliert, die kurz als „die Auswertung und Analyse mit Hilfe von automatischen und halbautomatischen Verfahren von großen Datenmengen, um aussagefähige Muster und Regeln zu finden“¹³, definiert werden kann.

Data Mining-Zyklus und Methoden

In [BL97] findet man einen Data Mining-Lebenszyklus. Abbildung 5.14 ist daran angelehnt: Danach besteht Data Mining aus den vier Phasen:

1. Problem identifizieren
2. Information erstellen
3. Handeln
4. Überprüfen

Zunächst wird das Problem identifiziert und die passenden Daten aus den Rohdaten extrahiert. Mit den ausgewählten Daten folgt der „eigentliche“ Data Mining-Schritt. Die Daten werden in geeigneter Form transformiert, so dass sie die gewünschten Informationen enthalten. Aufgrund der entstandenen Information wird gehandelt. Dadurch entstehen neue Daten, die zum Vergleich mit den Ausgangsdaten dienen.

¹³[BL97], S. 5; Übersetzung des Autors.

Abgesehen davon, dass Data Mining immer den gesamten Prozess umfasst, ist der methodische Kern doch der Übergang von den ausgewählten Daten zur Information. Data Mining hat seine Wurzeln in der Statistik und der künstlichen Intelligenz. Im Zusammenhang von deterministischen dynamischen Systemen sind statistische Verfahren ungeeignet. Klassisch für KI-Methoden sind Entscheidungsbaumverfahren, die im Kern auf „Wenn-Dann-Regeln“ zurückzuführen sind, genetische Algorithmen, die in den hier betrachteten Problemen ihre Güte per Simulation berechnen, sowie Backpropagation-Netze.

Letztere arbeiten in ihrer Nutzungsphase derart, dass sie aufgrund von Operationen, die so aufwändig sind wie Matrizenmultiplikationen, die gegebenen Eingangsdaten in Ausgangsdaten transferieren. Zuvor werden allerdings in einer Lernphase Daten benötigt, die aus Paaren von Ein- und Ausgangsdaten bestehen. Sie werden dazu benutzt, die Parametrisierung des Netzes zu optimieren. Eine etwas ausführlichere Darstellung findet man im Anhang B. Im Gegensatz zu den genetischen Algorithmen benötigen Netze in der Nutzungsphase keine Simulationen.

Data Mining mit Modelldaten

Im Zusammenhang mit der Kalibrierung von Modellen stellt sich die Situation im Data Mining-Zyklus folgendermaßen dar. Ausgangspunkt sind Systemfälle (x_S, y_S) . Im zugehörigen Fall des Modells ist es in irgendeiner Form gelungen, die Kalibrierung durchzuführen, so dass Tupel aus (x_M, p_M, y_M) entstehen. Prinzipiell könnte man diese Modelltupel auch ohne zugehörige Systemdaten erzeugen. Dann läuft man allerdings Gefahr, dass die Modellparametrisierungen völlig an der Realität vorbeigehen. Daher ist zu fordern, dass es sich bei den Modelltupeln mindestens um Szenarien, also plausible Parametrisierungen (vgl. Abschnitt 2.2.2), mit den zugehörigen Ausgaben handelt.

Problemidentifikation: Modelldaten sammeln und aufbereiten. Die erste Phase des Data Mining besteht in unserem Zusammenhang darin, erstens potenzielle Kalibrierungsparameter zu selektieren und zweitens die Datenbasis zu verbreitern. Ersteres geschieht in Kooperation mit dem Experten. Das naive Herangehen bei der Verbreiterung der Datenbasis wäre, die potenziellen Kalibrierungsparameter des Modells systematisch zu variieren und die durch die Simulationen entstehenden Tripel (x_M, p_M, y_M) zu speichern. Dieses Verfahren ist ungeeignet, da mutmaßlich etliche der Parameterkonstellationen keine Szenarien sind, die demnach sinnlose Tupel darstellen. In diesem Zusammenhang ist auch eine SENSITIVITÄTSANALYSE zu sehen: Dabei untersucht man die Empfindlichkeit der Modellausgabe auf die Veränderung eines Parameters. Diese Analyse ist hier ein sich ergebendes Nebenprodukt oder auch Voraussetzung der systematischen Variation.

Darüber hinaus scheidet dieses Vorgehen an der Größe des Suchraums: Gäbe es in einem festen Szenario nur 5 zu variierende Größen mit jeweils 10 möglichen Ausprägungen wären schon 10^5 Simulationen nötig. Abhilfe kann hier die Einsicht schaffen, dass zur Kalibrierung das Zusammenspiel von System- und Modelldaten entscheidend ist. Daher kann man ausgehend von gegebenen Tupeln von Systemdaten (x_S, y_S) (oder bereits simulierten Szenarien) durch systematisches Variieren der potenziellen Kalibrierungsparameter



Abbildung 5.15: Aufbau eines Kalibrierungsmodells

im zugehörigen Modellfall (x_M, p_M, y_M) innerhalb einer gewissen Spannbreite neue Fälle generieren.

Der abschließende Schritt der Modellidentifikation, die Datenaufbereitung, lässt sich nur schwer systematisieren: Stets kann man offensichtlich sinnlose Fälle aus der Datenbasis löschen, Statistiken über die Daten zu generieren und die Daten mit Hilfe von Visualisierungsprogrammen „Daumen-Mal-Pi“ zu untersuchen.

Information erstellen: Das Kalibrierungsmodell erzeugen. Das Kalibrierungsmodell hat in seiner funktionalen Sichtweise in der Anwendungsphase die Gestalt aus Abbildung 5.15. Aufgrund einer Teilmenge von Systemmesswerten $y_S^{(K)}$ hat das Kalibrierungsmodell eine Näherung der Kalibrierungsparameter p_K des Modells zu liefern.¹⁴

Aus der Phase der Problemidentifikation stehen nun alle Modellparameter p_M , insb. die Kalibrierungsparameter p_K und die im Modell erzeugten Ausgaben y_M zur Verfügung. Darin enthalten sind auf jeden Fall die zu den Systemmesswerte y_S gehörenden Modellwerte, von denen eine Teilmenge als Eingang in das Kalibrierungsmodell dienen soll. Demnach bietet sich ein Algorithmus an, der mit überwachtem Lernen arbeitet. Der am weitesten verbreitete Typ ist der Backpropagation-Algorithmus für neuronale Netze (vgl. Anhang B). Für die Kalibrierung von Modellen besteht nun die Aufgabe darin, aus den Systemmesswerten diejenigen auszuwählen, die aufgrund der gesammelten Modelldaten die besten Kalibrierungsparameter liefern.

Handeln: Werte der Kalibrierungsparameter für Fälle berechnen. Das erstellte Kalibrierungsmodell wird gemäß Abbildung 5.15 auf real gemessene Systemdaten angewandt. Im Zusammenhang von Backpropagation-Netzen spricht man von der Nutzungsphase.

Überprüfen: Werte auf Plausibilität und Prognosewert prüfen. Die gefundenen Kalibrierungsparameter werden zunächst darauf überprüft, ob sie in der Simulation die nach den gemessenen Daten gewünschten Kalibrierungszielwerte liefern; ggf. sind sie „nur“ als Startwerte für einen iterativen Algorithmus zu nutzen. Schließlich sind die gefundenen Werte auf ihre fachwissenschaftliche Relevanz zu prüfen. An dieser Stelle enden die Möglichkeiten der Informatik.

¹⁴Übrigens handelt es sich hier um eine Form des automatischen Wissenserwerb, wie er in Abschnitt 5.1.1 genannt worden ist.

Kapitel 6

Validierung

In diesem Kapitel werden zunächst drei Validierungskonzepte aus der Literatur analysiert. Daraus wird im zweiten Abschnitt eine Terminologie im Rahmen des Koordinatensystems hergeleitet, die schließlich kurz an Ideen des Software Engineerings überprüft wird.

Zunächst sollen die bisherigen Erkenntnisse zusammengefasst werden. Im ersten Teil der Arbeit ist im Abschnitt 2.1.2 dargestellt worden, dass ein Modell eines Systems stets einem bestimmten Zweck dient und es nur ein vereinfachendes Abbild der Realität ist. Daher kann dieses Abbild nicht richtig oder falsch im Sinne einer endgültigen Wahrheit sein, sondern höchstens für den formulierten Zweck geeignet. Dies ist im zweiten Teil im Kapitel 4.1 dahingehend präzisiert worden, dass eine Theorie und mithin ein Modell endgültig nur falsifiziert werden können. Scheitern FALSIFIKATIONSVERSUCHE, so wird man eine Theorie oder ein Modell als BEWÄHRT oder GÜLTIG oder VALIDE erachten. Somit sind die folgenden Begriffe von Gültigkeit oder Validität als intersubjektiv nachprüfbare Plausibilitäten in Bezug auf einen bestimmten Zweck und nicht als Wahrheiten zu verstehen.

6.1 Analyse der Ansätze in der Literatur

In diesem Abschnitt wird auf das Lehrbuch von Bossel [Bos94] zurückgegriffen, die Arbeit von Uthmann [Uth94] herangezogen und kurz auf den sozialwissenschaftlichen Kontext bei Falter [Fal77] eingegangen.

6.1.1 Validität bei Bossel

Zur Beurteilung, ob ein Modell im Sinne des Modellzwecks hinreichend gut ist, muss die Gültigkeit im Hinblick auf vier verschiedene Aspekte geprüft werden: Verhaltensgültigkeit, Strukturgültigkeit, empirische Gültigkeit, Anwendungsgültigkeit:¹

„**Verhaltensgültigkeit:** Hier muß gezeigt werden, daß für die im Rahmen des Modellzwecks liegenden Anfangsbedingungen und Umwelteinwirkungen des Originalsystems das Modellsystem das (qualitativ) gleiche dynamische Verhalten erzeugt.“

¹Vgl. [Bos94] S. 36. Dort sind auch die folgenden Zitate zu finden.

Damit verlangt Verhaltensgültigkeit QUALITATIV richtige Ergebnisse. Verhaltensphänomene, die sich am realen System zeigen, sind am Modell nachzubilden. Oft ist eine qualitativ richtige Parametrisierung eines Modells Voraussetzung hierfür.

„**Struktur­gültigkeit:** Hier muß gezeigt werden, daß die Wirkungsstruktur des Modells der (für den Modellzweck) essentiellen Wirkungsstruktur des Originals entspricht.“

Dies setzt natürlich voraus, dass es sich um ein Whitebox-Modell handelt, das auch die Struktur des Systems erklären und nicht nur dessen Verhalten beschreiben will. Hat man die Struktur­gültigkeit nachgewiesen, so folgt die Verhaltens­gültigkeit automatisch. Leider kann man die Struktur nur anhand des Verhaltens überprüfen, so dass auch hier wieder nur der negative Schluss zwingend ist: Ist das Modell nicht verhaltens­gültig, so kann es auch nicht struktur­gültig sein.² Hat sich das Modellverhalten bewährt, so wird man aufgrund von fehlender Falsifikation und Plausibilitäten Struktur­gültigkeit annehmen dürfen.

„**Empirische Gültigkeit:** Hier muß gezeigt werden, daß im Bereich des Modellzwecks die numerischen oder logischen Ergebnisse des Modellsystems den empirischen Ergebnissen des Originals bei gleichen Bedingungen entsprechen, bzw. daß sie (bei fehlenden Beobachtungen) konsistent und plausibel sind.“

Das Verhältnis von empirischer Gültigkeit und Verhaltens­gültigkeit führt einerseits auf die Problematik der Kalibrierung und andererseits auf die Unterscheidung zwischen einer Blackbox- und Whitebox-Modellierung. Bossel schreibt an anderer Stelle: Bei einer Mischform der beiden Modellierungsarten „wird versucht, die Wirkungsstruktur des Systems nach besten Kenntnissen so darzustellen, dass sich wenigstens Verhaltens­gültigkeit (qualitativ korrektes Verhalten) ergibt. Die unbekannt­en Modellparameter werden dann so angepasst, dass das Modellverhalten auch numerisch dem bereits beobachteten Verhalten des Originals möglichst genau entspricht (empirische Gültigkeit). Hier wird das Originalsystem also als 'grey box' oder als 'opaque' (halbdurchsichtig) verstanden.“³

In der bisherigen Darstellung ist speziell bei der Definition der Modellparameter davon ausgegangen worden, dass ein Teil der Modellparameter unbekannt ist und damit sinnvoll geschätzt werden muss. In diesem Zusammenhang sind auch Kalibrierungsparameter definiert worden, die sich hier bei Bossel nur in anderen Worten wiederfinden. Allerdings geht er auf die Schwierigkeiten der Kalibrierung nicht ein.

„**Anwendungsgültigkeit:** Hier muß gezeigt werden, daß Modell und Simulationsmöglichkeiten dem Modellzweck und den Anforderungen des Anwenders entsprechen.“

Für die Anerkennung von Anwendungsgültigkeit ist wieder der zugrunde liegende Systembegriff entscheidend. Den Maßstab für die Anwendungsgültigkeit liefert die Anforderungsdefinition. Zur Validierung muss demnach das eigentliche System zusammen mit dessen Umwelt als ein Obersystem betrachtet werden, dessen Modellierung sich dann gemäß der Anforderungsdefinition verhalten muss.

²Hier wird eine fehlerfreie Operationalisierung vorausgesetzt.

³[Bos94] S. 29.

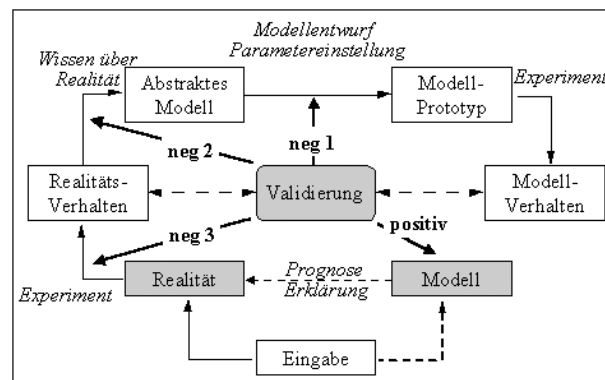


Abbildung 6.1: Interventionsmöglichkeiten bei fehlender Validität

5

6.1.2 Validität bei Uthmann

Aus den prototypischen und modellbasierten Zyklen der Wissensakquise (vgl. Kapitel 5.1.1) entwickelt Uthmann ein auch für die Modellbildung anwendbares KNOWLEDGE ENGINEERING-KONZEPT MIT MODELLBASIERTEM PROTOTYPING.⁴ Im Zentrum dieses Ansatzes steht der Prozess der Validierung, wie es Abbildung 6.1 andeutet. Es werden drei Arten von fehlender Validität unterschieden:

1. Der Übergang vom abstrakten Modell zum Modellprototypen durch den Modellentwurf und/oder die Parametereinstellung war fehlerhaft. Hier wird ein Feintuning der Parameter bzw. eine Ergänzung oder Verfeinerung des Prototypen empfohlen, um Validität herzustellen. Eine zusätzliche Wissensakquise ist nicht nötig.
2. Die „Verhaltens-Lücken“ des Modellprototypen rühren daher, dass das vorhandene Wissen über das System nicht „komplett in einem Schritt abgebildet wurde bzw. nicht ausreichend Wissen vorhanden war, so dass eine weitere Wissenserhebung notwendig wird.“
3. Schließlich stellen sich die Lücken im Wissen über das System als äußerst umfangreich heraus, oder das Wissen selbst und die der Akquise zugrunde liegende Anforderungsdefinition hat nicht zum intendierten Problembereich gehört. In diesem Fall ist stets eine weitere Wissensakquise notwendig, denn schon in der Anfangsphase bei der Feststellung der grundlegenden Problembeschreibung gab es Verständigungsschwierigkeiten. In diesem Fall ist die Systementwicklung „nahezu gescheitert.“⁶

Punkt 1 beinhaltet zwei Aspekte: Das abstrakte Modell wird im Allgemeinen zunächst formalisiert, bevor es zu einem prototypischen Simulationsmodell implementiert wird. Insofern gehören der Modellentwurf und die Parametereinstellung zu zwei verschiedenen Phasen der Modellbildung. Punkt 2 deutet auf ein Kommunikationsproblem oder einen

⁴Vgl. zunächst [Uth94] S. 55 ff und in der Folge für die Modellbildung [PLM97] S. 45 ff.

⁵Die Abbildung ist [PLM97] S. 45 entnommen.

⁶Vgl. [PLM97] S. 46 und [Uth94] S. 62.

typischen Fehler eines Prototypen hin. Entweder ist das Wissen zwischen Modellingenieur und Experten nicht adäquat kommuniziert worden, oder ein Teil des Wissens wurde bei der Bildung des Prototypen übergegangen. Den schlimmsten Fall stellt Punkt 3 dar. Hier hat man quasi an der Sache vorbei modelliert. Schon die Anforderungsdefinition war nicht dem Problem angemessen formuliert. In Systembegriffen heißt dies insbesondere, dass die in der Modellbildung zu betrachtenden Objekte (vgl. Abschnitt 1.1) nicht adäquat festgelegt worden sind.

Von vorhandener Validität wird gesprochen, wenn das „Ziel der Modellbildung — eine bzgl. des Modellzwecks HINREICHEND GENAUE Abbildung eines Realitätsausschnitts“⁷ erreicht wird. In einer ausführlicheren Darstellung des Modellzwecks werden in [Uth94] S. 55 die in einer Anforderungsdefinition üblichen Punkte aufgezählt.

6.1.3 Validität in den Sozialwissenschaften

Im Abschnitt 4.2.2 sind die Begriffe Indikatoren- und Konzeptvalidität eingeführt worden. Erstere fragt danach, ob die erhobenen Daten etwas mit der gewünschten Dimension zu tun haben, während bei letzterer zu prüfen ist, ob das theoretische Konstrukt eine Entsprechung in der Realität hat.

Im Gegensatz zur Indikatorenvalidität, die auf Objektebene darauf zielt, ob ein Zusammenhang zwischen der latenten Dimension und dem Verhalten existiert, beschäftigt sich das Standardkonzept der INHALTSVALIDITÄT damit, ob auf der sprachlich-instrumentellen Ebene die Operationalisierung geglückt ist. Die Inhaltsvalidität ist die theoretische Entsprechung der Indikatorenvalidität und würde in Abbildung 4.4 zwischen der theoretischen Ebene der Konstrukte und der Operationalisierungsebene angegeben werden.

6.2 Validität im Koordinatensystem

Zunächst sollen drei grundlegende Begriffe von einander abgegrenzt werden. In Kapitel 4.2.1 ist die VALIDIERUNG als Revisionsprozess der KALIBRIERUNG eingeführt worden. Dementsprechend soll die VALIDITÄT als das (gewünschte) Ergebnis des PROZESSES DER VALIDIERUNG bezeichnet werden.

Des Weiteren ist die VERIFIKATION von der VALIDIERUNG zu unterscheiden. Als Revisionsprozess der Operationalisierung fragt die Verifikation danach, ob das gedachte oder formale Systemmodell richtig in ein Simulationsmodell übertragen worden ist. Dies hat zur Konsequenz, dass man entweder wie im Software Engineering einzelne Teilaspekte der Implementation auf Korrektheit testen muss und/oder im Zuge der Validierung Systemdaten und Modelldaten vergleichen muss und ggf. eine falsche Implementierung als Fehlerquelle in Betracht ziehen muss.

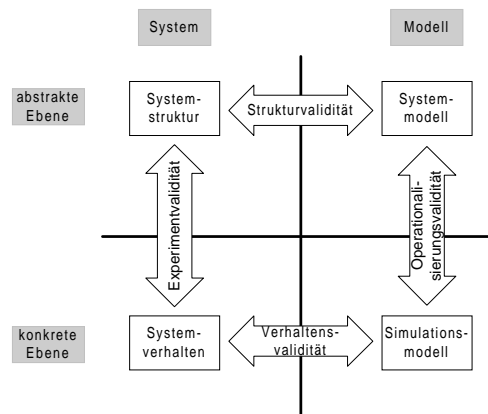


Abbildung 6.2: Validitätsarten im Koordinatensystem

6.2.1 Begriffsbildung

Aus dem allgemeinen Koordinatensystem in Abbildung 4.3 und in Anlehnung an die gerade genannten Begriffe werden gemäß Abbildung 6.2 verschiedene Arten von Validität eingeführt. Diese müssen sich je nach Einsatzort des Koordinatensystems weiter konkretisieren.

1. **Experimentvalidität:** Hier ist die Frage zu beantworten, ob und wie die Beobachtungen überhaupt mit dem zu untersuchenden Phänomen in Beziehung stehen, bzw. ob bei begrifflichen Konstrukten die vorgenommenen Messungen mit der entsprechenden realen Disposition in Verbindung stehen. (vgl. Indikatorenvalidität im Abschnitt 4.2.2).
2. **Struktur- oder Konzeptvalidität:** Für dingliche Objekte ist hier die Frage zu beantworten, ob das Systemmodell im Sinne des definierten Modellzwecks die zugrunde liegenden Objekte und deren Relationen zueinander richtig abbildet. Bei begrifflichen Konstrukten muss gefragt werden, ob es in der Realität auch Entsprechungen gibt.
3. **Operationalisierungsvalidität:** Hier ist entscheidend, welches Systemmodell der Operationalisierung zugrunde gelegen hat. Legt man zum Beispiel ein unformales Systemmodell zugrunde, so ist dessen Formalisierung (z.B. in Gleichungen) eine Operationalisierung. Für ein formalisiertes Systemmodell wird man dessen Implementation als Operationalisierung betrachten. Auf jeden Fall ist hier auch die Frage zu beantworten, ob der Übergang vom System- zum Simulationsmodell korrekt war. Insofern ist die Operationalisierungsvalidität auch Ergebnis einer Verifikation.
4. **Verhaltensvalidität:** „Stimmt das Modellverhalten nach festgelegten Kriterien mit dem Systemverhalten überein?“ Dies ist die Hauptfrage bei der Verhaltensvalidität.

⁷Vgl. [PLM97] S. 46 — Hervorhebung im Original.

Dabei werden die Entscheidungskriterien nach qualitativen, quantitativen und kontextuellen Aspekten unterteilt. Hier ist auch zu fragen, ob die Kalibrierung im Sinne einer Lösung der Kalibrierungsaufgabe gültige Ergebnisse liefert.

Es liegt nahe, von einem VALIDEN Modell zu sprechen, wenn je nach Modellierungsart alle anvisierten Validitätsarten gegeben sind. Dies setzt selbstverständlich gescheiterte Falsifikationsversuche voraus. Dennoch kann ein Modell stets nur als vorläufig gültig erachtet werden. Möglicherweise ist ein späterer Falsifikationsversuch erfolgreich.

6.2.2 Integration der Ansätze in der Literatur

Die vier im Koordinatensystem gefundenen Arten von Validität werden nacheinander darauf hin überprüft, ob sie mit den aus der Literatur gefundenen Ansätzen übereinstimmen.

Verhaltensvalidität. Schon der Begriff zeigt, dass bei Bossel und im Koordinatensystem hier in die gleiche Richtung gezielt wird. Allerdings hebt Bossel auf den QUALITATIVEN Aspekt ab, der sich damit begnügt, Phänomene zu beschreiben. Dies setzt im Allgemeinen eine qualitative Kalibrierung voraus. Damit ist etwa eine bestimmte Relation zwischen zwei Parametern gemeint.

Für eine exakte quantitative Untersuchung der Gültigkeit führt Bossel den zusätzlichen Begriff der empirischen Gültigkeit ein. Hierzu muss in der Anforderungsdefinition festgelegt werden, wie und in welchem Sinn genau die Ergebnisse übereinstimmen müssen.

Dies führt zum Begriff der Anwendungsgültigkeit. Zu deren Integration in das Koordinatensystem ist ein Wechsel des zugrunde liegenden Systembegriffs notwendig. Man hat die Systemumwelt im Sinne eines Experimentierfeldes in die Systemdefinition aufzunehmen. Dann handelt es sich um eine Unterart der Verhaltensvalidität.

Es ist schon darauf hingewiesen worden, dass Punkt 1 bei Uthmann zwei Aspekte enthält. Tatsächlich kann nur eine falsche Parametereinstellung der Verhaltensvalidität zugeordnet werden. Der Modellentwurf findet auf einer abstrakteren Ebene statt. Ansonsten geht Uthmann auf Verhaltensvalidität insofern ein, als auf die Anforderungsdefinition verwiesen wird.

Aus diesem Vergleich muss gefolgert werden, dass Verhaltensvalidität ein recht grober Begriff ist, der sich in der Praxis in mindestens vier Aspekte unterteilt:

1. **Qualität:** Die Modellphänomene und die Systemphänomene müssen übereinstimmen.
2. **Quantität:** Die Simulationsergebnisse müssen hinreichend genau sein.
3. **Kalibrierung:** Die Kalibrierungsaufgabe muss gelöst sein.
4. **Kontext:** Die Resultate müssen der Anforderungsdefinition, insbesondere dem Einsatzmodell, entsprechen.

Experimentvalidität. Während Bossel auf diesen Aspekt nicht eingeht, wird er bei Uthmann mit Punkt 3 hervorgehoben, der auf die Frage zielt, ob das Experiment im Zusammenhang mit dem gewünschten Objekt steht. Allerdings ist in der Praxis auch noch darauf zu achten, dass die mit dem Experiment verbundenen Messungen ungenau sein können und im Allgemeinen auch sind. Messfehler werden damit auch zu fehlender Experimentvalidität gezählt. Falter nennt die Experimentvalidität Indikatorenvalidität.

Strukturvalidität. Bei Uthmann ist eine Revision des Modellentwurfs in Punkt 1 ebenso hier einzuordnen wie Punkt 2, bei dem vorhandenes Wissen nicht mit abgebildet worden ist. Für begriffliche Objekte entspricht die Strukturvalidität der Konzeptvalidität.

Operationalisierungsvalidität. Auf diese Art von Validität geht keiner der Autoren ein. Möglicherweise wird eine richtige Operationalisierung im Sinne einer Verifikation vorausgesetzt. Leider ist es in der Praxis so, dass hier eine nicht unerhebliche Fehlerquelle liegt. Zur genaueren Unterscheidung der Operationalisierungsvalidität wird auf Abschnitt 2.3.1 über den Standardentwurf zurückverwiesen. Die Operationalisierung kann im Problembereich, dem Darstellungsmodell und dem Datenhintergrund scheitern. Im sozialwissenschaftlichen Kontext wird Operationalisierungsvalidität als Inhaltsvalidität bezeichnet.

Die Betrachtungen dieses Abschnitts zeigen, dass mit der Begriffsbildung des Koordinatensystems eine globale Sichtweise auf die Modellbildung ermöglicht wird, die dazu geeignet ist, die in der Literatur bekannten Arten von Validität zu integrieren und so dem Modellingenieur einen Leitfaden für sein Handeln zu geben. Im Abschnitt 9.4 wird das Koordinatensystem an einem Beispiel aus der Atmungsphysiologie auf die Wissensakquisition angewandt.

6.3 Validierung im Software Engineering

Die Begriffsbildung der Validitätsarten im Koordinatensystem bewährt sich nicht nur auf der abstrakten Ebene der Modellbildung. Dies wird hier wieder durch die Parallele zum Software Engineering gezeigt. Es ist also die Testphase im Software Engineering zu untersuchen und die Begriffe des Koordinatensystems darauf anzuwenden.

6.3.1 Begriffe

Unter einem Fehler versteht man im Software Engineering jede Abweichung des Verhaltens von dem in der Anforderungsdefinition festgelegten Verhalten. Mögliche Fehlerquellen sind die Spezifikation, der Entwurf, die Implementation oder eine fehlerhafte Eingabe. Die Techniken der Fehlersuche teilen sich in statische und dynamische Methoden auf. Zu den statischen Verfahren zählen die formale Verifikation (z.B. die Zusicherungsmethode) und die Analyse des Quellcodes durch dessen Inspektion.

Bei dynamischen Tests wird das Verhalten des zu untersuchenden Programms durch Eingabe von Testfällen geprüft. Dabei unterscheidet man zwischen Blackbox- und Whitebox-Tests. Im ersten Fall werden die Testfälle aufgrund des in der Anforderungsdefinition

festgelegten Verhaltens ausgewählt. Bei den Whitebox-Tests kennt man zusätzlich den Quellcode. Die Testfälle werden hier so konstruiert, dass bestimmte Codezeilen auch erreicht und durchlaufen werden.

6.3.2 Anwendung des Koordinatensystems

Ausgangspunkt ist die Tatsache, dass in der Anforderungsdefinition ein Systemmodell erstellt worden ist, das über den Entwurf und die Implementation in ein Simulationsmodell überführt wird, und so das Koordinatensystem in die gängigen Lebenszyklen integriert wird (vgl. Abschnitt 4.3).

Somit ist unter fehlender Konzept- oder Strukturvalidität im Kontext des Software Engineering zu verstehen, dass die Anforderungen an das Softwaresystem nicht geeignet spezifiziert worden sind. Nicht ohne Grund hat sich mit dem Requirement Engineering eine entsprechende Forschungsrichtung etabliert. Die weitaus meisten Fehlerquellen findet man in der Operationalisierungsphase. Hier können sowohl der Entwurf als auch dessen Implementation fehlerhaft sein. Bei fehlerhaften Eingaben als Fehlerquelle wird man von fehlender Verhaltensgültigkeit sprechen. Somit finden sich die möglichen Fehlerquellen auch im Koordinatensystem.

Entsprechend können die Testtechniken verschiedenen Bereichen zugeordnet werden. Die statischen sind auch in der hier vorgenommenen Begriffsbildung Verifikationsmethoden. Es wird geprüft, ob das Modell (Programm) richtig erstellt worden ist. Mit dynamischen Tests versucht man, das Programm zu validieren. Diese Parallelität ist von grundsätzlicher Bedeutung. Es ist nicht möglich, durch Testen die Fehlerfreiheit zu beweisen. Im Sinne einer fehlenden Falsifikation kann man höchstens davon sprechen, dass sich das Modell bewährt hat. Leider endet hier die Parallele und die Bedeutungen sind genau umgekehrt: Während in der entwickelten Terminologie eine Whitebox-Modellierung Voraussetzung für Strukturgültigkeit ist, sind es im Software Engineering gerade die Blackbox-Tests, die auf die Strukturgültigkeit abzielen.

Schließlich erlaubt die Unterscheidung zwischen Modul- und Integrationstests im Software Engineering eine einfache Interpretation von Abstraktionen im Lebenszyklus des Koordinatensystems.⁸

Die Module werden mit den konkreten Theorien in der vorherigen Darstellung identifiziert. Die abstraktere Theorie entspricht der Bestimmung der Interaktion der Module. Operationalisierung ist hier die Integration der Module. Ein Fehlverhalten bei der Validierung durch Testfälle im Integrationstest sollte nun aus der fehlerhaften Modellierung der Interaktion der Module resultieren. Dies kann aber nicht mit Sicherheit gesagt werden, da die Fehlerfreiheit der einzelnen Module nur durch Test überprüft und nicht bewiesen worden ist.

⁸Vgl. Abschnitt 4.2.3.

Teil III

Die Validierung am Beispiel der Atmungsphysiologie

Im folgenden Teil wird ein Modell der Atmungsphysiologie als Entscheidungshilfe in einem Expertensystem zur automatischen Entwöhnung von beatmeten Patienten beschrieben. Damit erfolgt die Überprüfung der Theorie der beiden ersten Teile durch DIE VALIDIERUNG AM BEISPIEL DER ATMUNGSPHYSIOLOGIE.

Im Sinne eines informalen Systemmodells werden zunächst die notwendigen (medizinischen) Zusammenhänge über die Systemobjekte EWS UND ATMUNGSPHYSIOLOGIE soweit dargestellt, wie es für das zu entwickelnde Modell und dessen Realisierung von Belang ist.

Die genannten medizinischen Grundlagen werden im Kapitel über DAS PHYSIOLOGISCHE MODELL operationalisiert. Der Text ist dabei nach den Hauptprozessen der Atmung untergliedert: Zunächst wird die Ventilation aus der Umwelt in die Lunge, anschließend die Perfusion der Lungenkapillaren modelliert. Schließlich folgt die Modellierung der Diffusionsprozesse. Das Kapitel schließt mit einer Beschreibung der Algorithmen, die im Zusammenhang mit dem physiologischen Modell entwickelt werden mussten.

ASPEKTE DER VALIDIERUNG stehen am Ende dieses Teils und der gesamten Arbeit. Es wird das Modell der Atmungsphysiologie soweit wie möglich überprüft und auch dazu verwendet, Aussagen der Arbeit aus den übrigen Teilen zu überprüfen. Hierzu wird zunächst die Implementation des physiologischen Modells in der im zweiten Teil vorgestellten generischen Struktur erläutert. Anschließend werden entlang den Daten eines Patienten typische Verhaltensphänomene gezeigt und schließlich wird das Koordinatensystem der Modellbildung als Lebenszyklus auf einen Teilaspekt des physiologischen Modells angewandt.

Kapitel 7

EWS und Atmungsphysiologie

Die Atmungsphysiologie wird soweit dargestellt, wie es für das Verständnis des zu entwickelnden physiologischen Modells von Belang ist.

Allgemein versteht man unter Atmung den Gasaustausch zwischen der Umwelt und den Zellen. Genauer werden die äußere und innere Atmung unterschieden.. Äußere Atmung bezeichnet den Gastransport aus der Umwelt über die Atemwege in die Lunge. Dabei fließt bei der Inspiration die Luft durch Mund oder Nase über die Luftröhre, die sog. TRACHEA und die baumartig verzweigten Bronchien und Bronchiolen in die ALVEOLEN. Diesen Vorgang nennt man VENTILATION. Zwischen den Alveolen und den bluttransportierenden KAPILLARGEFÄSSEN findet der Übergang von Sauerstoff ins Blut und von Kohlendioxid aus dem Blut in die Alveolarluft statt, die sog. DIFFUSION. Im weiteren Prozess, der PERFUSION, werden das Blut und die darin enthaltenen Gase abtransportiert. Zwischen den Gewebekapillaren und den Zellen findet eine erneute Diffusion und chemische Reaktionen statt. Dies nennt man innere Atmung.

Im Folgenden werden die einzelnen Prozesse Ventilation, Perfusion und Diffusion genauer dargestellt. Am Anfang steht ein kurzer Abriss des zu entwickelnden Gesamtsystems.

7.1 EWS — Das Evita-Weaning-System

7.1.1 Das EWS-Projekt

Diese Arbeit entsteht im Rahmen des EWS-PROJEKTES. Das Akronym EWS steht für *Evita-Weaning-System*. Dies wird von der Klinik für Anästhesiologie und dem Institut für Informatik der Johannes Gutenberg-Universität Mainz mit Unterstützung der Drägerwerk AG, Lübeck, entwickelt. Ausgangspunkt ist der Begriff des WEANINGS (Entwöhnung). Dies bedeutet eine schrittweise Verminderung der Atemhilfe durch das Beatmungsgerät, bis der Patient selbstständig atmet und vom Ventilator getrennt werden kann.

Idealerweise sollte der Patient so schnell wie möglich vom Beatmungsgerät getrennt werden, da die künstliche Beatmung zu Komplikationen wie z.B. Lungenentzündungen führen kann. Eine schnelle Entwöhnung führt zu kürzeren Aufenthaltszeiten auf der Intensivstation und damit zu Kosteneinsparungen. Bisher werden Patienten dadurch entwöhnt, dass

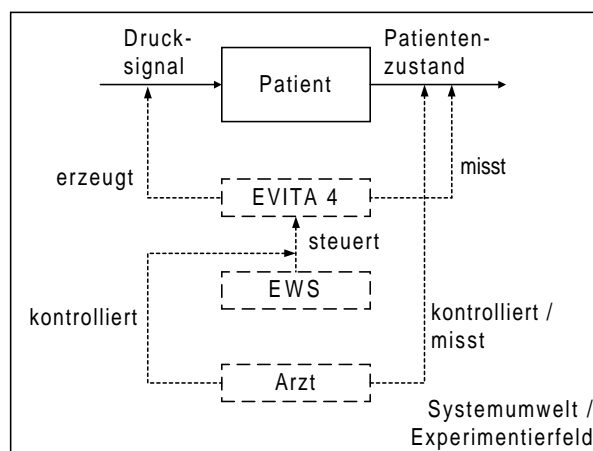


Abbildung 7.1: Systemkomponenten im Zusammenhang des EWS

die Beatmungsparameter durch den Arzt manuell angepasst werden, was zum Teil einen hohen Zeit- und Kostenaufwand erforderlich macht.

Aus diesen Gründen versucht man die Entwöhnung zu automatisieren. Erste Ansätze eines closed loop-Reglers stammen aus den frühen siebziger Jahren von Coles und seinen Mitarbeitern (vgl. [CBL73]). In der Folge haben auch verschiedene andere Gruppen versucht, die Entwöhnung zu automatisieren. Klinische Relevanz haben diese Systeme nach Weiler jedoch nicht.¹ In seiner Arbeit entwickelt Weiler mit der ADAPTIVE LUNG VENTILATION ein rückgekoppeltes Beatmungssystem mit den Zielen:

1. Die therapeutische Effizienz, die Sicherheit und den Bedienungsaufwand des neuen Systems im Vergleich mit einer konventionellen Beatmungstherapie ist zu untersuchen.
2. Es ist zu evaluieren, ob eine rückgekoppelte Regelung von Beatmungsparametern zu einem Fortschritt in der Beatmungstherapie führen kann.²

Es gelingt Weiler in seiner klinischen Studie zu zeigen, dass „der neue ALV-Regler der konventionellen Beatmungstherapie hinsichtlich der therapeutischen Effizienz, der Sicherheit und der Einfachheit der Handhabung überlegen war.“ Weilers Arbeit ist mindestens personell der Vorläufer des EWS-Projektes und führt in ihren abschließenden Worten fast unmittelbar in die vorliegende Problematik: „Aufgabe der nächsten Jahren muss es sein, Messmethoden und Modellbildung zu verfeinern, Expertenwissen und Strategien weiterzuentwickeln und die Systeme dann in kommerzielle Beatmungsgeräte zu integrieren, um ihren routinemäßigen Einsatz zu ermöglichen. Erst in diesem Stadium wird sich ihre potenzielle Überlegenheit endgültig nachweisen lassen.“³

Daher unterstützt die Firma Drägerwerk AG, Lübeck, die Entwicklung eines automatischen Entwöhnungssystems für den Respirator EVITA 4. In der zu entwickelnden Version

¹Vgl. [Wei98] S. 13 f.

²Vgl. [Wei98] S. 16.

³Vgl. [Wei98] S. 93.

handelt es sich um ein System, das auf einem PC arbeitet und mittels Remote Control mit dem Beatmungsgerät kommuniziert und dort in regelmäßigen Abständen die Beatmungsparameter einstellt. Abbildung 7.1 zeigt das Zusammenspiel der beteiligten Komponenten: Eine erste sinnvolle Systemgrenze liefert der Patient. In seiner Systemumwelt befinden sich das Beatmungsgerät, das Expertensystem im EWS und der Arzt. Der Respirator steuert die Atmung über ein zeitabhängiges Drucksignal und erhält Messwerte über den Patientenzustand. Im Zusammenhang mit dem EWS-Projekt wird das Beatmungsgerät vom Expertensystem gesteuert, das seinerseits vom Arzt kontrolliert wird. Für eine genauere Darstellung der Gesamtarchitektur vergleiche man [SG00].

Das automatische Entwöhnungssystem wird als regelbasiertes Expertensystem entwickelt, in das ein physiologisches Modell integriert ist, was ursprünglich nicht vorgesehen war. Allerdings ist das Problem aufgetreten, dass nicht alle Messwerte, die für die Beurteilung des Patientenzustandes notwendig sind, immer zur Verfügung stehen. So können beispielsweise Blutgasanalysen⁴ nicht ständig am Patienten durchgeführt werden, da dies stets mit einer Blutabnahme verbunden ist. Der erste Versuch, diese Werte innerhalb des Expertensystems mit Hilfe von „Wenn-Dann-Regeln“ abzuschätzen, hat sich als schwierig und aufwändig erwiesen. Daher haben sich die Mitarbeiter des Projektes entschieden, ein Modell der Atmungsphysiologie in das EWS zu integrieren. Das Modell soll u.a. eine Abschätzung der Blutgase liefern und stellt den Ausgangspunkt der Arbeit dar.

Aufgrund des vom Beatmungsgerät an das EWS gemeldeten Zustandes des Patienten und von Simulationswerten des physiologischen Modells entscheidet das Expertensystem mit Hilfe von Regeln, wie die Beatmungsparameter einzustellen sind.

7.1.2 Beatmung im Modus BIPAP-ASB am Respirator EVITA 4

Im Rahmen des EWS wird davon ausgegangen, dass der Patient im Beatmungsmodus BIPAP-ASB am Respirator EVITA 4 der Firma Drägerwerk AG beatmet wird. Daher genügt es für die Systembeschreibung, die Begriffe nur so weit zu entwickeln, dass dieser Beatmungsmodus definiert werden kann.

Man unterscheidet zwischen kontrollierter und assistierter Beatmung. Kontrollierte Beatmung ist dadurch gekennzeichnet, dass die Steuerung der Atemphasen, Inspiration und Expiration, ohne Einfluss des Patienten durch mögliche Eigenatmung erfolgt. Im Gegensatz dazu spricht man von assistierter Beatmung, wenn der Patient durch Eigenaktivität die Atemunterstützung des Respirators auslösen kann.

In beiden Formen wird jeweils zwischen volumenkontrollierter (bzw. volumenunterstützter) und druckkontrollierter (bzw. druckunterstützter) Beatmung unterschieden. Bei volumenkontrollierter Beatmung wird der Volumenfluss in den Patienten kontrolliert, so dass er dieses vorgegebene Volumen auch stets erhält. Bei druckkontrollierter Beatmung wird ein Druck am Respirator vorgegeben. Damit ist das applizierte Volumen eine abhängige variable Größe. Beim BIPAP-ASB-Verfahren handelt es sich um die Kombination aus einer druckkontrollierten und einer druckunterstützten Beatmungsform.

⁴Das ist die Bestimmung der arteriellen Sauerstoff- und Kohlendioxidpartialdrücken.

1. **BIPAP — Biphasic Positive Airway Pressure** kann als simultane Mischung von Spontanatmung und zeitgesteuerter, druckkontrollierter Beatmung definiert werden. In diesem Verfahren wird für zwei Zeitintervalle T_I und T_E jeweils ein Druck vorgegeben. Hervorzuheben ist beim BIPAP-Verfahren, dass der Patient auch die Möglichkeit hat, spontan zu atmen: Bringt er einen höheren Druck in der Lunge auf, als am Respirator anliegt, so expiriert der Patient.

Das obere Druckniveau bezeichnen wir mit P_{Insp} und das untere als PEEP („Positive Endexpiratory Pressure“). Der PEEP führt im Allgemeinen dazu, dass beatmete Patienten nicht das dem Druck der Umwelt entsprechende Volumen ausatmen, sondern nur soviel, bis in der Lunge der PEEP-Druck erreicht ist.⁵ Technisch wird dies so gelöst, dass die Ausatemluft über ein PEEP-VENTIL gelenkt wird, das den PEEP-Druck aufrecht erhält.⁶

2. **ASB — Assisted Spontaneous Breathing** wird definiert als assistierte, druckunterstützte, flussgesteuerte Atemhilfe. In der Praxis heißt das, dass der Respirator feststellt, wenn vom Patienten ein Atemzug durch Eigenaktivität eingeleitet wird. Dabei wird die Druckunterstützung begonnen, wenn

- (a) der spontane Inspirationsfluss den eingestellten Wert des FLOWTRIGGERS erreicht oder
- (b) das spontan eingeatmete Volumen 25 ml übersteigt.

Dann erhöht das Gerät den aktuellen PEEP-Druck auf einen ASB-Druck P_{ASB} . Dieser Druck wird solange gehalten

- (a) bis der Inspirationsfluss auf 0 zurückgeht oder
- (b) bis der Inspirationsfluss 25 % des Maximalfluss unterschritten hat
- (c) oder spätestens nach 4 Sekunden, wenn die beiden anderen Kriterien nicht wirksam wurden.⁷

Dies führt dazu, dass für ASB-ATEMZÜGE keine Inspirationszeit festgelegt werden kann.

Am Respirator EVITA 4 ist das BIPAP-ASB-Verfahren folgendermaßen realisiert: Der Arzt stellt die drei Beatmungsdrücke P_{Insp} , P_{ASB} und PEEP ein und gibt eine maschinelle Inspirationszeit sowie eine maschinelle Frequenz vor. Daraus ergibt sich die Expirationszeit T_E , da sich die Atemfrequenz als $f = \frac{60}{T_I + T_E}$ errechnet.

Leitet der Patient einen eigenen Atemzug in einem vorgegebenen Zeitfenster vor einem eigentlich maschinellen Atemzug ein, so entsteht die Frage, ob das Gerät einen BIPAP- oder einen ASB-Atemzug durchführt. An der EVITA 4 ist das Problem derart gelöst, dass ein BIPAP-Atemzug eingeleitet wird, dieser aber als ein sog. MANDATORISCHER ATEMZUG

⁵Eine therapeutische Diskussion des PEEP findet man beispielsweise in [OWA96] S. 99 f.

⁶Vgl. [OWA96] S. 124.

⁷vgl. [OWA96] S. 119 f.

gezählt wird. An dieser Stelle ergibt sich auch, dass ein Unterschied zwischen der eigentlich eingestellten maschinellen und der gemessenen Frequenz besteht. Die Dauer eines ASB-Atemzugs wird intern im Beatmungsgerät gemessen. Schließlich kann der Arzt eine RAMPENZEIT einstellen. Innerhalb dieser Zeit hebt das Beatmungsgerät den Druck vom PEEP-Niveau auf das Niveau der Inspiration an. Ein weiterer einzustellender Parameter ist die inspiratorische Sauerstofffraktion, also der Anteil des Sauerstoffs am eingeatmeten Volumen.

7.2 Die Ventilation

Die Luft fließt durch die Trachea über die immer feiner werdenden Verzweigungen des Bronchialbaums in die Alveolen. Dieser besteht aus insgesamt 23 Verzweigungsgenerationen. Bis zur 16. Generation hat der Bronchialbaum ausschließlich Leitungsfunktion. Ab der 17. Verzweigungsgeneration finden sich Alveolen, in denen die Diffusion zwischen dem Atem und Blut stattfindet. Die Anzahl der Alveolen nimmt mit der Verzweigungsgeneration zu. In der 23. Generation enden die Verzweigungen in den sog. Alveolarsäcken. In den folgenden Abschnitten wird auf die Grundlagen der Ventilation genauer eingegangen.

7.2.1 Die Atmungsmechanik

Die Atmungsmechanik wird in zwei Schritten beschrieben. Nach einer qualitativen Einführung werden die Größen exakter quantifiziert und damit die Grundlagen für die spätere Modellierung gelegt.

Qualitative Beschreibung

Die Ventilation wird von der Druckdifferenz zwischen Umwelt und Lunge bestimmt. Dabei kann auf zwei Arten durch Erhöhung des Druckgradienten Fluss entstehen. Entweder erhöht man den Druck in der Umwelt oder er wird in der Lunge verringert. Die Druckerhöhung in der Umwelt ist die Grundlage der kontrollierten Beatmung. Im Folgenden werden die Vorgänge bei der Verringerung des Druckes in der Lunge genauer betrachtet.

Der Lungenraum, der THORAX, wird nach unten durch das Zwerchfell und nach vorne durch die Rippenbögen begrenzt. Das Zwerchfell ist eine Muskelplatte, die sich bei Kontraktion absenkt. Erschlafft der Muskel, hebt sich das Zwerchfell wieder. Die Lunge ist von sog. Pleuralblättern umgeben, die ihrerseits über eine dünne Flüssigkeitsschicht, dem sog. kapillaren Spalt, mit dem Thorax verbunden sind, so dass die Lunge den Volumenänderungen des Thorax folgt.

Durch das Absenken des Zwerchfelles erhöht sich das Volumen im Thorax. Es folgt eine Druckverringerng zwischen den Pleuralblättern der Lunge und dem Thorax. Der dort herrschende Druck wird INTRAPLEURALER DRUCK genannt. Da die Lunge den Volumenänderungen des Thorax folgt, führt das Absenken des Zwerchfelles auch zu einer Verringerung des Druckes in der Lunge, dem INTRAPULMONALEN DRUCK. Damit ist ei-

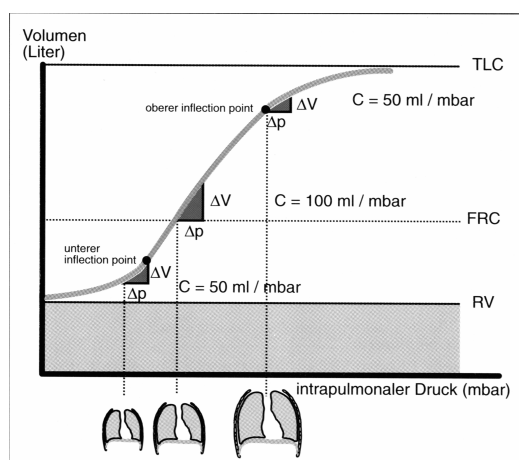


Abbildung 7.2: Eine Druck-Volumen-Kurve

8

ne Erhöhung des für die Ventilation verantwortlichen Druckgradienten verbunden. Die Expiration erfolgt im Allgemeinen passiv über das Erschlaffen des Zwerchfelles.

Quantitative Beschreibung

Grundlage der Quantifizierung sind die beiden atmungsmechanischen Größen: der Atemwegswiderstand, RESISTANCE, und die Dehnbarkeit des Gefäßes, COMPLIANCE. Vorstellen kann man sich diese Größen am Beispiel eines Luftballons: Die Resistance R gibt an, wie schwer es ist, die Öffnung des Luftballons zu überwinden. Die Compliance C gibt ein Maß an, wie sehr sich der Ballon ausdehnen lässt.

Genauer wird die Compliance als das Verhältnis von Volumen- zu Druckänderung definiert. Mathematisch formuliert ist die Compliance die Ableitung des Volumens nach dem Druck:

$$C = \frac{dV}{dP} \quad (7.1)$$

Trägt man in einem Druck-Volumen-Diagramm (Abbildung 7.2) das Volumen gegen den Druck auf, so erkennt man die Compliance als Steigung in jedem Punkt. In ihrem steilsten mittleren Bereich ist die Kurve annähernd linear, die Compliance damit näherungsweise konstant. Man kann in diesem Fall vereinfachend einen proportionalen Zusammenhang zwischen Druck P und Volumen V annehmen. Dies ist auch dadurch gerechtfertigt, dass es sich hier um den physiologischen Normalfall handelt.

$$P = \frac{V}{C} \quad (7.2)$$

Ähnlich wird die Resistance eingeführt. Nimmt man einen proportionalen Zusammenhang zwischen Druck und Volumenfluss pro Zeiteinheit an, wird der Widerstand der Atemwege als das Verhältnis der beiden zueinander definiert:

⁸Die Abbildung ist [OWA96] S. 23 entnommen.

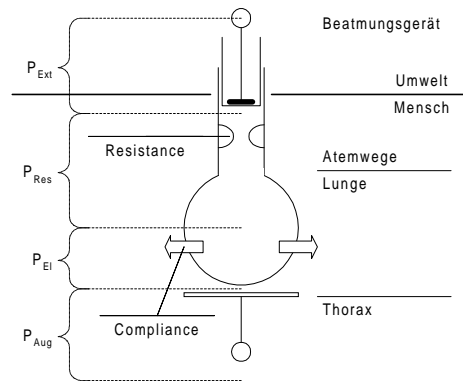


Abbildung 7.3: Schema eines Einkompartiment-Modells

$$R = \frac{P}{\dot{V}} \quad (7.3)$$

Betrachtet man die Zusammenhänge in einem EINKOMPARTIMENT-MODELL, in dem die gesamte Lunge eine einheitliche Atmungsmechanik und Dynamik (in dem hier zu definierenden Sinn) hat, kann man sich die Zusammenhänge in Abbildung 7.3 veranschaulichen. Eine Druckdifferenz entsteht durch einen externen Überdruck des Beatmungsgerätes P_{Ext} oder die Spontanaktivität über einen internen Unterdruck P_{Aug} durch Absenken des Thorax. Ferner entstehen in einem solchen Modell Drücke durch die Atemwegswiderstände P_{Res} und aufgrund der Compliance P_{El} , der sog. ELASTISCHE DRUCK. Im stabilen Zustand müssen sich die Drücke zu Null addieren, so dass gilt:⁹

$$P_{\text{Ext}} = P_{\text{El}} + P_{\text{Res}} + P_{\text{Aug}} \quad (7.4)$$

Aufgrund der angenommenen linearen Zusammenhänge bei Resistance und Compliance in den Gleichungen (7.3) und (7.2) ergibt sich:

$$\begin{aligned} P_{\text{Ext}} &= \frac{V}{C} + R\dot{V} + P_{\text{Aug}} \\ &\iff \\ \dot{V} &= \frac{1}{R} \left(P_{\text{Ext}} - \frac{V}{C} \right) - \frac{P_{\text{Aug}}}{R} \end{aligned} \quad (7.5)$$

Diese Gleichungen werden Grundlage der Modellierung Ventilation im physiologischen Modell sein.

Zusammen bilden die Resistance und die Compliance ein Maß für die Dynamik der Lunge. Dies erkennt man, wenn man die beiden Druckgeneratoren P_{Ext} und P_{Aug} zu einer zeitabhängigen Funktion $u(t)$ zusammenfasst. Dann wird Gleichung (7.5) zu der inhomogenen Differenzialgleichung

$$\dot{V} = -\frac{1}{RC}V + u(t) \quad (7.6)$$

⁹Dies entspricht der Kirchhoffschen Maschenregel im Anhang D.

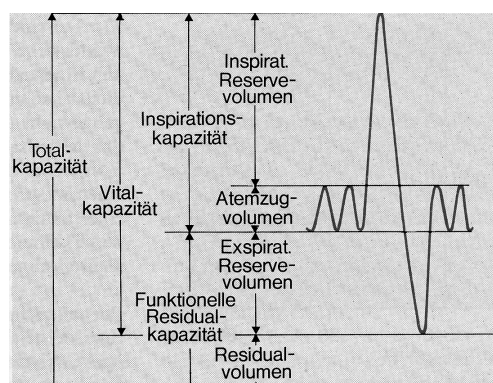


Abbildung 7.4: Aufteilung des Lungenvolumens

10

Die konkrete Form der Lösung hängt von der Inhomogenität $u(t)$ ab. Ihre Dynamik aber wird von der ZEITKONSTANTEN $\tau = RC$, dem Produkt aus Compliance und Resistance, bestimmt. Diese ist in unserem Zusammenhang von besonderer Bedeutung, da sie im EWS aus expiratorischen Daten bestimmt werden kann und so dem physiologischen Modell zur Verfügung steht. Im allgemeinen besitzen verschiedene Lungenareale unterschiedliche Zeitkonstanten. Daher fasst man die Lungenareale mit der selben Zeitkonstanten zu einem KOMPARTIMENT zusammen.

7.2.2 Aufteilung des Lungenvolumens

An „normalen“ Atemzügen ist nicht das gesamte Luftvolumen in der Lunge beteiligt. Vielmehr hat die Lunge sowohl inspiratorisch als expiratorisch noch Reserven. Die in der Medizin übliche Einteilung kann man in der Graphik 7.4 ablesen: Mit ATEMZUGVOLUMEN wird die Differenz aus inspiriertem und expiriertem Volumen bei einem „normalen“ Atemzug bezeichnet. Die Summe der Atemzugvolumina pro Minute wird ATEMMINUTEN-VOLUMEN genannt. Man unterscheidet je nach Typ des Atemzugs zwischen dem SPONTANEN und MASCHINELLEN Atemminuten-Volumen.

Für die Entwicklung des physiologischen Modells ist die Rolle der FUNKTIONELLEN RESIDUALKAPAZITÄT (KURZ: FRC) von Bedeutung. Aus der Graphik erkennt man, dass es sich bei der FRC um das Volumen handelt, das am Ende einer „normalen“ Expiration noch in der Lunge verbleibt. Wäre die FRC nicht vorhanden und würde die gesamte inspirierte Luft am Gasaustausch teilnehmen, so würden die relativen Anteile des Sauerstoffs und Kohlendioxids am Gesamtvolumen mit jeder Atemphase abwechselnd zu- oder abnehmen. Da sich aber die inspirierte Luft stets mit der der FRC mischt, kommt es nur zu geringen zeitlichen Schwankungen. Die FRC wirkt als Puffer.

7.2.3 Das Gasmisch

Die in der Atemluft enthaltenen Gase Sauerstoff und Kohlendioxid spielen bei den Diffusionsprozessen eine wesentliche Rolle. Deren Konzentration in der Atemluft muss im

¹⁰Die Abbildung ist [ST90] S. 570 entnommen.

physiologischen Modell berücksichtigt werden. Ausgangspunkt zu dieser Betrachtung ist das Gesetz von Avogadro.

Regel 7.1 (Gesetz von Avogadro). *Gleiche Raumeile von Gasen enthalten bei gleichem Druck und gleicher Temperatur gleichviele Teilchen.*

Die Anzahl der Teilchen, die sog. Stoffmenge, wird in der Einheit mol gemessen. Sie hängt bei einem festgelegten Volumen nach dem Gesetz von Avogadro vom Druck und der Temperatur ab. Möchte man also beispielsweise angeben, wie viel Sauerstoff in dem Volumen der inspirierten Luft ist, müsste man hierzu stets angeben, welcher Druck und welche Temperatur gerade herrschen. Dies hat dazu geführt, dass man drei Normbedingungen im Vergleich zu den herrschenden Bedingungen definiert, auf die die Angaben stets bezogen werden.¹¹

1. **STPD-Bedingungen:** (engl. **S**tandard **T**emperature **P**ressure **D**ry): Hier werden die Angaben auf eine Temperatur von $T = 273\text{ K}$, einen Druck von $P = 1013\text{ mbar}$ und einen Wasserdampfpartialdruck von $P_{\text{H}_2\text{O}} = 0\text{ mbar}$ bezogen. Betrachtet man unter diesen Bedingungen das Gesetz von Avogadro, führt dies auf den Begriff des molaren Normvolumens von $22,4141/\text{mol}$, das angibt, in wie vielen Litern sich ein Mol des Gases befinden.
2. **BTPS-Bedingungen:** (engl. **B**ody **T**emperature **P**ressure **S**aturated): Damit sind die Bedingungen gemeint, die in der Lunge herrschen. Man geht also von einer Temperatur von $T = 273\text{ K} + 37\text{ K} = 310\text{ K}$, dem herrschenden Barometerdruck und einem Wasserdampfpartialdruck $P_{\text{H}_2\text{O}} = 47\text{ mbar}$ aus.
3. **ATPS-Bedingungen:**¹²(engl. **A**mbient **T**emperature **P**ressure **S**aturated): Dies sind die aktuellen Messbedingungen außerhalb des Körpers in Bezug auf Temperatur und Druck. Hier geht man von 100 %-Wasserdampfsättigung aus.

Bei festgelegten Bedingungen kann eine Stoffmenge mit dem Volumen, das sie einnimmt, identifiziert werden. Dies führt auf den Begriff der FRAKTION: Sie gibt an, welcher Anteil des gesamten Gasvolumens ein bestimmter Stoff einnimmt. So ist beispielsweise in der „normalen“ Inspirationsluft unter STPD-Bedingungen eine Sauerstoff-Fraktion $F_{\text{I}}\text{O}_2$ von etwa 0,21 oder 21 Vol%. In BTPS-Bedingungen entspricht dies etwa 20 Vol%.

7.2.4 Ventilation in den seriellen Totraum

Ausgehend von der Tatsache, dass sich in der Atemluft je nach Atemphase unterschiedliche Fraktionen von Sauerstoff und Kohlendioxid befinden, betrachten wir den Prozess der Atmung in den leitenden Atemwegen genauer. Diesen Vorgang kann man sich auch am Beispiel eines Rohres vorstellen, durch das Luft geleitet wird, wie es Abbildung 7.5 andeutet.

¹¹Diese Definitionen sind im Wesentlichen [ST90] S. 593 entnommen.

¹²Diese Bedingungen werden im weiteren Verlauf der Arbeit nicht mehr benötigt und sind nur der Vollständigkeit halber genannt.

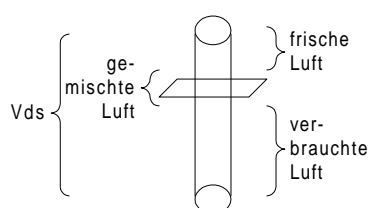


Abbildung 7.5: Vorstellung des seriellen Totraums als Rohr

Zu Beginn einer Expiration befindet sich in den leitenden Atemwegen, die man zusammenfassend als seriellen Totraum (kurz: V_{d_s}) bezeichnet und am Beatmungsgerät als ständiger Messwert zur Verfügung stehen, frische inspirierte Luft. Beim Ausatmen drückt die „verbrauchte“ Luft die „frische“ Luft im seriellen Totraum nach außen in die Umwelt, so dass zunächst die frische Luft expiriert wird. Diese beiden Luftschichten mischen sich im V_{d_s} kaum. Erst wenn die frische Luft expiriert ist, kann die verbrauchte Luft aus der Lunge ausgeatmet werden. Hat umgekehrt eine Inspiration begonnen, so enthält der V_{d_s} noch verbrauchte Luft vom vorherigen Atemzug. Durch das Einatmen wird also analog der Inspiration zunächst verbrauchte Luft aus dem letzten Atemzug in die Lunge geleitet. Diese Prozesse führen zu einem typischen Expirogramm, wie die folgende Abbildung 7.6 zeigt.

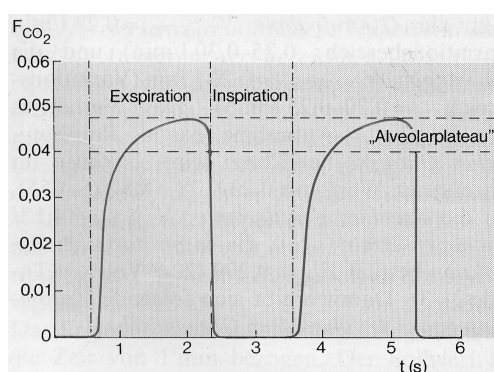


Abbildung 7.6: Ein typisches Expirogramm

13

Dort wird die expirierte Kohlendioxidfraktion gegen die Zeit aufgetragen. Man erkennt, dass zu Beginn der Expiration der Anteil des Kohlendioxids Null ist und später erst auf den Anteil steigt, der in den Alveolen herrscht und daher als ALVEOLARPLATEAU bezeichnet wird. Für die Modellierung der Physiologie sind diese Zusammenhänge deshalb von Belang, weil die endtidale Kohlendioxidfraktion am Ende eines Atemzugs, die sog. END-EXPIRATORISCHE KOHLENDIOXIDFRAKTION e_{tCO_2} , am Beatmungsgerät als Messwert zur Verfügung steht und Auskunft über Diffusion im Körper gibt.

7.3 Die Perfusion

Ausgehend von einer kurzen Darstellung der Zusammensetzung des Blutes werden die Transportarten des Sauerstoffs und des Kohlendioxids im Blut dargestellt.

¹³Das Diagramm ist aus [ST90] S. 594 entnommen.

7.3.1 Die Blutzusammensetzung

Von den verschiedenen Funktionen des Blutes ist für die hier zu modellierende Atmungsphysiologie vor allem dessen Transportfunktion von Belang.¹⁴ Im Blut sind Kohlendioxid und Sauerstoff sowohl physikalisch gelöst als auch chemisch gebunden. Mit dem Blut wird der Sauerstoff aus der Lunge in das Gewebe und das Kohlendioxid von dort wieder zurück transportiert. Wesentlichen Anteil an der Transportfunktion des Blutes haben dessen Hauptbestandteile, das PLASMA, eine proteinhaltige Flüssigkeit, und die darin suspendierten roten Blutzellen, die ERYTHROZYTEN.

1. **Blutplasma:** Das Blutplasma besteht zu 90–91 % aus Wasser. Ferner sind in einem Liter etwa 65 – 80 g Eiweiß enthalten. Diese Plasmaproteine dienen vor allem dazu, die Pufferfunktion des Blutes zu realisieren, d.h. einen relativ stabilen pH-Wert aufrecht zu erhalten.
2. **Erythrozyten:** Der Anteil der Erythrozyten im Blut wird als HÄMATOKRITWERT bezeichnet. Er beträgt beim erwachsenen Mann etwa 0,44 – 0,46 und bei der erwachsenen Frau 0,41 – 0,43. Die roten Blutkörperchen bestehen im Wesentlichen aus Wasser und Hämoglobin, dessen Molmasse 64500 g/mol beträgt. Dieses ermöglicht die chemische Bindung des transportierten Sauerstoffs im Blut.

7.3.2 Der Sauerstofftransport

Das in unserem Zusammenhang wichtige Wissen für den Sauerstofftransport im Blut ist in der folgenden Gleichung aggregiert:¹⁵

$$C_aO_2 = [\text{Hb}] * S_aO_2 * 1,34 + \tilde{\alpha} * P_aO_2 \quad (7.7)$$

Der Gesamtgehalt des Sauerstoffs in den Arterien C_aO_2 setzt sich aus dem ans Hämoglobin gebundenen (erster Summand) und dem physikalisch gelösten (zweiter Summand) Sauerstoff zusammen. Im folgenden Abschnitt werden die einzelnen Bestandteile der Gleichung schrittweise erläutert. Ferner wird ein weiterer Zusammenhang zwischen den drei Zielgrößen Gehalt C_aO_2 , Sättigung S_aO_2 und Partialdruck P_aO_2 über die Sauerstoff-Bindungskurve hergestellt.

Physikalisch gelöster Sauerstoff

Zunächst wird der zweite Teil der Gleichung betrachtet und erläutert, wie der Gehalt an physikalisch gelöstem Sauerstoff im Blut bestimmt wird. Dazu wird der Partialdruck eingeführt:

¹⁴Die Darstellung lehnt sich an [ST90] S. 422 f und S. 612 f an.

¹⁵Allgemein sind folgende Indizierungen üblich und werden auch hier zusammenfassend aufgelistet:

| Abkürzung | Bedeutung |
|-----------|----------------|
| a | arteriell |
| A | alveolar |
| \bar{v} | gemischt-venös |

Regel 7.2 (Daltonsches Partialdruck-Gesetz). *In einem Gemisch idealer, chemisch nicht miteinander reagierender Gase ist der Gesamtdruck gleich der Summe der Drücke der einzelnen Komponenten. Sie werden als PARTIALDRÜCKE bezeichnet.*

Obwohl das Partialdruck-Gesetz für Gase formuliert ist, kann man es mit Einschränkungen auf die Lösung von Gasen im Blut anwenden. Hierzu nimmt man an, dass die Gase im Blut äquilibriert, d.h. gleichmäßig verteilt, sind. Dann kann man dem Gas im Blut den gleichen Partialdruck zuordnen, den es als nicht gelöstes Gas hätte. Der Partialdruck eines Gases bestimmt zusammen mit dem sog. BUNSEN-LÖSLICHKEITSKOEFFIZIENTEN α dessen Gehalt. Dieser Zusammenhang wird in einem weiteren Gesetz von Dalton quantifiziert.

Regel 7.3 (Henry-Dalton-Gesetz). *Die Konzentration eines physikalisch gelösten Gases [Gas] ist gegeben durch*

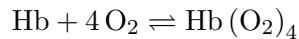
$$[\text{Gas}] = \frac{\alpha}{760} * P_{\text{Gas}}$$

Der Divisor 760 kommt dadurch zustande, dass von der Einheit ml/atm des Löslichkeitsfaktors in ml/mmHG umgerechnet werden muss, da der Partialdruck üblicherweise in mmHG gemessen wird. Dieser Umrechnungsfaktor ist durch die Bezeichnung $\tilde{\alpha}$ in die Ausgangsgleichung (7.7) integriert.

Der Löslichkeitskoeffizient hängt vom gelösten Gas, dem Lösungsmittel und der Temperatur ab. Für Blut von 37 Celsius beträgt der Löslichkeitskoeffizient $\alpha_{\text{O}_2} = 0,024$. Das heißt, es wird nur wenig Sauerstoff physikalisch im Blut gelöst.

Chemisch gebundener Sauerstoff

Die physikalische Lösung des Sauerstoffs im Blut ist Voraussetzung für dessen chemische Bindung an das Hämoglobin. Diese erfolgt über eine Reaktion mit dem Hämoglobin Hb in den Erythrozyten. Dabei gilt folgendes Reaktionsgleichgewicht:



Betrachtet man diese Gleichung unter STPD-Bedingungen (vgl.7.2.3) und würde das Reaktionsgleichgewicht ganz auf der rechten Seite liegen, so könnte 1 mol Hb $4 * 22,4141$ O₂ binden. Da 1 mol Hb 64500 g wiegt, kann 1 g Hb $4 * 22,414/64500 = 1,39$ ml/g O₂ an sich binden. In der Praxis liegt die gebundene Menge bei etwa 1,34 ml/g und wird als HÜFNER-ZAHL bezeichnet.

Da also nicht das gesamte Hämoglobin für die Bindung des Sauerstoffs zur Verfügung steht, wird der Begriff der Sauerstoffsättigung SO_2 eingeführt: Sie gibt an, welcher Prozentsatz des Hämoglobins als sog. Oxyhämoglobin, an das Sauerstoff angelagert ist, vorliegt.

Mit diesem Wissen ist jetzt auch der erste Summand der Gleichung zu erklären: Der Gehalt des chemisch gebundenen Sauerstoffs ergibt sich aus dem Produkt der Hüfner-Zahl, der Hämoglobinkonzentration und der arteriellen Sauerstoffsättigung als $1,34 * [\text{Hb}] * \text{S}_a\text{O}_2$. Somit folgt die zu Beginn des Abschnitts genannte Gleichung (7.7).

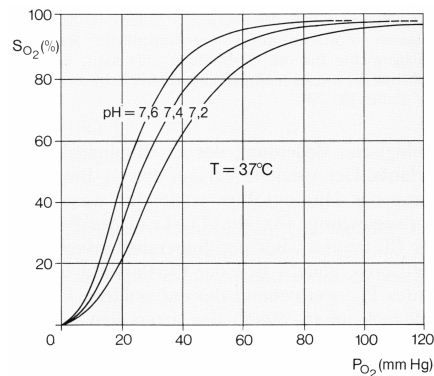


Abbildung 7.7: Sauerstoff-Bindungskurven mit verschiedenen pH-Werten

16

Die Sauerstoff-Bindungskurve

Die Sauerstoffsättigung hängt vom Partialdruck des Sauerstoffs ab: Trägt man die Sauerstoffsättigung gegen den Partialdruck auf, ergibt sich der charakteristische s-förmige Verlauf der Sauerstoff-Bindungskurve, den Abbildung 7.7 zeigt: Der genaue Verlauf der Kurve wird von mehreren Faktoren bestimmt. In Abbildung 7.7 ist mit dem pH-Wert eine wesentliche Einflussgröße ausgewählt worden. Mit sinkendem pH-Wert sinkt auch die Affinität des Sauerstoffs zum Hämoglobin, so dass ein höherer Druck aufgebracht werden muss, um die gleiche Sauerstoffsättigung zu erreichen. Ein niedriger pH-Wert führt also zu einem „Rechts-Shift“. Dieser Zusammenhang wird als BOHR-EFFEKT bezeichnet. Neben dem genannten pH-Wert beeinflussen die Temperatur und der Basenüberschuss¹⁷ den Verlauf der Kurve wesentlich. Daher kann man eine mehrdimensionale Tabelle anlegen und daraus den gemeinsamen Einfluss der genannten Faktoren auf die Sauerstoff-Bindungskurve ablesen, wie Abbildung E.3 im Anhang zeigt: Zunächst bestimmt man aus dem rechten Teil der Tabelle zu den gegebenen Temperatur-, Basenüberschuss- und pH-Werten die zugehörigen Faktoren, multipliziert diese zu einem Korrekturfaktor und bestimmt abschließend durch das Produkt des gemessenen Sauerstoffpartialdruckes mit dem erhaltenen Korrekturfaktor einen virtuellen Sauerstoff-Partialdruck. Die Sauerstoffsättigung des virtuellen Partialdruckes ist die zu den aktuell gemessenen Werten gehörende.

Abschließend wird festgestellt, dass es unter gewissen physiologischen Umständen, wie beispielsweise bestimmten Anämieformen, zu Änderungen der Transportbedingungen des Sauerstoffs kommen kann.¹⁸ Diese werden im physiologischen Modell unberücksichtigt bleiben. Allerdings können dann auch derartige Fälle nicht durch das physiologische Modell abgebildet werden.

¹⁶Die Abbildung ist [ST90] S. 618 entnommen.

¹⁷Damit wird die Abweichung vom Wert der Normalpufferbasen (48 mmol/l) im arteriellen Blut bezeichnet. Dieser Wert wird im Zuge einer Blutgasanalyse (vgl. Abschnitt 7.1.1) bestimmt.

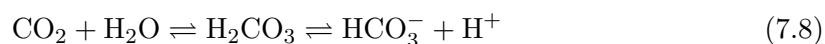
¹⁸Vgl. [ST90] S. 619 oder [Nun87] S.266f.

7.3.3 Der Kohlendioxidtransport

Die notwendigen Zusammenhänge des CO_2 -Transports im Blut werden überblicksartig dargestellt. Zunächst wird aufgezeigt, auf welche Arten Kohlendioxid transportiert werden kann, um anschließend die zugehörige Bindungskurve zu beschreiben. Sie führt auf den sog. Haldane-Effekt.

Arten des Kohlendioxidtransportes

Ein Großteil des vom Blut transportierten Kohlendioxids liegt in der Form von Hydrogencarbonat oder auch Bicarbonat (HCO_3^-) vor. Dieses kann aus einer Reaktion mit dem Wasser des Blutes entstehen:



Diese Reaktion erfolgt sehr langsam, so dass der Großteil der Umwandlung von Kohlendioxid zu HCO_3^- in den Erythrozyten erfolgt. In diesen ist das Enzym Carboanhydrase enthalten, das als Katalysator wirkt und so zu einer schnellen Produktion von Kohlensäure führt. Diese zerfällt in den Erythrozyten sofort wieder zu Bicarbonat und einem Wasserstoff-Ion. Das entstandene Bicarbonat wird entweder im Austausch gegen ein Chloranion an das Plasma abgegeben („Chlorverschiebung“) oder unter Sauerstoffabgabe an das Hämoglobin gebunden.¹⁹ Diese Reaktionen sind aus der Sicht der Bindung des Kohlendioxids im Blut beschrieben. In den Alveolen erfolgen sie unter Aufnahme von Sauerstoff in der umgekehrten Richtung. Abbildung 7.8 fasst dies zusammen. Insgesamt ergeben sich für das Kohlendioxid vier Transportmöglichkeiten im Blut, die in der folgenden Tabelle nochmals zusammengefasst werden:

| Stoff | Typ | Anteil | Ort |
|-------------------------------------|-------------------|--------|--------------|
| HCO_3^- | chemisch gebunden | 50% | Plasma |
| $\text{CO}_2/\text{H}_2\text{CO}_3$ | phys. gelöst | 12% | Plasma |
| Carbaminoverbindung am Hb | chemisch gebunden | 11% | Erythrozyten |
| HCO_3^- | chemisch gebunden | 27% | Erythrozyten |

Kohlendioxid-Bindungskurven

Wie der Sauerstoffgehalt ist auch der Kohlendioxidgehalt im Blut vom herrschenden Partialdruck abhängig. Diese Zusammenhänge sind in Abbildung E.2 im Anhang dargestellt. Hier sind ebenfalls die Anteile der unterschiedlichen Arten des Kohlendioxidtransportes, wie sie im Abschnitt zuvor erläutert worden sind, zu erkennen. Im Diagramm ist der CO_2 -Gehalt gegen den CO_2 -Partialdruck aufgetragen. Man erkennt, dass der Gehalt des Kohlendioxids mit steigendem Druck wächst, allerdings nicht gegen einen festen Sättigungsgrad, sondern im Prinzip unbeschränkt. Ferner unterscheiden sich die Kurven für das arterielle und das venöse Blut. Der Grund hierfür liegt darin, dass das mit Sauerstoff

¹⁹Vgl. [ST90] S. 621.

²⁰Die Abbildung ist [ST90] S. 621 entnommen.

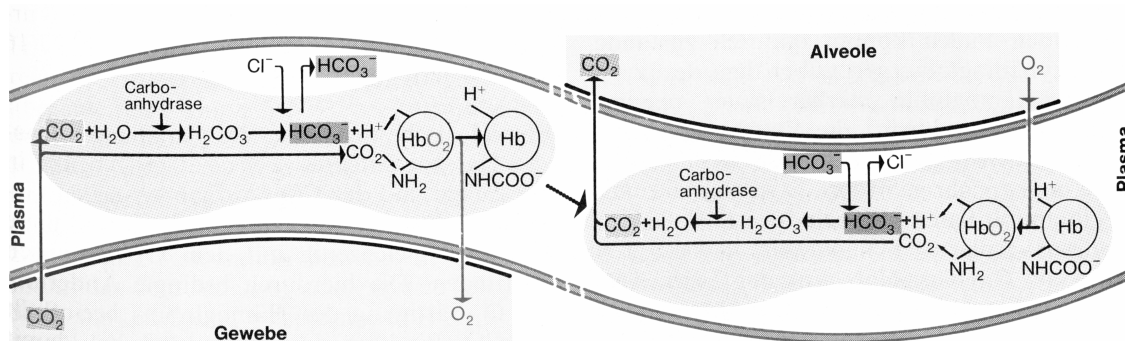


Abbildung 7.8: Zusammenfassung der Prozesse beim Gasaustausch in den Erythrozyten

20

versorgte arterielle Blut bzw. das in den Erythrozyten enthaltene Hämoglobin saurer reagiert. Umgekehrt wird im venösen Blut CO_2 leichter dissoziiert, so dass bei gleichem Druck ein höherer CO_2 -Gehalt erreicht werden kann. Diese Abhängigkeit der CO_2 -Bindungskurve vom Oxygenierungsgrad des Hämoglobins im Blut wird als HALDANE-EFFEKT bezeichnet.

7.4 Die Diffusion

Diffusionsprozesse finden sowohl zwischen den Alveolen in der Lunge und den kapillaren Blutgefäßen statt als auch zwischen den Blutgefäßen im Körper und den Sauerstoff verbrennenden Zellen. Zunächst wird in diesem Abschnitt die Gesetzmäßigkeiten des zuerst genannten Diffusionsprozesses beschrieben. Diesen nennt man ARTERIALISIERUNG und versteht darunter „die durch den pulmonalen Gasaustausch herbeigeführte Änderung der O_2 - und CO_2 -Partialdrücke.“²¹ Anschließend werden mögliche Störungen des Diffusionsprozesses erläutert, was zu den Definitionen der Lungenkompartimente SHUNT und TOTRAUM führt. In einer Zusammenfassung des ersten Diffusionsprozesses werden die Faktoren, die die Arterialisierung beeinflussen, durch Maßzahlen charakterisiert.

7.4.1 Arterialisierung

Abbildung 7.9 zeigt schematisch die ablaufenden Prozesse. Die Alveolen werden durch die Ventilation mit Luft gefüllt, die Kapillargefäße mit Blut perfundiert; zwischen beiden findet der Diffusionsprozess statt. Er wird von den verschiedenen Partialdrücken in den Alveolen und den Kapillargefäßen gesteuert. Während in den Alveolen ein hoher Sauerstoffpartialdruck herrscht (ca. 100 mmHG) beträgt er im venösen Blut nur ca. 40 mmHG. Das Kapillarblut ist hier von der Atemluft durch eine sehr dünne Gewebsschicht von einem $1 \mu\text{m}$ getrennt. Diese Gewebsschicht kann aufgrund der Druckdifferenzen des Sauerstoffs überwunden werden, so dass Sauerstoff ins Blut übergeht. Umgekehrt läuft der Prozess für das Kohlendioxid ab, da im Blut ein höherer Partialdruck als im Atem vorliegt. Quan-

²¹siehe [ST90] S. 598.

²²Die Abbildung ist aus [ST90] S. 599 entnommen.

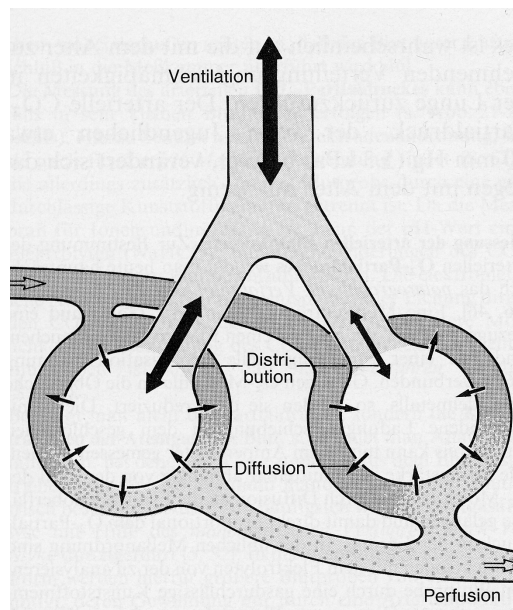


Abbildung 7.9: Prinzip des Zusammenspiels von Ventilation, Perfusion und Diffusion

22

tifiziert wird die Arterialisierung über das 1. FICKSCHE DIFFUSIONSGESETZ, das hier für gelöste Gase formuliert wird.

Regel 7.4 (1. Ficksches Diffusionsgesetz). ²³

Der Diffusionsstrom \dot{M} , d.h. die Substanzmenge, die durch eine Schicht der Fläche F und der Dicke d hindurchtritt, ist der wirksamen Partialdruck-Differenz direkt proportional.

$$\dot{M} = K \frac{F}{d} \Delta P$$

Der Faktor K wird als Krogh-Diffusionskoeffizient oder Diffusionsleitwert genannt. Er hängt von der Art der Teilchen und der Temperatur ab. Im Zusammenhang mit der Arterialisierung werden die drei Faktoren K, d und F zu einer einzigen Konstanten D_L für den Sauerstoff zusammengefasst. Sie wird als O₂-DIFFUSIONSKAPAZITÄT DER LUNGE bezeichnet. Da die Partialdrücke im gesamten Kapillar nicht gleich sind, wird zur Bestimmung von D_L die mittlere Partialdruck-Differenz $\overline{\Delta P_{O_2}}$ benutzt:

$$D_L = \frac{\dot{V}_{O_2}}{\overline{\Delta P_{O_2}}}$$

Tritt nun ein Erythrozyt für die Dauer von 0,3 Sekunden (die sog. Kontaktzeit) mit einem Kapillar in Kontakt, so gleichen sich durch die Diffusion die jeweiligen Partialdrücke von Kohlendioxid und Sauerstoff an, so dass am Ende im Blut der entsprechende Partialdruck der Alveole herrscht: „In der Lunge der Gesunden gleichen sich die Partialdrücke im Blut den alveolaren Werten praktisch vollständig an.“²⁴

²³Zitat nach [ST90] S. 596.

²⁴[ST90] S. 597; Hervorhebung im Original.

7.4.2 Venöse Beimischung: Shunt

Bisher sind wir davon ausgegangen, dass für den Diffusionsprozess sowohl das Kapillargefäß mit Blut perfundiert als auch die Alveole belüftet wird. Darüber hinaus gibt es auch venöses Blut, das nicht in Kontakt mit den Alveolen kommt und sich in den Arterien mit dem arterialisierten Blut mischt (vgl. Abbildung 7.10: Dort ist mit dem Totraum noch ein drittes Kompartiment enthalten). Das Resultat ist, dass sich in den Arterien nicht der alveolare Partialdruck einstellt. Die Differenz zwischen erwartetem arteriellen Partialdruck und tatsächlichem Partialdruck führt zum Konzept der venösen Beimischung. Sie ist nicht die „*eigentliche* Menge an venösem Blut, die sich mit dem arteriellen Blut vermischt, sondern die *geschätzte* Menge, die den beobachteten arteriellen Sauerstoffpartialdruck reproduziert.“²⁵ Bei der venösen Beimischung handelt es sich um ein theoretisches Konzept, während mit SHUNT das tatsächlich „kurzgeschlossene“ Blut gemeint ist. Es werden beide nicht weiter voneinander unterschieden und im Folgenden wird nur noch kurz von Shunt gesprochen.

Wie man diese Zusammenhänge quantifizieren kann, zeigt die folgende Beispiel-Rechnung. Sie beruht auf einem Modell mit zwei Kompartimenten, einem Normal- und einem Shunt-Kompartiment. Sie führt zu einer Gleichung, die man auch zur Definition des Shunts verwenden kann. Ferner zeigt die Rechnung, wie das Mischen von Blut aus verschiedenen Kompartimenten quantitativ modelliert werden kann. Der entscheidende Punkt hierbei ist, dass sich die verschiedene Gehalte addieren.

$$\dot{Q}_N + \dot{Q}_S = \dot{Q}_{\text{Total}} \quad (7.9)$$

$$C_{N'}\text{O}_2 * \dot{Q}_N + C_{\bar{v}}\text{O}_2 * \dot{Q}_S = C_a\text{O}_2 * \dot{Q}_{\text{Total}} \quad (7.10)$$

Dabei wird das pro Zeiteinheit vom Herzen in den Körper abgegebene Blutvolumen als HERZ-ZEIT-VOLUMEN (kurz: HZV) \dot{Q} bezeichnet. In dieser Rechnung teilt sich das gesamte HZV \dot{Q}_{Total} in die Anteile des Normal-Kompartimentes \dot{Q}_N und des Shunts \dot{Q}_S auf. Dementsprechend addieren sich Produkte aus dem jeweiligen Sauerstoffgehalt und dem HZV zum Produkt aus arteriellem Gehalt $C_a\text{O}_2$ und dem gesamten HZV. Im Normal-Kompartiment N wird der Gehalt am Ende des Kapillargefäßes — also nach der Arterialisierung — benutzt, worauf die Indizierung N' in $C_{N'}\text{O}_2$ hinweist. Im Shunt wird der gemischt-venöse Gehalt des Sauerstoffs $C_{\bar{v}}\text{O}_2$ benutzt. Daraus kann das Verhältnis des Herz-Zeit-Volumen des Shunts \dot{Q}_S zum gesamten HZV berechnet werden. Diese Gleichung wird auch zur Definition des Shunts benutzt:

$$\frac{\dot{Q}_S}{\dot{Q}_{\text{Total}}} = \frac{C_{N'}\text{O}_2 - C_a\text{O}_2}{C_{N'}\text{O}_2 - C_{\bar{v}}\text{O}_2}$$

Somit ergibt sich das Verhältnis von Shunt- zu gesamtem Herz-Zeit-Volumen als der Quotient aus der Differenz von alveolarem und arteriellem Sauerstoffgehalt und der Differenz von alveolarem und gemischt-venösem Sauerstoffgehalt.

²⁵Vgl. [Nun87] S. 167; Übersetzung des Autors mit vorhandener Hervorhebung im Original.

7.4.3 Nicht perfundierte Lungenareale: physiologischer Totraum

Der Diffusionsprozess kann derart gestört sein, dass das Lungenareal zwar belüftet aber die Kapillaren nicht perfundiert werden. Zunächst dienen die Bronchien bis zur 16. Verzweigungsgeneration nur dazu Luft weiterzuleiten und werden als serieller Totraum V_{d_s} bezeichnet. Erst ab der 17. Generation finden sich an den Atemwegen Alveolen, in denen der Diffusionsprozess stattfinden kann. Eine verminderte Perfusion der zugehörigen Kapillaren kann dazu führen, dass ein Teil der Alveolen nicht im Diffusionsprozess wirksam ist. In diesem Fall spricht man von ALVEOLAREM TOTRAUM $V_{d_{alv}}$. Die Summe aus alveolarem und seriellen Totraum wird als PHYSIOLOGISCHER TOTRAUM $V_{d_{phys}} = V_{d_s} + V_{d_{alv}}$ bezeichnet.

7.4.4 Zusammenfassung: Arterialisierungsfaktoren

Die „treibende Kraft“ bei der Diffusion ist die Partialdruckdifferenz zwischen dem kapillaren Blutgefäß und der Alveole. Beim Patienten mit gesunder Lunge reicht die Kontaktzeit zum Druckausgleich der beiden Systeme aus, so dass am Ende der Arterialisierung sich der Sauerstoffpartialdruck der Alveole im Blutgefäß eingestellt hat, während der Kohlendioxidpartialdruck des Blutgefäß sich auf die Alveole übertragen hat.

Wie zuvor dargelegt worden ist, kann dieser Diffusionsprozess auf zwei Arten gestört sein. Wird das Blut nicht abtransportiert, ist die Perfusion vermindert bzw. nicht vorhanden. Andererseits kann auch der Transport des Atems durch die Atemwege und somit die Ventilation gestört sein. Eine Klassifizierung des Diffusionsprozesses erfolgt mit Hilfe des Verhältnisses aus alveolaren Ventilation \dot{V}_{alv} zum Herz-Zeit-Volumen \dot{Q} , dem VENTILATIONS-PERFUSIONS-VERHÄLTNIS (kurz: VPV), ab:

$$VPV = \frac{\dot{V}_{alv}}{\dot{Q}} = \begin{cases} 0,8 & \text{normale Situation} \\ \infty & \text{keine Perfusion} \\ 0 & \text{keine Ventilation} \end{cases} \quad (7.11)$$

In den gesunden „normalen“ Lungenarealen beträgt das VPV 0,8. Auf der (unendlichen) Skala des VPV werden zwei Extremfälle unterschieden: Ist das Lungenareal nicht perfundiert und das $VPV = \infty$, so spricht man von TOTRAUM. Ist umgekehrt bei einem $VPV = 0$ keine Ventilation vorhanden, wird das entsprechende Lungenareal SHUNT genannt. So findet man in dieser Kennzahl die bereits eingeführten Kompartimente Shunt und Totraum wieder. Dazwischen sind entlang der Skala des VPV alle Abstufungen von Lungenarealen möglich.

Auch wenn sowohl Ventilation als auch Perfusion normal sind, kann die Diffusion gestört sein. Der Grund hierfür ist eine Verringerung der Diffusionskapazität der Lunge D_L (vgl. Abschnitt 7.4.1). Sie allein ist, wie VPV, nicht aussagekräftig. Vielmehr wird sie auf das Herz-Zeit-Volumen bezogen, so dass eine Abnahme des Quotienten D_L/\dot{Q} als Diffusionsstörung definiert werden kann. Gemäß dem Fickschen Diffusionsgesetz können bei kleinerer Diffusionskapazität weniger Sauerstoff-Teilchen diffundieren, so dass weniger Sauerstoff ins Blut gelangt. Man spricht in dieser Situation davon, dass die Diffusionswiderstände erhöht sind.

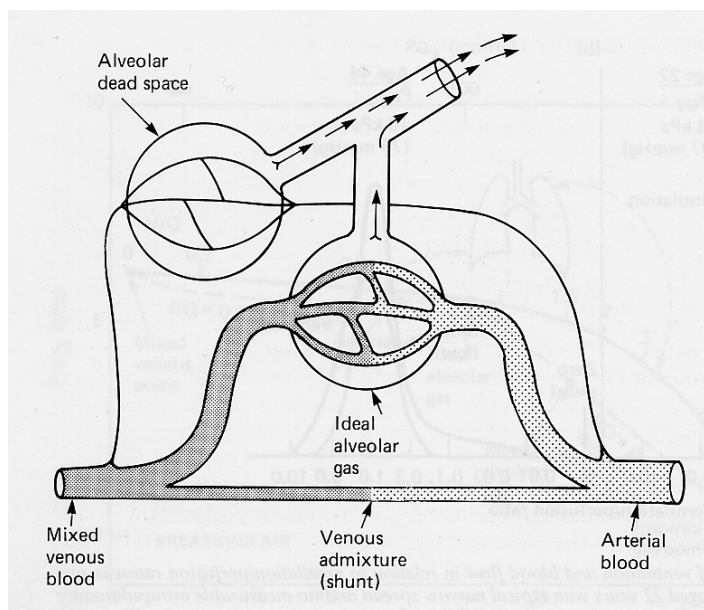


Abbildung 7.10: Das Dreikompartimente-Modell

26

7.5 Das Dreikompartimente-Modell

Fasst man, entsprechend Gleichung 7.11, die Lunge in drei Teile zusammen, entsteht ein DREIKOMPARTIMENTE-MODELL, wie es Abbildung 7.10 andeutet: Der alveolare Totraum (oben links zu sehen) wird belüftet, aber nicht perfundiert. Der untere Teil zeigt den nicht ventilerten Shunt, in dem die venöse Beimischung stattfindet. Nur im mittleren Teil findet man eine ideales Kompartiment, in dem das VPV etwa 0,8 beträgt. Den Anteil des Shunts und des Totraums an der gesamten Lunge bezeichnet man als SHUNT- BZW. TOTRAUMFRAKTION.

In diesen Kompartimenten herrschen unterschiedlich hohe Partialdrücke für Kohlendioxid und Sauerstoff. Erst wenn sich das „Lungenblut“ zum arteriellen Blut mischt, stellt sich der arterielle Partialdruck ein. Ist kein Shunt vorhanden, werden die Partialdrücke in den Kompartimenten und den Arterien im Wesentlichen identisch sein; davon kann aber im Zusammenhang mit dem EWS-Projekt nicht ausgegangen werden, da auch an der Lunge erkrankte Patienten von der Beatmung entwöhnt werden sollen. Dies ist insofern wichtig, als sich Normalwerte für die Blutgaswerte P_{aO_2} und P_{aCO_2} in der Physiologie auf den Alveolarraum beziehen.²⁷ Wir benutzen zur Definition der Normalwerte der Partialdrücke allerdings die Werte in den Arterien, da sich bei einer kranken Lunge mit Shunt möglicherweise die Werte in der Lunge und den Arterien unterscheiden. Somit legen wir als Normalwerte $P_{aO_2} = 100$ mmHG, $P_{aCO_2} = 40$ mmHG fest und definieren als NORMVENTILATION, diejenige Ventilation, die zu einem arteriellen Kohlendioxid-Partialdruck von 40 mmHG führt. Der Partialdruck des Kohlendioxids steuert die Atmungsarbeit der Patienten: Ist der Partialdruck des Kohlendioxids niedrig, atmet der Mensch nicht.

²⁶Die Abbildung ist aus [Nun87] S. 156 entnommen.

²⁷Vgl. [ST90] S. 595.

7.6 Die Gefäße

Der zweite Diffusionsprozess zwischen dem Blut und den Zellen wird wesentlich vom RESPIRATORISCHEN QUOTIENTEN RQ bestimmt. Er wird definiert als das Verhältnis aus CO₂-Produktion und O₂-Aufnahme der Zellen:²⁸

$$\text{RQ} = \frac{\text{CO}_2\text{-Produktion}}{\text{O}_2\text{-Aufnahme}} \quad (7.12)$$

In der Atmungsphysiologie ist kurzfristige Hyperventilation²⁹ ein den RQ bestimmender Faktor. Dabei wird vorübergehend vermehrt Kohlendioxid abgeatmet, so dass der RQ steigt. Ferner steht der RQ im Zusammenhang mit dem Energiehaushalt des Körpers.³⁰ Unterschiedliche Organe können durch Verbrennung verschiedener Nährstoffe (z.B. Fett oder Glukose) lokal zu differierenden RQ-Werten führen. Allerdings ist das obige Verhältnis beim Gasaustausch zwischen der Lunge und dem Blut ein globales Maß für den RQ im gesamten Körper, so dass für die Modellierung alle Gefäße zu einem einzigen zusammengefasst werden können. Die genaueren Zusammenhänge sprengen den Rahmen dieser Arbeit.

²⁸Der RQ wird aus der „Sicht“ des Blutes als CO₂-Produktion in den Gefäßen und eigene O₂-Abgabe an die Gefäße formuliert.

²⁹Dies ist eine „Steigerung der alveolaren Ventilation, die über die Stoffwechselbedürfnisse hinausgeht“, vgl. [ST90] S. 595.

³⁰Vgl. [ST90] Kapitel 24 Energiehaushalt S. 650 ff.

Kapitel 8

Das physiologische Modell

Im vorherigen Kapitel ist die Atmungsphysiologie und das Gesamtsystem EWS dargestellt worden. Das informale Systemmodell der Atmungsphysiologie wird hier in ein formales zur Simulation fähiges Systemmodell transferiert, wie es im EWS genutzt wird.

8.1 Der Zweck des physiologischen Modells

Am Anfang der Modellbildung steht eine Analysephase, die damit beginnt, einen Systemzweck festzulegen, wie es in Abschnitt 2.1.2 dargestellt worden ist. Zunächst müssen die Systemgrenzen gezogen werden, um anschließend entsprechend der Anforderungsdefinition im Software Engineering die Zielbestimmung zu konkretisieren.

Die Systemgrenze. Wie schon mehrfach erwähnt worden ist (vgl. z.B. Abschnitt 1.3), ist die im Zuge der Modellierung gezogene Systemgrenze nicht eindeutig und vom Standort des Betrachters abhängig. Da das physiologische Modell als Prognosemodell (vgl. Abschnitt 2.2.2) zur Entscheidungshilfe im Expertensystem EWS dienen soll und wenn möglich so aufgebaut sein soll, dass es noch in anderen Bereichen einsetzbar ist, sind gemäß der Darstellung in Abschnitt 7.1 (insb. Abbildung 7.1) folgende Komponenten zu beachten:

1. Der Mensch mit seiner Atmungsphysiologie.
2. Das Beatmungsgerät mit druckkontrollierter und druckunterstützter Beatmung.
3. Das Expertensystem mit dem zugehörigen Wissen, den Schätz- und Messwerten.
4. Der Arzt mit seinem Know-How und seiner Kontrollfunktion.

Für die Modellierung sind daraus verschiedene Konsequenzen zu ziehen:

1. Die Atmung ist druckkontrolliert und druckunterstützt zu modellieren.
2. Man kann bei der Modellbildung mindestens von den Messwerten der EVITA 4 als Eingangsgrößen ausgehen.
3. Da es das Ziel des Projektes ist, die Entwöhnung zu automatisieren, soll der Arzt nicht mehr Teil des zu betrachtenden Systems ist.
4. Wo immer möglich, ist eine „Whitebox“-Modellierung durchzuführen.

An dieser Stelle wäre, gemäß Punkt 2 der Aufzählung eine Liste aller Messwerte, die zur Verfügung stehen, anzugeben. Dies wird mit dem Hinweis darauf unterlassen, dass erstens das physiologische Modell dem Expertensystem untergeordnet ist und von dort seine Informationen erhält und zweitens der an beiden Teilprojekten beteiligte Experte die Datenlage kennt. Im Anhang C wird angegeben, welche Werte tatsächlich genutzt werden. Die anvisierte Autonomie des Gesamtsystems vom Arzt in Punkt 3 der Aufzählung bedeutet für die Modellierung, dass das Modell nur von der Kommunikation mit dem Expertensystem abhängig sein soll. Punkt 4 macht es erstens für den Anwender leichter, das Modellverhalten zu beurteilen, da dessen Komponenten eine reale Bedeutung tragen, und zweitens können dadurch leichter weitere Anwendungsfelder erschlossen werden.

Einsatzmodell. Das physiologische Modell soll zur Entscheidungshilfe im Rahmen des Evita-Weaning-Systems eingesetzt werden. Wie in Abschnitt 7.1 beschrieben worden ist, muss das physiologische Modell zusammen mit dem Expertensystem auf nur einem Rechner arbeiten. Daraus ergeben sich sowohl technische als auch inhaltliche Randbedingungen. Für die technische Realisierung heißt dies, Restriktionen bei der Laufzeit zu beachten, da das System in Echtzeit am Patienten arbeiten soll. Diese Einschränkungen sind vorab nicht zu spezifizieren gewesen; allerdings war die Laufzeit bei der Entwicklung des Systems stets im Auge zu behalten.

Inhaltlich ist das Modell so anzulegen, dass die Patienten, die vom EWS entwöhnt werden sollen, auch im Modell abgebildet werden können. An dieser Stelle müsste eine Darstellung der Typologisierung der Patienten erfolgen. Wiederum wird dies mit dem Hinweis unterlassen, dass der selbe Experte sowohl bei der Entwicklung des Modells als auch des Expertensystems beteiligt war und diese Einschränkung dadurch eingehalten werden konnte. Ferner soll das physiologische Modell im Rahmen des Gesamtsystems EWS dem Expertensystem untergeordnet werden. Dies folgt dem Grundgedanken, dass das Expertenwissen im Expertensystem gebündelt werden soll und sich das Wissen über die Atmungsphysiologie soweit wie möglich nur in der Modellstruktur findet.

Funktionsmodell. Gemäß der Darstellung in Abschnitt 2.1.2 sind Funktionsmodelle für das System und das Modell zu erstellen. Das Funktionsmodell des Systems ist bei der Festlegung der Systemgrenze besprochen worden: Das System Patient wird von einem druckkontrollierten bzw. druckunterstützten Beatmungssignal gesteuert und liefert als Ausgabe alle möglichen Mess- und Schätzwerte des EWS.

Für das Funktionsmodell des Modells sind im Sinne einer relationalen Betrachtung des Modells nach Definition 1.4 die unbedingt erforderlichen Ausgabegrößen, sowie die Eingangssignale des physiologischen Modells zu nennen : Da es zur Entscheidungshilfe bei der Entwöhnung dienen soll, ist es sinnvoll, das Modell so anzulegen, dass man es sich als einen zweiten beatmeten Patienten vorstellen kann. Daraus ergibt sich insbesondere, dass das Modellverhalten von einer den Beatmungsmodus BIPAP-ASB darstellenden Druckkurve abhängt.¹ Daher sind die beeinflussenden Umwelteinflüsse auf das System sehr einfach über das Drucksignal des Respirators zu modellieren.

In der relationalen Sichtweise auf das physiologische Modell sind die unbedingt nötigen Ausgabegrößen anzugeben:

1. Arterieller Partialdruck des Sauerstoffs P_aO_2 .
2. Arterieller Partialdruck des Kohlendioxids² P_aCO_2 .
3. Normventilation.

Wie in Abschnitt 7.5 beschrieben worden ist, wird die Normventilation als diejenige Ventilation definiert, die zu einem arteriellen P_aCO_2 von 40 mmHG führt. Daher ist sie entgegen den beiden anderen Größen nicht direkt über Simulationen zu bestimmen. Es ergibt sich als Realisierungsraum gemäß Definition 1.4:

$$\mathbb{R}_0^+ \times \left([P_aO_2^{(\text{Min})}, P_aO_2^{(\text{Max})}] \times [P_aCO_2^{(\text{Min})}, P_aCO_2^{(\text{Max})}] \right).$$

Hier wird von einem Eingangssignal mit einem nichtnegativen, im Prinzip beliebig hohem Druck ausgegangen. Die Grenzen für die extremalen Blutgaswerte werden entweder vom Experten vorgegeben und/oder ergeben sich aufgrund der vorgenommenen Implementati-on. Hierzu vergleiche man auch den Anhang E.2.

Ferner soll das physiologische Modell über Verfahren verfügen, die es in die Lage versetzen, sich an eine gegebene Patientensituation anzupassen. Dies wird in den Kalibrierungsaufgaben formuliert. Nach der Darstellung in Abschnitt 5.3 werden sinnvollerweise die beiden benötigten Ausgabegrößen P_aO_2 und P_aCO_2 als Kalibrierungszielwerte festgelegt (vgl. Vereinbarung 5.2), zumal sie von Zeit zu Zeit in einer Blutgasanalyse³ bestimmt werden können.

1. **Blutgas-Kalibrierungsaufgabe:** Wähle die Kalibrierungsparameter aus und stelle sie so ein, dass die simulierten und die am Patienten gemessenen Blutgase übereinstimmen. Darüber hinaus sollen die prognostizierten Blutgaswerte ebenfalls sinnvoll sein. Dabei soll der P_aO_2 auf 5 % und der P_aCO_2 auf 3 % genau vorhergesagt werden.⁴

¹Vgl. Kapitel 7.1 und 7.1.2.

²Vgl. Kapitel 7.3.2

³Vgl. Abschnitt 7.1.1.

⁴Im Zuge der Entwicklung sind die Grenzen in einem Spezialfall noch modifiziert worden.

2. **Volumen-Kalibrierungsaufgabe:** Um Blutgase kalibrieren zu können, müssen die gemessenen und simulierten Atemminuten-Volumina⁵ übereinstimmen. Daraus ergibt sich, dass eine weitere Kalibrierungsaufgabe zu lösen ist: Die gemessenen und simulierten Atemminuten-Volumina sollen auf 5 % genau übereinstimmen.

Datenstruktur-Modell. Bei der Atmung des Menschen handelt es sich um ein System mit kontinuierlicher Zeitskala. Dabei fließt, wie in Kapitel 7.2 dargestellt worden ist, Luft aufgrund von Druckgradienten durch die Luftröhre in die Lunge. Hier findet der Gasaustausch von Sauerstoff und Kohlendioxid statt. Dabei wird das Kapillarblut, das mit der Alveolarluft in Kontakt tritt, mit Sauerstoff versorgt und Kohlendioxid abgegeben. Ferner konnte im Abschnitt 7.2.1 die Atmung über Differenzialgleichungen beschrieben werden.

Dies legt nahe, als Zustandsgrößen Luftröhre und Kompartimente in der Lunge einzuführen. Sie werden als Level modelliert. Zwischen den Level herrscht ein Luftfluss, der in Raten modelliert wird. Damit ist man in der Lage, den sich ständig wandelnden Prozess abzubilden. Ferner stellen Level-Raten-Modelle eine zur Simulation geeignete Diskretisierung kontinuierlicher Differenzialgleichungen dar (vgl. Abschnitt 3.3.3). Schließlich besitzen Level-Raten-Modelle mit den zugehörigen Diagrammen ein geeignetes Darstellungsmodell, das die Kommunikation und den Wissenstransfer zwischen dem medizinischen Experten und dem Modellingenieur erleichtert, wie es in Abschnitt 5.1.3 besprochen worden ist. Kurzum: L/R-Modelle verbinden damit eine zeitorientierten Simulationstyp mit einem strukturorientierten Modellierungsansatz und sind zur Modellierung der Atmungsphysiologie geeignet.

8.2 Die Zeitskalen im physiologischen Modell

Kennzahlen der Atmung wie etwa das Atemzugs- oder Atemminuten-Volumen werden atemzugs- oder minutenweise angegeben, so dass u.a. Atemzüge bzw. Minuten geeignete Beobachtungsskalen in diesem Zusammenhang darstellen. Darüber hinaus handelt es sich bei der Atmung um einen Prozess, der auch zu jedem Zeitpunkt interessante Phänomene wie etwa der maximale inspiratorische Volumenfluss zeigt. Insgesamt folgt, dass die Übertragung der kontinuierlichen Zeitskala im Systemmodell auf das Simulationsmodell feiner sein sollte als die Atemzugsebene, so dass die Größenordnung der Daten im „vernünftigen“ Bereich ist. Eine Simulation in Sekunden erscheint ausreichend.

Die gröbere Minuten-Skala ist noch aus einem weiteren Grund notwendig: Kennzahlen wie der P_{aO_2} und das Atemzugvolumen hängen auf Atemzugsebene vom Inspirationsdruck ab. Er wird dadurch bestimmt, ob es sich um einen maschinellen oder spontanen Atemzug handelt. Diese Schwankungen werden durch geeignete Aggregation auf Minutenebene wie Summation und Mittelwertbildung über die Atemzüge eliminiert.

⁵Vgl. Abschnitt 7.2.2.

8.3 Modellierung des Drucksignals

Aufgrund der Darstellung des Modellzwecks 8.1 in 7.1.2 und über die Beatmungsformen muss das Modell druckkontrolliert bzw. druckunterstützt durch den Beatmungsmodus BIPAP-ASB gesteuert werden. Für die Codierung dieses Beatmungsmodus sind drei Arten von Atemzügen zu unterscheiden:

1. **Rein maschinelle Atemzüge:** Atemzüge mit einem Inspirationsdruck auf P_{Insp} -Niveau ohne Unterstützung des Patienten.
2. **ASB-Atemzüge:** Atemzüge, die vom Patienten durch Eigenaktivität eingeleitet werden und mit einem Druck auf P_{ASB} -Niveau unterstützt werden.
3. **Mandatorische Atemzüge:** Atemzüge, die der Patient „kurz“ vor einem maschinellen Atemzug einleitet und deshalb auf P_{Insp} -Niveau „vorgezogen“ werden.

Mandatorische unterscheiden sich von rein maschinellen Atemzügen dadurch, dass sie vom Patienten eingeleitet werden und deshalb vermutlich ein höheres Maß an Unterstützung durch den Patienten haben als maschinelle. Dass sie nicht genau in das durch die Beatmungsparameter vorgesehene Zeitraster passen, kann vernachlässigt werden. Daher können für die steuernde Druckkurve mandatorische und maschinelle Atemzüge zusammengefasst werden. Sie ergeben die MASCHINELLE FREQUENZ als Differenz der Frequenzen der gemessenen Atemzügen und der ASB-Atemzüge. Diese Modellierung setzt allerdings voraus, dass die noch zu beschreibende Spontanaktivität (vgl. Kapitel 8.6) nicht zwischen den mandatorischen und den maschinellen Atemzügen unterscheidet, sondern beide Typen gleich unterstützt werden. Die Frequenz der ASB-Atemzüge ist ebenfalls eine Messgröße am Beatmungsgerät, die SPONTANE FREQUENZ.

Das physiologische Modell aggregiert seine Daten auf Minutenebene. Ebenso erhält es Informationen über die spontane Atemzugsfrequenz nicht umgehend vom Patienten, sondern als Information über die letzte Minute. So können die ASB-Atemzüge nicht zu jenem Zeitpunkten simuliert werden, an denen sie durchgeführt wurden. Daher sind die ASB-Atemzüge gleichmäßig unter den maschinellen Atemzügen verteilt worden. Dieses erfolgt dadurch, dass die Atemzüge pro Minute nummeriert werden und jeder $\lfloor f_{\text{Gesamt}}/f_{\text{Spont}} \rfloor$ -te Atemzug⁶ ein ASB-Atemzug sein soll. Ist die Anzahl der maschinellen Atemzüge kleiner als die der spontanen Atemzüge, werden analog die maschinellen Atemzüge verteilt.

Schematisch zeigt Abbildung 8.1, von welchen Größen das Drucksignal des physiologischen Modells gesteuert wird. Die Dauer eines Atemzugs ergibt sich aus der Gesamtfrequenz. Zusätzlich wird noch die Rampenzeit (vgl. Abschnitt 7.1.2). angegeben. Schließlich sei darauf aufmerksam gemacht, dass der P_{ASB} als Druck „über 0 mbar“ angegeben wird.⁷

Im Zuge einer formalen Systembeschreibung bildet das beschriebene Drucksignal die externe Steuerung aus der Systemumwelt. Durch t-Konkatenation aus Definition 1.3 wird das Signal einer Minute auf mehrere Minuten ausgedehnt.

⁶Dabei bezeichnet $\lfloor \]$ die Gaußklammer, die größte ganze Zahl, die kleiner gleich dem Argument ist.

⁷Am Beatmungsgerät wird er als Druck „über PEEP“ eingestellt.

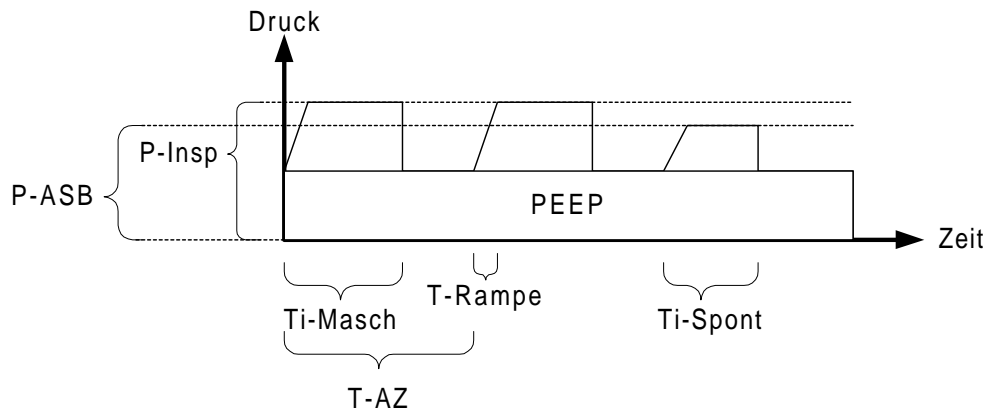


Abbildung 8.1: Eine im physiologischen Modell verwendete Druckkurve

8.4 Modellierung der Ventilation

Zunächst wird das grundsätzliche Vorgehen bei der Modellierung der Atmung in Level und Raten anhand eines Modells mit einem Kompartiment genauer beschrieben. Davon ausgehend wird das medizinische Dreikompartimente-Modell in ein L/R-Modell übertragen. Dabei wird zunächst der Spezialfall betrachtet, dass die Atmung ohne Eigenaktivität stattfindet und daher P_{Aug} aus Gleichung (7.5) gleich Null ist. Anschließend wird dies in die Sprache der Elektrotechnik übersetzt. Aus dieser Perspektive wird ein Ansatz zur Integration der Eigenaktivität in das Modell gefunden, die bei der konkreten Modellierung der Spontanaktivität in die Sprache der L/R-Modelle zurück transferiert wird.

8.4.1 Das grundsätzliche Vorgehen

Anhand eines einfachen Einkompartiment-Modells, wie es in Abschnitt 7.2.1 eingeführt worden ist, wird gezeigt, wie Level-Raten-Modelle konkret zur Modellierung der Atmungsphysiologie eingesetzt werden. Die Luft wird in einem Level *Kompartiment* gespeichert und von einer Rate Fl_{Luft} geregelt. Die atmungsmechanischen Größen *Compliance* und *Resistance* werden im Sinne eines allgemeinen Level-Raten-Modells aus Definition 3.12 als unabhängige Parameter realisiert, so dass eine druckgesteuerte Modellierung der Atmung möglich ist. Abbildung 8.2 zeigt das L/R-Diagramm eines Einkompartiment-Modells. Dabei bezeichnet *P-Ext* das Drucksignal vom Respirator oder aus der Umwelt. Zusammen mit dem abhängigen Parameter *P-Kompartiment*, dem Druck im Kompartiment, der sich aus dem Volumen im Kompartiment und dessen Compliance ergibt, erzeugt es einen Druckgradienten. Er führt mit der Resistance zu einem Volumenfluss, der im Kompartiment endet oder dort beginnt. Diese Zusammenhänge sind die Übertragung der Prozessbeschreibung der Grundlagen der Atmung aus Abschnitt 7.2.1 in die Terminologie von L/R-Modellen. Die genauen Zusammenhänge werden in folgenden Modellgleichungen angegeben:

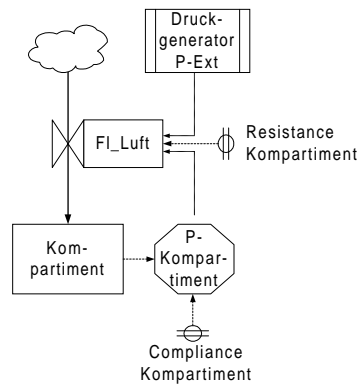


Abbildung 8.2: Das Einkompartiment-Modell als Level-Raten-Modell

$$C_k = \text{const}, \quad (8.1)$$

$$R_k = \text{const}, \quad (8.2)$$

$$\text{Vol}_k(0) = \text{const}, \quad (8.3)$$

$$P_{\text{Ext}}(t) = f(t), \quad (8.4)$$

$$P_k(t) = \frac{\text{Vol}_k(t)}{C_k}, \quad (8.5)$$

$$\text{Fl_Luft}(t, t + \Delta t) = \frac{1}{R_k} \left(P_{\text{Ext}}(t) - P_k(t) \right), \quad (8.6)$$

$$\text{Vol}_k(t + \Delta t) = \text{Vol}_k(t) + \Delta t * \text{Fl_Luft}(t, t + \Delta t). \quad (8.7)$$

Das Drucksignal $f(t)$ wird so modelliert, wie es im Abschnitt 8.3 beschrieben worden ist. Die Ratengleichung (8.6) entspricht der Gleichung (7.5) für die Volumenänderung \dot{V} im Kapitel 7.2.1, wobei allerdings hier die Eigenaktivität durch P_{Aug} noch unberücksichtigt geblieben ist.

Anhand dieses Einkompartiment-Modells können einige grundsätzliche Bemerkungen zur Dynamik der Atmung gemacht werden. Ist der Druck in der Umwelt P_{Ext} größer als im Kompartiment, so ist der Druckgradient und somit der Fluss in die Lunge Fl_Luft positiv: Luft fließt ins Kompartiment. Ist umgekehrt dort der Druck größer als in der Umwelt, so fließt Luft in diese zurück. Somit kann man In- und Expiration durch das Vorzeichen des Flusses voneinander unterscheiden.

8.4.2 Das Grundmodell der Ventilation

Das medizinische Dreikompartimente-Modell aus Abschnitt 7.5 wird hier in die Terminologie von L/R-Modellen übertragen. Es stellt sich als ein Modell mit drei Level `NORMAL`, `TOTRAUM` und `SHUNT` für die Kompartimente der Lunge dar. Die Kompartimente sind anhand des Ventilations-Perfusions-Verhältnisses unterteilt worden. Shunt und Totraum stehen am jeweiligen Ende einer kontinuierlichen Skala mit einem VPV von 0 bzw. ∞ . Im Normal-Kompartiment werden alle Lungenareale zusammengefasst, die ein normales VPV aufweisen. Die Kompartimente erhalten ihre Luft aus einem Level `TRACHEA`, das

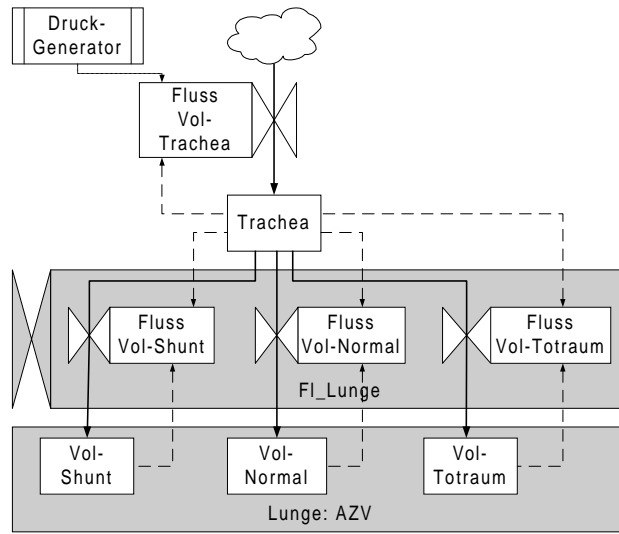


Abbildung 8.3: Das Volumenmodell ohne die Druckberechnungen

seinerseits aus der Umwelt belüftet wird, wobei der Druckgradient über einen Druckgenerator am Beatmungsgerät erzeugt wird. Die Luftflüsse werden dabei in völliger Analogie zu obigem Einkompartiment-Modell erzeugt: Jedes Level erhält eine Compliance und eine Resistance, so dass der Fluss zwischen Trachea und den Kompartimenten über Druckdifferenzen geregelt wird. Eine abschließende Modellierung der Level ist an dieser Stelle nicht möglich, da noch der Diffusionsprozess betrachtet werden muss.

$$R_{\text{Trachea}} = \text{const}, \quad (8.8)$$

$$C_k = \text{const}, \quad (8.9)$$

$$R_k = \text{const}, \quad (8.10)$$

$$\text{Vol}_{\text{Trachea}}(0) = \text{const}, \quad (8.11)$$

$$\text{Vol}_k(0) = \text{const}, \quad (8.12)$$

$$P_{\text{Ext}}(t) = f(t), \quad (8.13)$$

$$P_{\text{Trachea}}(t) = \frac{\text{Vol}_{\text{Trachea}}(t)}{C_{\text{Trachea}}}, \quad (8.14)$$

$$P_k(t) = \frac{\text{Vol}_k(t)}{C_k} \quad (8.15)$$

$$\text{FL}_{\text{Trachea}}^{\text{Vol}}(t, t + \Delta t) = \frac{1}{R_{\text{Trachea}}} \left(P_{\text{Ext}}(t) - P_{\text{Trachea}}(t) \right), \quad (8.16)$$

$$\text{FL}_k^{\text{Vol}}(t, t + \Delta t) = \frac{1}{R_k} \left(P_{\text{Trachea}}(t) - P_k(t) \right). \quad (8.17)$$

Dabei hat die Gleichung (8.15) nur vorläufigen Charakter, da sie nur den elastischen Druck im Kompartiment darstellt und noch die Modellierung der Spontanaktivität hinzuzufügen ist. Der Shunt wird als nicht ventiliertes Kompartiment gesondert behandelt: Anstelle von Gleichung (8.17) wird angenommen, dass überhaupt kein Fluss stattfindet, die Rate also konstant Null ist.

Abbildung 8.3 zeigt den gesamten Aufbau des Modells bis hierhin. Es sind allerdings der Einfachheit halber, die weiteren Parameter nicht mit abgebildet worden. Deren Bedeutung ist in den lang gestrichelten Pfeilen angedeutet. Die drei Kompartimente bilden zusammen die gesamte Lunge. Sie ist in Abbildung 8.3 als ein abstraktes Level grau unterlegt symbolisiert.⁸ Man kann also die relative Größe PZ_k der Kompartimente zur gesamten Lunge angeben. Die jeweilige relative Größe entspricht der im Abschnitt 7.5 eingeführten Shunt- bzw. Totraumfraktion und für das Normal-Kompartiment einer NORMAL-FRAKTION. Darüber hinaus bestimmen die Anteile, wie viel des Atemvolumens am Gasaustausch mit dem Blut teilnehmen kann. Nach dem Dreikompartimente-Modell gehört zu einem Kompartiment auch dessen Versorgung mit Blut. Diese wird ebenfalls über die relative Größe der Kompartimente geregelt (vgl. Abschnitt 8.7). Damit haben die relativen Größen PZ_k der Kompartimente erheblichen Einfluss sowohl auf die Ventilation des gesamten Atemvolumens als auch auf die arteriellen Blutgase des Modells. Sie werden als Kalibrierungsparameter eine zentrale Rolle zur Lösung der Kalibrierungsaufgaben aus Abschnitt 8.1 spielen.

8.4.3 Das Dreikompartimente-Modell in elektrotechnischer Formulierung

Für das Lösen der Volumen-Kalibrierungsaufgabe sowie die Integration der zu modellierenden Spontanaktivität ist es günstig, das entwickelte Volumenmodell der Ventilation elektrotechnisch zu formulieren. Hierzu werden Stromkreise aus einer Spannungsquelle, Widerständen und Kondensatoren betrachtet. Letztere dienen dazu, kurzzeitig elektrische Ladung zu speichern. Der Fluss des Stromes wird durch Widerstände behindert. Für Kondensatoren und Widerstände gelten im Stromkreis bei Gleichstrom Regeln, die im Anhang D dargestellt werden und unter anderem zur Kalibrierung der Atmungsmechanik benutzt werden. Diese elektrotechnische Sichtweise eröffnet einen anderen Blickwinkel auf die Atmungsphysiologie. Die Parallelen zwischen beiden Systemmodellen werden in der folgenden Tabelle aufgelistet, falls in der Atmungsphysiologie das Luftvolumen betrachtet wird:

| Atmungsphysiologie | Elektrotechnik |
|--------------------|----------------|
| Kompartiment | Kondensator |
| Compliance | Kapazität |
| Volumen | Ladung |
| Volumenfluss | Stromfluss |
| Druck | Spannung |
| Resistance | Widerstand |
| Umgebungsluft | Erde |

Aufgrund der gezogenen Parallelen stellt sich das Grundmodell wie in Abbildung 8.4 dar. Kondensatoren werden als zwei Parallelen, Widerstände als Rechtecke und die Erde als drei kleiner werdende Parallelen symbolisiert. Die Spannungserzeuger entsprechen zwei Kreisen. Entgegen der bisherigen Darstellung, die den Thorax vernachlässigt hat, wird

⁸Auf der Ebene des Darstellungsmodells im Graphen kann man hier das Modell um weitere Kompartimente erweitern, indem man die Regel zum Einbetten von Netzen 3.2 verwendet.

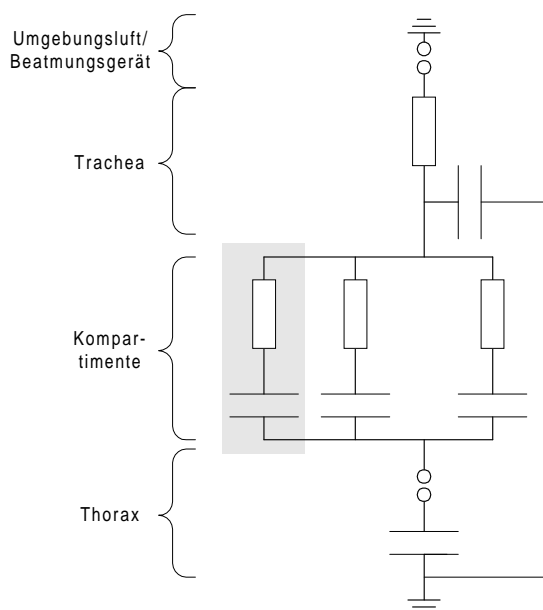


Abbildung 8.4: Das Dreikompartimente-Modell in Kondensatoren-Darstellung

hier ein Thorax-Kondensator mit berücksichtigt. Er ist in Reihe mit den drei parallelen Kompartiment-Kondensatoren geschaltet. Zwischen dem Thorax-Kondensator und den Kompartimenten findet sich ein Spannungserzeuger, der die Eigenaktivität darstellt. Das bisherige Grundmodell ist hier als Spezialfall enthalten, indem die Thorax-Compliance als unendlich groß angenommen wird und so das Lungenvolumen nicht durch den Thorax begrenzt wird. Ferner ist dort der zusätzliche Spannungserzeuger nicht aktiv. Wie sich dieser Kondensator und die zusätzliche Spannungsquelle im L/R-Modell für die Modellierung der Spontanaktivität niederschlägt, wird im entsprechenden Abschnitt 8.6 dargestellt.

Der Fluss in die Kompartimente wird von den jeweiligen vorgeschalteten Widerständen geregelt. Für den Shunt ist er unendlich groß. Der Fluss durch die Trachea wird über einen Widerstand und einen mit den Kompartimenten parallel geschalteten Trachea-Kondensator modelliert. Schließlich soll darauf aufmerksam gemacht werden, dass das L/R-Diagramm in Bezug auf die Modellierung der Trachea in die Irre führt: Es suggeriert, dass die Trachea in Reihe mit den Kompartimenten geschaltet ist. Diese Vorstellung führt spätestens bei der Anwendung der im Anhang genannten Kapazitätsregeln bei der Lösung Volumen-Kalibrierungsaufgabe zu Fehlern (vgl. Abschnitt 8.9.2).

8.4.4 Das Atemzugvolumen

Das Volumen in der Lunge, das an der Atmung teilnimmt, ist die Summe der Volumina in den Kompartimenten. Daher kann man in Gleichung (8.18) die Variable $AZV(t)$, das ATEMZUGVOLUMEN, als die Summe der Volumina der drei Kompartimente bilden. Ferner wird zur Vereinfachung eine Variable für den Fluss in die Lunge $Fl_{Lunge}(t, t + \Delta t)$ nach Gleichung (8.17) als die Summe der Flüsse in die Kompartimente eingeführt, die im weiteren Verlauf benötigt wird. Sowohl $AZV(t)$ als auch $Fl_{Lunge}(t, t + \Delta t)$ sind in Abbildung 8.3 als grau unterlegte abstrakte Größen zu erkennen.

$$AZV(t) = \sum_{k \in K} Vol_k(t), \quad (8.18)$$

$$Fl_{Lunge}(t, t + \Delta t) = \sum_{k \in K} FL_k^{Vol}(t, t + \Delta t). \quad (8.19)$$

Dabei wird jeweils über die Menge K der im Modell vorhandenen Kompartimente addiert. Der Name Atemzugvolumen wird dadurch gerechtfertigt, dass die Differenz aus dieser Größe zum Ende der Inspiration und zum Ende der Expiration eben das Atemzugvolumen ergibt, wie es schematisch in Abbildung 7.4 zu sehen ist.

8.5 Modellierung von Sauerstoff- und Kohlendioxidaustausches

Äußere Atmung besteht darin, Sauerstoff in die Lunge zu transportieren und produziertes Kohlendioxid an die Umwelt abzugeben. In der bisherigen Modellierung ist allerdings lediglich das Gesamtvolumen betrachtet worden, das zwischen Umwelt und Lunge ausgetauscht wird. Welche Anteile davon Sauerstoff bzw. Kohlendioxid sind, ist unberücksichtigt geblieben, aber für die Modellierung der Diffusion zwischen Lunge und Blut von Bedeutung. Für das grundsätzliche Problem, wie diese Anteile abzubilden sind, liefert das Beatmungsgerät einen Ansatz.

8.5.1 Das grundsätzliche Vorgehen

Am Respirator stellt man die inspiratorische Sauerstofffraktion $F_{I}O_2$ (vgl. Abschnitt 7.1.2) ein. Da dieser Wert damit für das physiologischen Modell als Parameter zur Verfügung steht und aus dem Volumenteil des Modells zu jedem Zeitpunkt feststeht, wie viel Luft gerade zwischen dem Patienten und der Umwelt ausgetauscht wird, kann — zunächst für die Inspiration — angegeben werden, wie viel Sauerstoff vom Patienten aufgenommen wird. Es bietet sich also an, parallel zum Modell für das Volumen zwei strukturell identische Modellteile für den Sauerstoff und das Kohlendioxid zu erzeugen, die jeweils die MENGE an Sauerstoff bzw. Kohlendioxid in der Trachea und den Kompartimenten angeben.

Zu Beginn der Inspiration ist dieses Vorgehen beim Gasaustausch zwischen Patient und Umwelt unproblematisch: Der Sauerstofffluss bestimmt sich aus dem Produkt von Volumenfluss und inspiratorischer Sauerstofffraktion $F_{I}O_2$; mit einer inspiratorischen Fraktion des Kohlendioxids von Null wird entsprechend kein Kohlendioxid eingeatmet. Ferner erlaubt dieses Vorgehen auch das Bilden von Fraktionen: Der Quotient aus dem Inhalt des Sauerstoff- bzw. Kohlendioxids-Levels und dem entsprechenden Volumen liefert die Fraktion. Jedoch ist der Übergang von den oberen leitenden Atemwegen zur Lunge noch nicht zu modellieren: Zu Beginn der Inspiration wird noch verbrauchte Luft eingeatmet, die in den leitenden Atemwegen vom vorherigen Atemzug vorhanden ist. Erst in deren Verlauf folgt frische Luft, wie es in Abschnitt 7.2.4 dargestellt worden ist. Gleiches gilt entsprechend für die Expiration. Praktisch muss daher die Trachea als Rohr gemäß der Darstellung in Abschnitt 7.2.4 modelliert werden.

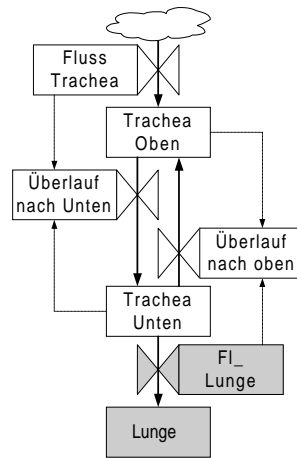


Abbildung 8.5: Die Modellierung der Trachea mit Hilfe zweier Level

8.5.2 Modellierung der Trachea

Aufgrund des in Abschnitt 8.4 eingeführten Volumen-Grundmodells ist das gesamte aus der Umwelt eingeatmete Volumen gemäß Gleichung (8.16) bekannt. Dieser Fluss ist so in den leitenden Atemwegen darzustellen, dass die Expirogramme aus Abbildung 7.6 reproduziert werden können. Hierzu erfolgt eine Modellierung als Trachea. Deren wesentlicher Punkt ist, dass am oberen Ende der Trachea andere Fraktionen vorhanden sind als am unteren Ende. Dies hat dazu geführt, dass die Trachea durch zwei Level in einen oberen und einen unteren Abschnitt unterteilt wird. Entgegen dem bisherigen Paradigma, die Flüsse druckkontrolliert über Compliance und Resistance zu realisieren, machen wir uns hier zunutze, dass das Beatmungsgerät einen Messwert des seriellen Totraums liefert. Im Modell könnte dieser die gesamte Größe der Trachea angeben. Daher werden die Flüsse zwischen den beiden Level volumenkontrolliert modelliert. Für die Größe der Flüsse ist der Leitgedanke, dass die zwei Luftschichten sich nur in einem kleinen Bereich mischen und nacheinander ab- bzw. eingeatmet werden.

Abbildung 8.5 zeigt das Schema: In die obere Trachea fließt die Rate „Fluss Trachea“, die schon aus dem Volumenteil des Modells bekannt ist. Aus dem „Trachea-Modell“ fließen alle Abflüsse gemäß Gleichung (8.19) in die Lunge ab: Die Flüsse aus der Trachea in die Kompartimente sind zu einer Rate FL_{Lunge} zusammengefasst. Zwei mit Überlauf bezeichnete Raten steuern den Transfer von Luft zwischen den beiden beteiligten Level gemäß den folgenden Gleichungen:

$$FL_{\ddot{U}U}^{Vol}(t, t + \Delta t) = \begin{cases} \max\left(0, FL_{TR}^{Vol}(t, t + \Delta t)\right), & \text{falls } (*) \\ 0, & \text{sonst,} \end{cases}$$

wobei $(*)$ bedeutet: $TR_{Unten}^{Vol}(t) \leq (1 - \text{Trennschicht}) * Vd_s$,

$$\text{FL}_{\ddot{U}O}^{\text{Vol}}(t, t + \Delta t) = \begin{cases} \max\left(0, -\text{Fl}_{\text{Lunge}}(t, t + \Delta t)\right), & \text{falls (**)} \\ 0 & \text{sonst,} \end{cases}$$

wobei (**) bedeutet: $\text{TR}_{\text{Oben}}^{\text{Vol}}(t) \leq (1 - \text{Trennschicht}) * \text{Vd}_s$.

Der Anteil der Luft in der Trachea, der sich mischt, wird über die Größe $(1 - \text{Trennschicht})$ modelliert, mit Vd_s wird der serielle Totraum bezeichnet. Die Ratengleichungen sind folgendermaßen zu interpretieren: Solange im unteren Level $\text{TR}_{\text{Unten}}^{\text{Vol}}(t)$ noch genügend Luft vorhanden ist, fließt bei der Inspiration ($\text{FL}_{\text{TR}}^{\text{Vol}} > 0$ nach Abschnitt 8.4.1) die eingeatmete Luft ins obere Level. Erst wenn das untere fast leer ist, d.h. wenn die Bedingung $\text{TR}_{\text{Unten}}^{\text{Vol}}(t) \leq (1 - \text{Trennschicht}) * \text{Vd}_s$ erfüllt ist, wird die inspirierte Luft direkt — also ohne zunächst ins obere zu gehen — ins untere Level umgeleitet. Technisch wird dieses Umleiten dadurch realisiert, dass in einem Zeitschritt die inspirierte Luft sowohl ins obere Level addiert wird als auch dort subtrahiert und wieder zum unteren addiert wird. Dies ist über den „Überlauf nach Unten“ $\text{FL}_{\ddot{U}U}^{\text{Vol}}$ modelliert. Bei der Expiration arbeitet ein analoger Mechanismus für die Rate $\text{FL}_{\ddot{U}O}^{\text{Vol}}$, die vom Inhalt des oberen Trachea-Levels $\text{TR}_{\text{Oben}}^{\text{Vol}}(t)$ abhängig ist.

Wendet man nun gemäß dem zuvor geschilderten grundsätzlichen Vorgehen dies auch auf den Sauerstoff und das Kohlendioxid an, können die jeweiligen oberen und unteren Fraktionen gebildet werden. Damit ergibt sich auch der Anteil des expirierten Kohlendioxids, was ein Messwert am Beatmungsgerät ist und in Expirogrammen dargestellt wird. Die endtidale CO_2 -Fraktion berechnet sich im Modell aus der oberen Trachea als:⁹

$$\text{Fr}_{\text{TR}_{\text{Oben}}}^{\text{CO}_2}(t) = \frac{\text{TR}_{\text{Oben}}^{\text{CO}_2}(t)}{\text{TR}_{\text{Oben}}^{\text{Vol}}(t)} \quad (8.20)$$

Ebenso werden die Fraktionen von Kohlendioxid und Sauerstoff in den übrigen „Trachea-Level“ gebildet, so dass alle Flüsse zwischen den Trachea-Level sowie die inspiratorischen Flüsse in die Sauerstoff- und Kohlendioxidkompartimente bestimmt werden können. Um schließlich auch noch die Flüsse aus der Lunge in die Trachea modellieren zu können, müssen die Fraktionen in der Lunge bekannt sein.

8.5.3 Die Fraktionen in der Lunge

In Kapitel 7.2.2 ist dargelegt worden, dass stets ein Restvolumen, die sog. funktionelle Residualkapazität FRC, in der Lunge verbleibt, und sich die hinzugekommene Luft mit dieser vermischt, so dass die Anteile an Sauerstoff und Kohlendioxid nicht übermäßig schwanken. Daher ist es notwendig, auch die FRC für die Bildung der Fraktionen in den Kompartimenten mit zu berücksichtigen. Hierzu benötigt man zunächst eine Näherung der FRC. Da weitere Informationen über die FRC im Kontext des physiologischen Modells fehlen, wird für deren Schätzung ein physiologischer Standardwert von 30 ml pro

⁹Im Folgenden werden stets Fraktionen mit Fr bezeichnen. Tiefgestellt wird das jeweilige Kompartiment angegeben, während der obere Index die Verbindung anzeigt.

kg Körpergewicht benutzt. Dieser Schätzwert bezieht sich auf die gesamte Lunge. Da die Lunge als die Summe der darin enthaltenen Kompartimente betrachtet wird, muss die FRC auf die Kompartimente verteilt werden. Hierzu wird die Variable $\text{Vol}_k^{\text{Lunge}}$, das LUNGENVOLUMEN DES KOMPARTIMENTES k als die Summe aus dem geatmeten Volumen und dem jeweiligen Anteil an der FRC eingeführt. Nur der Shunt S erhält kein zusätzliches Residualvolumen, da er nicht belüftet wird. Insgesamt ergeben sich folgende Gleichungen für das Lungenvolumen im Normal-Kompartiment N , dem Totraum T und dem Shunt S :

$$\text{Vol}_N^{\text{Lunge}}(t) = \text{Vol}_N(t) + \text{PZ}_N * \text{FRC}, \quad (8.21)$$

$$\text{Vol}_T^{\text{Lunge}}(t) = \text{Vol}_T(t) + \text{PZ}_T * \text{FRC}, \quad (8.22)$$

$$\text{Vol}_S^{\text{Lunge}}(t) = \text{Vol}_S(t). \quad (8.23)$$

Es ergeben sich analog Gleichung (8.20) die folgenden Fraktions-Gleichungen:

$$\text{Fr}_k^{\text{Ele}}(t) = \frac{k^{\text{Ele}}(t)}{\text{Vol}_k^{\text{Lunge}}(t)} \quad \text{Ele} \in \{\text{O}_2, \text{CO}_2\}, k \in \{N, T, S\} \quad (8.24)$$

Somit sind die notwendigen Werte vorhanden, um die Flüsse der Ventilation abbilden zu können.

8.5.4 Zusammenfassung: Die Sauerstoff- und Kohlendioxidflüsse

Wir fassen diesen Abschnitt dadurch zusammen, dass der Fluss des Sauerstoffs aus der Umwelt über die Trachea zu den Kompartimenten in beiden Atemphasen verfolgt wird: Bei der Inspiration bestimmt sich die Menge des zwischen Umwelt und Patienten ausgetauschten Sauerstoffs aus dem Austausch zwischen Umwelt und Trachea (vgl. Gleichung 8.16) und der inspiratorischen Sauerstofffraktion.¹⁰ Bei der Expiration wird stattdessen die Fraktion in der oberen Trachea verwendet. Da In- und Expiration über die Vorzeichen der Flüsse unterschieden werden können, ergibt sich folgender Zusammenhang:

$$\text{FL}_{\text{Trachea}}^{\text{O}_2}(t, t + \Delta t) = \begin{cases} \text{FL}_{\text{Trachea}}^{\text{Vol}}(t, t + \Delta t) * \text{F}_{\text{I}}\text{O}_2 & , \text{FL}_{\text{Trachea}}^{\text{Vol}} \geq 0, \\ \text{FL}_{\text{Trachea}}^{\text{Vol}}(t, t + \Delta t) * \text{Fr}_{\text{TR}_{\text{Oben}}}^{\text{O}_2}(t) & , \text{FL}_{\text{Trachea}}^{\text{Vol}} < 0. \end{cases} \quad (8.25)$$

Innerhalb der Trachea ergeben sich die Überläufe aus den Volumenüberläufen (Gleichung (8.20)) und den entsprechenden Fraktionen:

$$\text{FL}_{\dot{U}U}^{\text{O}_2}(t, t + \Delta t) = \text{FL}_{\dot{U}U}^{\text{Vol}}(t, t + \Delta t) * \text{Fr}_{\text{TR}_{\text{Oben}}}^{\text{O}_2}(t), \quad (8.26)$$

$$\text{FL}_{\dot{U}O}^{\text{O}_2}(t, t + \Delta t) = \text{FL}_{\dot{U}O}^{\text{Vol}}(t, t + \Delta t) * \text{Fr}_{\text{TR}_{\text{Unten}}}^{\text{O}_2}(t). \quad (8.27)$$

¹⁰Eigentlich sind die Einheiten von Sauerstoff und Kohlendioxid in Stoffmengen anzugeben. Da aber das Modell bei Normbedingungen (vgl. Kapitel 7.2.3) betrachtet wird, kann die Stoffmenge mit dem Volumen, das sie einnimmt, identifiziert werden.

Schließlich erfolgt die Steuerung der Sauerstoffflüsse aus der unteren Trachea in die Kompartimente (und zurück) wieder über das Vorzeichen der Volumenflüsse:

$$\text{FL}_k^{\text{O}_2}(t, t + \Delta t) = \begin{cases} \text{FL}_k^{\text{Vol}}(t, t + \Delta t) * \text{Fr}_{\text{TR}_{\text{Unten}}}^{\text{O}_2}(t), & \text{FL}_k^{\text{Vol}} \geq 0, \\ \text{FL}_k^{\text{Vol}}(t, t + \Delta t) * \text{Fr}_k^{\text{O}_2}(t) & , \text{FL}_k^{\text{Vol}} < 0. \end{cases} \quad (8.28)$$

Dabei ist wiederum zu beachten, dass der Fluss in den Shunt konstant Null ist und die Kompartimente mit k indiziert werden. Ferner gelten diese Gleichungen völlig analog für das Kohlendioxid.

8.6 Modellierung der Spontanaktivität

Zur Modellierung der Spontanaktivität werden zunächst die wichtigsten Punkte aus Kapitel 7.2.1 rekapituliert und daraus ein Ansatzpunkt im Dreikompartimente-Modell der Ventilation entwickelt, mit dem die Spontanaktivität modelliert werden kann. Diese wird im folgenden Abschnitt vorgestellt. Schließlich wird die Modellierung der Spontanaktivität wieder in ein L/R-Modell zurück übertragen.

8.6.1 Das grundsätzliche Vorgehen

Eigenatmung entsteht dadurch, dass das Volumen in der Lunge vergrößert wird. Durch Anheben der Rippenbögen (Brustatmung) oder Absenken des Zwerchfelles (Bauchatmung) vergrößert sich der Raum in der Lunge. Diese folgt der Volumenvergrößerung, weil sie über eine dünne Flüssigkeitsschicht an den Thorax gebunden ist. Durch diese Volumenänderung folgt — vermittelt über den intrapleurale Druck - eine Druckverringering in der Lunge, dem intrapulmonalen Druck. So entsteht ein Druckgradient zwischen der Umwelt und der Lunge; es kommt zur Inspiration.

Dies ist die Stelle, an der neben dem elastischen Druck in Gleichung (7.5) $P_{\text{El}} = V/C$, der durch die Compliance entsteht, auch der Druck durch die Eigenaktivität P_{Aug} in das Volumenmodell integriert werden kann. Das Modell ist dahingehend zu erweitern, dass ein Druckgenerator PG_{Aug} zu entwerfen ist, der zusätzlich zum elastischen Druck wirkt und dadurch die Gesamtdruckdifferenz vergrößert.

Da es nicht Ziel des physiologischen Modells ist, die Größe der Eigenaktivität zu bestimmen, sondern vielmehr notwendig ist, das gemessene Atemminuten-Volumen zu reproduzieren, um so erst „sinnvolle“ Blutgaswerte möglich zu machen, wird die Eigenaktivität als AUGMENTIERUNG und nicht mit dem medizinischen Fachbegriff SPONTANAKTIVITÄT bezeichnet. Ohne auf die konkrete Modellierung der Augmentierung einzugehen, stellt sich der bisher gefundene Ansatz für den schließlich wirkenden Druck im Druckgradienten entsprechend Gleichung (7.5) folgendermaßen dar:

$$P_k(t) = \frac{\text{Vol}_k(t)}{C_k} + \text{PG}_{\text{Aug}}(t) \quad (8.29)$$

Ziel der folgenden Darstellung ist es, diesen zusätzlichen Druckgenerator $\text{PG}_{\text{Aug}}(t)$ geeignet zu modellieren.

8.6.2 Die Durchführung

Der zusätzliche Druckgenerator $PG_{\text{Aug}}(t)$, der den AUGMENTIERUNGSDRUCK liefert, wird zeitabhängig modelliert. Er soll während der Inspiration für zusätzlichen Inspirationsdruck sorgen, während er die Expiration forcieren soll. Dies motiviert folgende Vorgehensweise: Es wird eine zeitabhängige Volumenkurve $\text{Vol}_{\text{Aug}}(t)$ vorgegeben. Sie stellt den Raum dar, den der Thorax umschließt. Zu jedem Zeitpunkt ist mit dem Atemzugvolumen $AZV(t)$ aus Abschnitt 8.4.4 bekannt, wie viel Luft in der Lunge ist. Daraus ergibt sich ein noch zu augmentierendes Volumen $\text{Vol}_{\text{Aug}}(t) - AZV(t)$, im Falle der Inspiration, bzw. ein zusätzlich zu auszuatmendes Volumen im Falle der Expiration, das als PLEURAL-VOLUMEN bezeichnet werden soll. Aus der Compliance des Thorax und dem Pleural-Volumen kann der intrapleurale Druck bestimmt werden. Es wird angenommen, dass sich dieser intrapleurale Druck ohne Zeitverzögerung auf den intrapulmonalen Druck der Lunge und deren Kompartimente überträgt, so dass sich der intrapleurale Druck als zusätzlich wirkender Druck im Atemzug niederschlägt. Diese Zusammenhänge stellen sich in den Modellgleichungen folgendermaßen dar:

Volumengenerator. Als Grundlage der zeitabhängigen Generatorkurve ist die Funktion $f(t) = t^{20}$ für $0 \leq t < 1$ gewählt worden, die erstens zu jedem Zeitpunkt das Gewünschte als „Soll-Volumen-Kurve“ leistet, falls sie geeignet verschoben wird, und zweitens einfach ist. Zunächst muss eine Verschiebung nach oben sicherstellen, dass bei der Inspiration die Kurve stets über dem aktuellen Lungenvolumen liegt. Als Mindestvolumen dient das theoretische Mindestvolumen V_{MIN} in der Lunge nach einer perfekten Expiration, dem Produkt aus dem außen angelegten Druck PEEP und der Compliance der Lunge. In der Ausnahmesituation, dass der PEEP Null sein sollte, wird ein Mindestvolumen in Höhe der Startwerte der am Lungenvolumen beteiligten Kompartimente angenommen.

$$V_{\text{MIN}} = \begin{cases} \text{PEEP} * C_{\text{Lunge}} & , \text{PEEP} \neq 0 \\ \sum_{k \in K} \text{Vol}_k(0) & , \text{PEEP} = 0 \end{cases} \quad (8.30)$$

Darüber hinaus wird die Ausgangsfunktion mit einem Faktor Par_{Aug} gestreckt. Dieser Faktor liefert eine Abschätzung der Augmentierung und wird in dieser Modellierung in ml gemessen. Während der Inspirationszeit T_I ist zusätzlich nicht die Funktion selbst, sondern deren Umkehrung $f^{-1}(t) = \sqrt[20]{t}$ zu betrachten. In der Expirationszeit T_E ist die ursprüngliche Funktion im Intervall $0 \leq t < 1$ in umgekehrter Richtung zu durchlaufen. So dass das Zeitintervall $[T_I, T_I + T_E)$ auf das Intervall $[0, 1)$ transformiert werden muss. Insgesamt ergibt dies als Augmentierungsfunktion $\text{Vol}_{\text{Aug}}(t)$, die schematisch in Abbildung 8.6 dargestellt wird.

$$\text{Vol}_{\text{Aug}}(t) = \begin{cases} \text{Par}_{\text{Aug}} * \sqrt[20]{\frac{t}{T_I}} + V_{\text{MIN}} & , 0 \leq t \leq T_I, \\ \text{Par}_{\text{Aug}} * \left(1 - \frac{t-T_I}{T_E}\right)^{20} + V_{\text{MIN}} & , T_I \leq t < T_I + T_E. \end{cases} \quad (8.31)$$

Der Generator als Druckgenerator. In der Physiologie ist es üblich, die Spontanaktivität als Druck anzugeben. Die Eigenaktivität hängt dann auch von der Compliance

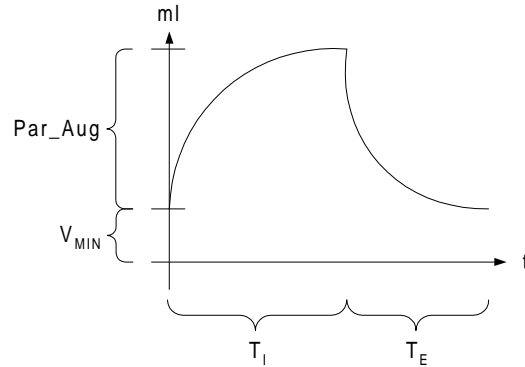


Abbildung 8.6: Schematische Darstellung der Augmentierungsfunktion

Thorax ab. Mit Hilfe dieser Thorax-Compliance C_{Thorax} kann der Volumengenerator in einen Druckgenerator überführt werden. So ergibt sich die gewohnte Darstellung der Spontanaktivität.

$$P_{\text{Aug}}(t) = \frac{\text{Vol}_{\text{Aug}}(t)}{C_{\text{Thorax}}}. \quad (8.32)$$

Insbesondere folgt als Parameter der Augmentierung ein weiterer Wert, der in mbar angegeben wird.¹¹

$$\widetilde{\text{Par_Aug}} = \frac{\text{Par_Aug}}{C_{\text{Thorax}}}. \quad (8.33)$$

Der intrapleurale Druck. Da nunmehr das gesamte im Thorax vorhandene Volumen, sowie aus der Simulation das gesamte Atemzugvolumen bekannt sind, kann aus diesen Informationen zusammen mit der Thorax-Compliance eine Kennzahl für den intrapleurale Druck $\Delta P_{\text{Aug}}(t)$ zwischen Thorax und Pleuralblättern bestimmt werden:

$$\Delta P_{\text{Aug}}(t) = \frac{\text{Vol}_{\text{Aug}}(t) - \text{AZV}(t)}{C_{\text{Thorax}}}. \quad (8.34)$$

Der Druck in den Kompartimenten. Ziel der Darstellung ist, den Druckgenerator $\text{PG}_{\text{Aug}}(t)$ aus Gleichung (8.29) zu konkretisieren. Dies kann unter der vereinfachenden Annahme, dass sich der intrapleurale Druck ohne Zeitverzögerung als intrapulmonalen Druck auf den flussrelevanten Druckgradienten in der Lunge auswirkt, durch

$$\text{PG}_{\text{Aug}}(t) = -\Delta P_{\text{Aug}}(t) \quad (8.35)$$

realisiert werden. Dabei ist auf das Vorzeichen zu achten: Intrapleurale und intrapulmonale Druck haben verschiedene Vorzeichen. Insgesamt ergibt sich als Druck in den Kompartimenten:

¹¹Zwar ist eine Schätzung der als Augmentierung bezeichneten Spontanaktivität nur bedingt Ziel des Modells, doch lässt sich sagen, dass die gefundenen Werte nicht außerhalb des physiologisch sinnvollen Bereichs liegen.

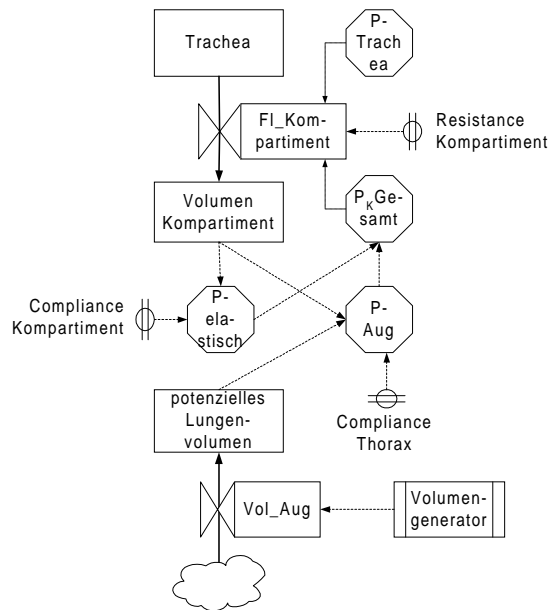


Abbildung 8.7: Die Modellierung der Spontanaktivität als L/R-Modell

$$P_k(t) = \frac{\text{Vol}_k(t)}{C_K} - \Delta P_{\text{Aug}}(t). \quad (8.36)$$

Atemzüge mit Augmentierung. Wenn der Patient Eigenaktivität hat, kann er selbst mandatorische Atemzüge auf P_{Insp} -Niveau einleiten. Diese erzeugen dann ein Atemzugvolumen, das größer ist, als es durch die Druckunterstützung des Respirators zu erklären ist. Umgekehrt wird der Patient auch maschinelle, aber nicht mandatorische Atemzüge möglicherweise unterstützen. Da die Modellierung der Spontanaktivität der Blutgase untergeordnet ist, wirkt der Druckgenerator auf alle Atemzüge, so dass durch die Augmentierung das gesamte gemessene Volumen erreicht wird. Mit dieser Modellierung muss nicht zwischen maschinellen und mandatorischen Atemzügen unterschieden werden (vgl. 8.3).

8.6.3 Das L/R-Modell der Spontanaktivität

Ausgehend vom völlig maschinell atmenden Patienten und dem zugehörigen L/R-Modell ist über dessen elektrotechnische Formulierung die Eigenaktivität in das physiologische Modell integriert worden. Im letzten Schritt wird der Volumenteil des Modells vervollständigt, in dem die gefundenen Zusammenhänge wieder in die Terminologie von L/R-Modellen übertragen werden.

Hierzu zeigt Abbildung 8.7 im oberen Teil eines der Kompartimente, wie sie bisher modelliert worden sind. Im unteren Teil wird die Augmentierung dargestellt. Das Level potenzielles Lungenvolumen stellt die Kurve der Augmentierungsfunktion $\text{Vol}_{\text{Aug}}(t)$ dar, wie sie in Gleichung (8.31) für einen Atemzug angegeben ist. Sie entsteht durch die Rate Vol_Aug , die ihrerseits von einem Volumengenerator, das als Signal realisiert ist, gesteuert wird. Dieses Signal entspricht der Ableitung von $\text{Vol}_{\text{Aug}}(t)$, so dass durch Integration

über die Rate die gewünschte Funktion entsteht. Über die Thorax-Compliance C_{Thorax} und die beiden Volumina wird im Parameter P-Aug der intrapleurale Druck $\Delta P_{\text{Aug}}(t)$ berechnet. Er bildet zusammen mit dem elastischen Druck den für den Druckgradienten entscheidenden Gesamtdruck im Kompartiment P_k -Gesamt gemäß Gleichung (8.36).

8.7 Modellierung der Perfusion

Das Herz-Zeit-Volumen (kurz: HZV oder Q) wird als Konstante aus dem Körpergewicht des Patienten abgeschätzt. Es beträgt gemäß einer allgemeinen Faustregel 70 ml/min pro kg Körpergewicht.

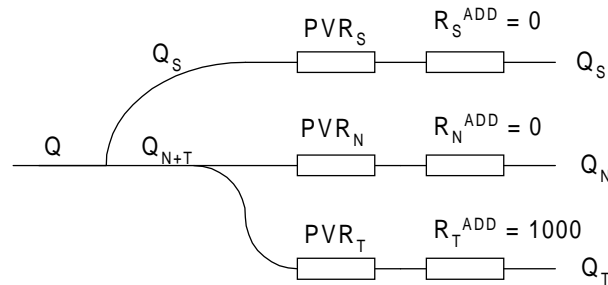


Abbildung 8.8: Aufteilung des Herz-Zeit-Volumens mit Hilfe von Gefäßwiderständen

Das HZV wird nach Diagramm 8.8 auf die Kompartimente verteilt: Es spaltet sich an zwei Verzweigungsstellen auf. Zunächst wird das Shunt-Kompartiment mit Blut versorgt. Der verbleibende Teil des Volumens wird auf das Normal- und das Totraum-Kompartiment aufgeteilt. Dies erfolgt über den Gefäßwiderstand PVR (engl. Pulmonary Vascularly Resistance). Ausgehend von einem Lungengefäßwiderstand von $PVR_{\text{Gesund}} = 0,18 \text{ mbar} \cdot \text{s/ml}$ eines gesunden Menschen wird gemäß der relativen Größe des Kompartimentes PZ_k und der Widerstandsregel für parallele Widerstände D.3 der jeweilige Gefäßwiderstand der Kompartimente PVR_k bestimmt. Die Gefäße erhalten durch einen zweiten in Reihe geschalteten Widerstand R_k^{add} ihre Funktion:

$$PVR_k = \frac{PZ_k}{PVR_{\text{Gesund}}} \quad \text{und} \quad PVR_k^{\text{Tot}} = PVR_k + R_k^{\text{add}}.$$

In der Praxis heißt dies, dass der Totraum durch einen großen zusätzlichen Widerstand de facto nicht perfundiert wird, während der zusätzliche Widerstand der beiden anderen Kompartimente auf Null gesetzt wird.

8.8 Modellierung der Diffusion

Ausgehend von der Beschreibung der Diffusion im Kapitel 7.4 wird ein Ansatz zu deren Modellierung angegeben und der Unterschied zu den Verhältnissen am Patienten dargestellt. Für die genaue Modellierung werden Bindungskurven des Kohlendioxids und des Sauerstoffs benötigt, deren Realisierungen im Anhang E.2 dokumentiert sind.

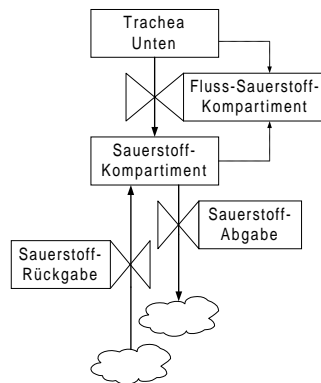


Abbildung 8.9: Sauerstoffdiffusion im Kompartiment

8.8.1 Das grundsätzliche Vorgehen

Diffusion in der Lunge geschieht, wenn venöses Blut in den Kapillargefäßen mit der Alveolarluft in Kontakt tritt, wobei die Flüssigkeit und Gas nur von einer dünnen Membranschicht getrennt sind. Dabei werden die Partialdruckdifferenzen von Sauerstoff und Kohlendioxid ausgeglichen, indem das Blut Sauerstoff aufnimmt und Kohlendioxid abgibt. Im physiologischen Modell werden die Einheiten aus Kapillargefäß und Alveole zu Kompartimenten mit gleichem VPV (vgl. Gleichung (7.11)) zusammengefasst: dem Normal-, Totraum- und Shunt-Kompartiment. Dies sind die Stellen, an denen für jedes Kompartiment im Prinzip in der gleichen Weise die Diffusion stattfindet. Lediglich auf den Shunt als nichtbelüftetes Kompartiment muss wieder genauer eingegangen werden. In der Realität findet ein Kontakt zwischen dem Kapillarblut und der Alveolarluft statt und nur zusätzlicher Sauerstoff kommt hinzu bzw. überschüssiges Kohlendioxid wird abgegeben, wie es in Abbildung 7.9 angedeutet wird. Einfacher zu modellieren ist dieser Austausch dadurch, dass der gesamte Sauerstoff vom Blut in die Alveole zurückkehrt und diese wiederum den zum Druckausgleich notwendigen Sauerstoff an das Blut abgibt. Dabei wird davon ausgegangen, dass die verschiedenen Partialdrücke sofort ausgeglichen und kein Diffusionswiderstand zu überwinden ist (vgl. Abschnitt 7.4.4). Analoges gilt für das Kohlendioxid.

Diese Zusammenhänge führen in der Symbolik der Level-Raten-Modelle zum Diagramm 8.9, das beispielhaft die Situation beim Sauerstoff darstellt: Die Alveole wird als Sauerstoff-Kompartiment modelliert. Es wird mit Sauerstoff aus der unteren Trachea versorgt. Die beiden unteren Raten, die in einer Senke enden, stellen zum einen die *Rückgabe* des Sauerstoffs aus dem „venösen“ Blut dar, während die *Abgabe* des Sauerstoffs in das arterielle Gefäßsystem erfolgt. Der Einfachheit halber ist auf die Modellierung eines „Blutpools“ im Körper verzichtet worden. Die genauere Darstellung der Vorgänge der Diffusion erfolgt anhand dreier Teilprozesse:

1. Die alveolare Arterialisierung.
2. Der Gasaustausch in den Gefäßen.
3. Der venöse Rückfluss.

Für die technische Umsetzung haben die Rechnungen in Abschnitt 7.4.2 gezeigt, dass diese Prozesse bei der Betrachtung verschiedener Kompartimente einfach zu modellieren sind, wenn die GEHALTE (CONTENTS) von Sauerstoff und Kohlendioxid benutzt werden. Dies führt auf die Betrachtung der Bindungskurven, die den Abschnitten über die drei Teilprozesse vorangeht.

8.8.2 Die Bindungskurven

Grundlage für die Operationalisierung des Transports des Sauerstoffs und des Kohlendioxids sind die Erläuterungen aus den Abschnitten 7.3.2 und 7.3.3. Hierzu bieten sich die im Anhang abgebildeten Tabellen (Abbildung der Sauerstoff-Bindungskurve E.3 und der Kohlendioxid-Bindungskurve E.2), die auf Messreihen beruhen. Im physiologischen Modell werden diese Zusammenhänge für beide Bindungskurven in zwei Richtungen benötigt:

1. *Druck* \rightarrow *Gehalt*: Aus dem Partialdruck in der Alveole soll der Gehalt im Kapillargefäß bestimmt werden.
2. *Gehalt* \rightarrow *Druck*: In den Arterien wird aus dem Gehalt der jeweilige Partialdruck berechnet, da dieser gemäß des Modellzwecks an das Expertensystem ausgegeben werden soll.

Die O₂-Bindungskurve

Um die in der Tabelle aus Abbildung E.3 im Anhang gegebene Bindungskurve zu realisieren, werden als weitere Parameter die Temperatur, der Basenüberschuss und der pH-Wert benötigt. Sie werden im Zuge einer Blutgasanalyse bestimmt. Anhand der Tabelle bestimmt man aufgrund dieser Informationen aus dem gemessenen einen virtuellen Sauerstoffpartialdruck, der zu einer Sättigung führt. Damit lässt sich aufgrund von Gleichung (7.7), die hier wiederholt wird, der Gehalt bestimmen:

$$C_{aO_2} = [\text{Hb}] * S_{aO_2} * 1,34 + \tilde{\alpha} * P_{aO_2} \quad (8.37)$$

Die Tabelle ist mit Stützstellen, wie sie im Anhang in Tabelle E.3 angegeben sind, implementiert. Zwischen den Stützstellen wird linear interpoliert. Dies liefert die beiden benötigten Funktionen: Ist der Druck bekannt, wird aus der Sauerstoff-Bindungskurve die Sättigung und daraus der Gehalt gemäß obiger Gleichung bestimmt. Umgekehrt ist ein Iterationsverfahren notwendig. Hierzu wird als initiale Näherung die Sättigung aus dem Gehalt geschätzt, indem auf den physikalisch gelösten Sauerstoff (zweiter Summand der Gleichung) verzichtet wird.

Die CO₂-Bindungskurve

Wieder ist zunächst entsprechend Abbildung E.2 eine Tabelle mit physiologischen Messwerten implementiert worden. Hierzu gibt man die entsprechenden Werte als Tabelle ein (vgl. Tabelle E.1 im Anhang), interpoliert dazwischen linear und extrapoliert auf höhere Gehalte. Ein Vorteil dieser Methode ist ihre Einfachheit. Darüber hinaus spricht für diese

Kurve, dass sie sich in der Atmungsphysiologie als typisch durchgesetzt hat. Der Nachteil besteht offensichtlich darin, dass physiologische Besonderheiten von Patienten wie zum Beispiel die gemessene Hämoglobin-Konzentration nicht in die Kurve integriert sind.

Die zuletzt genannten Nachteile werden von einem Algorithmus verhindert, der auf einer Arbeit von Kelman beruht. Er wird im Anhang E.2.1 nach [WW77] dargestellt. Bei einem experimentellen Vergleich der beiden Bindungskurven hat sich gezeigt, dass der Unterschied zwischen ihnen kaum festzustellen ist, so dass der einfacheren, auf experimentellen Daten beruhenden Kurve der Vorzug gegeben worden ist.

8.8.3 Die alveolare Arterialisierung

Zur Mischung des alveolaren zum arteriellen Blut werden die zugehörigen Gehalte betrachtet. Hierzu sind bei der Arterialisierung, wie sie in Abschnitt 7.4.1 beschrieben worden ist, in den Kapillaren drei Rechenschritte nötig. Diese sind in den (belüfteten) Kompartimenten mit Ausnahme des Shunts für Sauerstoff und Kohlendioxid identisch, so dass die Gleichungen mit $\text{Ele} \in \{\text{O}_2, \text{CO}_2\}$ indiziert werden:

$$\text{Pa}_k^{\text{Ele}}(t) = \max \left(1, (\text{Barometerdruck} - 47) * \text{Fr}_k^{\text{Ele}}(t) \right), \quad (8.38)$$

$$\text{Ca}_k^{\text{Ele}}(t) = \text{ContentAusDruck}(\text{Pa}_k^{\text{Ele}}(t)), \quad (8.39)$$

$$\text{FL_Ab}_k^{\text{Ele}}(t, t + \Delta t) = Q_k * \text{Ca}_k^{\text{Ele}}(t). \quad (8.40)$$

Aus den Fraktionen, die gemäß Gleichung (8.24) in Abschnitt 8.5.3 berechnet werden, wird der Partialdruck in Gleichung (8.38) bestimmt. Dabei wird das Daltonsche Gesetz 7.2 verwendet. Da aber die Fraktionen in der Lunge unter BTPS-Bedingungen, den Normbedingungen in der Lunge (vgl. Abschnitt 7.2.3), angegeben werden, und der aktuelle Barometerdruck auch den Partialdruck des Wasserdampfes (47 mmHG) enthält, muss dieser vom Barometerdruck abgezogen werden. Durch das Maximum werden ggf. unsinnige Werte verhindert, die zum Beispiel bei nicht stimmigen Startwerten entstehen können.

Es wird angenommen, dass sich der Partialdruck umgehend und ohne Diffusionswiderstand auf das Kapillargefäß überträgt. Mit diesem Druck wird der Gehalt im Blut über die Funktion $\text{Druck} \rightarrow \text{Gehalt}$, ContentAusDruck , in Gleichung (8.39) bestimmt. Entsprechend der Berechnung in Kapitel 7.4.2 zur Definition des Shunts ergibt sich der Abfluss des arterialisierten Blutes in Gleichung (8.40) aus dem Produkt des entsprechenden Anteil des Herz-Zeit-Volumens Q_k und dem Gehalt.

Handelt es sich um ein Shunt-Kompartiment, das nicht belüftet ist, so ist der alveolare Gehalt und damit auch der Gehalt im Kapillar nicht sinnvoll zu berechnen. Daher wird zur Berechnung der venösen Beimischung der gemischt-venöse Gehalt $C_{\bar{v}}$, der sich aus der Modellierung der Gefäße ergibt, benutzt.

$$\text{FL_Ab}_{\text{Shunt}}^{\text{Ele}}(t, t + \Delta t) = Q_{\text{Shunt}} * C_{\bar{v}}^{\text{Ele}}(t). \quad (8.41)$$

8.8.4 Der Gasaustausch in den Gefäßen

Nachdem in den Kapillaren das Blut arterialisiert worden ist, mischt es sich in den Arterien. Dies führt zu einem arteriellen Gehalt und Partialdruck von Kohlendioxid und Sauerstoff, den Zielgrößen des Modells aus dem Modellzweck. Ihre Bestimmung erfolgt gemäß den folgenden Gleichungen:

$$\text{FL_Ab}_{\text{Tot}}^{\text{Ele}}(t, t + \Delta t) = \sum_k \text{FL_Ab}_k^{\text{Ele}}(t, t + \Delta t), \quad (8.42)$$

$$C_A^{\text{Ele}}(t) = \frac{\text{FL_Ab}_{\text{Tot}}^{\text{Ele}}(t, t + \Delta t)}{Q}, \quad (8.43)$$

$$P_A^{\text{Ele}}(t) = \text{DruckausContent} \left(C_A^{\text{Ele}}(t) \right). \quad (8.44)$$

Zunächst wird durch Summation über die Kompartimente die Gesamtabgabe bestimmt. Daraus ergibt sich der jeweilige arterielle Gehalt $C_A^{\text{Ele}}(t)$. Mit Hilfe der Bindungskurven wird der Partialdruck $P_A^{\text{Ele}}(t)$ berechnet.

Für das Weitere ist der respiratorische Quotient RQ ein bestimmender Faktor. Dieser ist in Abschnitt 7.6 als das Verhältnis CO_2 -Produktion und O_2 -Aufnahme in der Lunge definiert worden. Da auf die Modellierung eines „Blutpools“ verzichtet worden ist, stellt das aktuell arterialisierte Blut das gesamte Blut dar. In den Gefäßen wird ihm Sauerstoff entzogen und Kohlendioxid zugeführt. Daraus ergibt sich der Gehalt von Sauerstoff und Kohlendioxid in den Venen:

$$C_{\bar{V}}^{\text{O}_2} = \frac{\text{FL_Ab}_{\text{Ges}}^{\text{O}_2} - \text{O}_2\text{-Aufnahme}}{Q}, \quad (8.45)$$

$$C_{\bar{V}}^{\text{CO}_2} = \frac{\text{FL_Ab}_{\text{Ges}}^{\text{CO}_2} + \text{CO}_2\text{-Produktion}}{Q}. \quad (8.46)$$

Da nur ein Gefäß modelliert worden ist, handelt es sich dabei um den gemischt-venösen Gehalt.

8.8.5 Der venöse Rückfluss

Das gemischt-venöse Blut fließt zurück in die Kapillargefäße, in denen dessen Sauerstoff und Kohlendioxid gemäß dem Modellierungsansatz wieder völlig in die Alveolen zurückkehrt. Damit ergibt sich als Rückfluss in die Kompartimente:

$$\text{FL_Rueck}_k^{\text{Ele}}(t, t + \Delta t) = Q_k * C_{\bar{V}}^{\text{Ele}}(t).$$

8.8.6 Einfluss des RQ auf die Ventilation

In der Praxis ist der respiratorische Quotient kleiner Eins. Er wird im physiologischen Modell mit dem Standardwert 0,8 abgeschätzt. Dies bedeutet, dass mehr Sauerstoff in

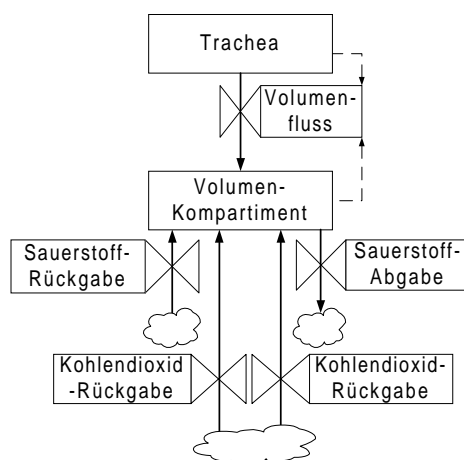


Abbildung 8.10: Diffusion im Volumen-Kompartiment

den Zellen aufgenommen als Kohlendioxid abgegeben wird. Dieses Volumen „geht verloren“. Dies hat auch einen Einfluss auf das Atemzugvolumen, was dazu geführt hat, dass die Ab- und Rückgabe von Sauerstoff und Kohlendioxid auch in das „Volumenmodell“ integriert worden ist, wie es Abbildung 8.10 zeigt. Daraus lassen sich die vollständigen Levelgleichungen der Volumen-Kompartimente ableiten: Das Volumen-Kompartiment erhält je zwei weitere Zu- und Abflüsse, die aus dem Diffusionsmodell stammen. Somit können aus der Abbildung die endgültigen Level-Gleichungen der Volumenkompartimente abgelesen werden, was in Abschnitt 8.4.2 noch nicht möglich war:

$$\text{Vol}_k(t + \Delta t) = \text{Vol}_k(t) + \Delta t \left(\text{FL}_k^{\text{Vol}}(t, t + \Delta t) + \sum_{\text{Ele} \in \{\text{O}_2, \text{CO}_2\}} \left(\text{FL_Ab}_{\text{Ges}}^{\text{Ele}} - \text{FL_Rueck}_{\text{Ges}}^{\text{Ele}} \right) \right) \quad (8.47)$$

8.9 Algorithmen des physiologischen Modells

8.9.1 Eine Simulation

Wie es bei zeitorientierten Simulationen üblich ist (vgl. Abschnitt 2.2), besteht eine Simulation im physiologischen Modell aus zwei Phasen. Zunächst wird die Simulation initialisiert und dann die gewünschte Anzahl der Simulationsschritte in einer Schleife durchgeführt. Wie in dem Abschnitt zu den Zeitskalen dargestellt, sind im physiologischen Modell die auf Atemzüge und Minuten aggregierten Werte von Interesse. Letztere ergeben sich aus den auf Atemzügen aggregierten Werten. So ist z.B. das Atemminuten-Volumen die Summe der Atemzugvolumina. Daher wird sich ganzzahlige Atemzugsfrequenzen beschränkt. So ist im physiologischen Modell ein doppelter Schleifendurchlauf geeignet: Es wird eine Folge von Atemzügen simuliert. Innerhalb eines Atemzugs wird auf der sekundlichen Zeitskala simuliert.

Die Initialisierung. Die Initialisierungsphase besteht aus drei Schritten: Zunächst werden die Startwerte gesetzt. Entweder es werden die zuletzt verwendeten Startwerte benutzt, oder es werden die Endzustände der vorhergehenden Simulation benutzt. Letzteres entspricht einem Voranschreiten in der Zeit. Eine Ausnahme macht der Startwert der „unteren Trachea“ $TR_{\text{Unten}}^{\text{Vol}}$, da mit dem V_{d_s} stets ein passender Messwert zur Verfügung steht. Anschließend wird die interne Verwaltung der berechneten Werte zurückgesetzt. Schließlich erfolgt die initiale Berechnung der Raten und der zeitabhängigen Größen wie Beatmungsdruck und Augmentierung.

Simulation eines Atemzugs. Zu Beginn der Simulation eines Atemzugs wird entschieden, ob dieser Atemzug spontan ist; dementsprechend wird die Inspirationszeit gesetzt. Dann wird innerhalb einer Schleife solange simuliert, bis die Dauer des Atemzugs beendet ist. In einem Schleifendurchlauf werden die zeitabhängigen Signalen der beiden Druckgeneratoren von Beatmungsgerät und Spontanaktivität bestimmt, das PEEP-Ventil bei der Expiration als zusätzlicher Widerstand in der Trachea einbezogen, ein Integrationsschritt durchgeführt und die Simulationsergebnisse gespeichert, wenn eine Speicherfrequenz (vgl. Kapitel 5.2.3) erreicht ist.

8.9.2 Kalibrierung der Mechanik

Ziel der Kalibrierung der Mechanik ist, die Volumen-Kalibrierungsaufgabe aus Abschnitt 8.1 zu lösen: Die Kalibrierungsparameter der Mechanik sind so einzustellen, dass die gemessenen und die simulierten Atemminuten-Volumina übereinstimmen. Als Kalibrierungsparameter stehen die Resistance und Compliance der einzelnen Kompartimente zur Verfügung. Diese werden im ersten Schritt auf sinnvolle Werte gesetzt. Anschließend werden diese Einstellungen getestet.

Bestimmung der Parameter. Ausgangspunkt der Kalibrierung der Mechanik ist die Darstellung des Ventilations-Modells in seiner elektrotechnischen Formulierung in Abbildung 8.4 sowie die im Zusammenhang mit dem EWS vorhandenen Daten. Für das physiologische Modell ist die Compliance des „Gesamtsystems Patient“ ein Übergabe-Parameter. Bei rein maschineller Atmung kann man die Compliance auch aus dem Atemzugvolumen und den Beatmungsdrücken berechnen.¹² Da aber bei einsetzender Eigenatmung diese Rechnung nicht mehr durchgeführt werden darf und auch keine stets gültige Berechnungsmethode bekannt ist, erfolgt die Feststellung der Compliance stets über Expertenwissen im EWS. Die aus dem EWS übergebene Compliance wird im Folgenden mit C_{gem} bezeichnet. Die effektive Lungen-Compliance C_{Lunge} ergibt sich gemäß dem Schaltbild 8.4 und den Kapazitätsregeln D.4:

$$C_{\text{Lunge+Thorax}} = C_{\text{gem}} - C_{\text{Trachea}}, \quad (8.48)$$

$$\frac{1}{C_{\text{Lunge}}} = \frac{1}{C_{\text{Lunge+Thorax}}} - \frac{1}{C_{\text{Thorax}}}. \quad (8.49)$$

¹²Vgl. [Nun87] Appendix F.

Die Compliance-Werte der Trachea und des Thorax werden mangels genauerer Informationen auf einen konstanten Standardwert gesetzt.¹³

Es bleiben die Größen der beiden Widerstände und die Kapazitäten der beiden Kondensatoren „Normal“ N und „Totraum“ T zu berechnen. Es sollen sich die einzelnen Compliances C_k der Kompartimente zur Compliance der Lunge gemäß Gleichung (8.50) summieren, dabei hat der Shunt nach dem Dreikompartimente-Modell eine Compliance von Null.

$$C_{\text{Lunge}} = C_N + C_T, \quad (8.50)$$

$$C_N = \frac{PZ_N}{PZ_N + PZ_T} * C_{\text{Lunge}}, \quad (8.51)$$

$$C_T = \frac{PZ_T}{PZ_N + PZ_T} * C_{\text{Lunge}}. \quad (8.52)$$

An dieser Stelle kann man davon ausgehen, dass die relativen Größen PZ_N und PZ_T der Kompartimente bekannt sind, obwohl sie erst bei der Kalibrierung der Blutgase als Kalibrierungsparameter bestimmt werden.¹⁴ Deren relative Größe ist so zu verteilen, dass keine Compliance „verloren geht“ und die Kompartimente einen Anteil an der Atmung gemäß ihrer relativen Größe haben sollen. Dies ist gewährleistet, wenn die Compliance der beiden Kompartimente wie in den Gleichungen (8.51) und (8.52) eingestellt werden.

Zur Berechnung der Widerstände muss ein Umweg beschritten werden, da die Zeitkonstante τ während der Expiration gemessen wird (vgl. Kapitel 7.2.1) und nach Abschnitt 7.1.2 bei der Expiration die Luft den zusätzlichen Widerstand des PEEP-Ventil überwinden muss.

$$R_{\text{Ges}} = \frac{\tau}{C_{\text{gem}}}, \quad (8.53)$$

$$R_{N+T} = R_{\text{Ges}} - (R_{\text{PEEP-Ventil}} + R_{\text{Trachea}}) \quad (8.54)$$

Die rechnerische Gesamt-Resistance R_{Ges} in Gleichung (8.53) muss noch gemäß Gleichung (8.54) modifiziert werden. Dabei bedeutet der Index $N + T$, dass die Gesamtresistance der beiden Kompartimente Normal und Totraum gemeint ist. Die Widerstandsregel für parallele Widerstände führt bei einem nichtbelüfteten Shunt mit „unendlich großem Widerstand“ zu folgenden Zusammenhängen:

$$\frac{1}{R_{N+T}} = \frac{1}{R_N} + \frac{1}{R_T}, \quad (8.55)$$

$$R_N = \frac{PZ_N + PZ_T}{PZ_N} * R_{N+T}, \quad (8.56)$$

$$R_T = \frac{PZ_N + PZ_T}{PZ_T} * R_{N+T}. \quad (8.57)$$

¹³Die Compliance der Trachea $C_{\text{Trachea}} = 2$ ist klein, so dass die Vorstellung von in Reihe geschalteten Kondensatoren in Gleichung (8.48) zu einer negativen Lungen-Compliance führen würde, da sich in diesem Fall die Kehrwerte zum Kehrwert der Compliance addieren würden. Die Compliance des Thorax C_{Thorax} wird mit 200 ml/mbar abgeschätzt.

¹⁴Durch die Änderung der relativen Größe der Kompartimente ändert sich das Atemminuten-Volumen nur minimal, so dass hier die initiale Standardnäherung benutzt werden kann.

Dabei werden wiederum die Anteile relativ zu deren Gesamtgröße verwendet.

Testen der Einstellungen. Der dargestellte Algorithmus führt dazu, dass die Zeitkonstanten $\tau_k = R_k C_k$ der beiden an der Atmung teilnehmenden Kompartimente übereinstimmen und so in ihnen zu jedem Zeitpunkt derselbe Druck herrscht. Diese Rechnungen stellen sicher, dass das zur Compliance der Lunge gehörende Volumen auch im Modell geatmet wird. Allerdings besteht immer noch die Möglichkeit, dass die übergebene Compliance nicht den tatsächlichen Verhältnissen entspricht, da der Patient durch Eigenaktivität zusätzliches Volumen atmet. Daher wird zum Abschluss der Kalibrierung das Modell auf seinen Ventilationsteil beschränkt und eine Simulation durchgeführt. Dies ist möglich, da die übrigen Teile nur unwesentlich über den respiratorischen Quotienten Einfluss auf das Atemminuten-Volumen haben. Dann wird im Expertensystem entschieden, ob die Volumina übereinstimmen. Da diese Simulation in allen Stadien der Entwöhnung mit dem Modell aufgerufen werden soll, muss gewährleistet sein, dass die Startwerte der beteiligten Level stets im plausiblen Bereich liegen. Daher werden sie auf das Produkt aus Compliance und endexpiratorisch herrschendem Druck gesetzt.

Ohne auf die genauen Regelungen im EWS eingehen zu können, ist hier eine Bemerkung zum grundsätzlichen Zusammenspiel angebracht: Das EWS unterschätzt tendenziell die Compliance der Lunge, so dass das Modell erst durch eine zusätzliche Augmentierung das gewünschte Volumen erreicht. Im umgekehrten Fall müsste man durch negative Augmentierungen das simulierte Atemminuten-Volumen verringern. Dies wäre physiologisch nicht zu begründen. Wie die Augmentierung bestimmt wird, zeigt der folgende Abschnitt.

8.9.3 Bestimmung der Augmentierung

Ziel ist, die Augmentierung so einzustellen, dass das gemessene und das simulierte Atemminuten-Volumen übereinstimmen. Dies ist nur notwendig, wenn das simulierte Volumen kleiner ist als das gemessene, da ansonsten die Compliance verringert werden muss. Zunächst wird die aktuelle Parametrisierung des Modells getestet. Im zweiten Schritt wird in 1000 ml-Schritten die Augmentierung in die passende Richtung verändert. Wenn das Zielvolumen zwischen zwei Näherungen liegt, wird über Intervallhalbierung die genaue Größe bestimmt. Als Sicherheitsgrenzen werden folgende Werte eingeführt:

| Konstante | Wert |
|-----------------------|--|
| min. Augmentierung | 0 |
| max. Augmentierung | $60 \text{ mbar} * 200 \frac{\text{ml}}{\text{mbar}} = 12000 \text{ ml}$ |
| tolerierte Abweichung | 5 % |
| initiale Schrittweite | $5 \text{ mbar} * 200 \frac{\text{ml}}{\text{mbar}} = 1000 \text{ ml}$ |

Da in der Medizin die Augmentierung in mbar gemessen wird, sind die Konstanten als Vielfachen der Thorax-Compliance von 200 ml/mbar angegeben worden. Somit gibt das Vielfache das Maß für die Augmentierung in mbar an. Würde der Algorithmus eine Augmentierung finden, die kleiner als 0 mbar oder größer als 60 mbar beendet der Algorithmus mit einer Fehlermeldung.

8.9.4 Einschwingen: Steady-State des physiologischen Modells

Das physiologische Modell bildet einen Patienten in dessen stabilen Zustand, dem „Steady-State“, ab. Da bei der Initialisierung die Startwerte des Modells mehr oder weniger zufällig gesetzt werden, ist ein Algorithmus **Einschwingen** zu entwerfen, der das Modell in einen stabilen Zustand überführt. Der Algorithmus arbeitet in drei Schritten.

Initiale Kalibrierungsparameter. Die initialen Kalibrierungsparameter des physiologischen Modells können in Abhängigkeit von einem booleschen Übergabeparameter gesetzt werden. Dadurch ist dem Anwender die Möglichkeit gegeben, auf das Setzen der Kalibrierungsparameter durch das Modell zu verzichten und selbst vor dem Einschwingen durch eigenes Expertenwissen eine initiale Näherung der Parameter vorzugeben bzw. die physiologischen Standardwerte zu benutzen. Im anderen Fall ermittelt das Modell mit Hilfe eines neuronalen Netzes (vgl. den kommenden Abschnitt 8.9.6) initiale Näherungen für die Kalibrierungsparameter und schwingt dann mit diesen Einstellungen ein.

Startwerte setzen. Für alle Level sind physiologische Standard-Startwerte festgelegt worden. Allerdings kann für die Volumen-Kompartimente aufgrund von deren Compliance C_k und der vorhandenen Eigenaktivität Par_Aug (vgl. Abschnitt 8.6.2) ein theoretisches Mindestvolumen berechnet werden:

$$C_k * \left(\text{PEEP} + \frac{\text{Par_Aug}}{C_{\text{Thorax}}} \right).$$

Der Grund für die Anpassung der Startwerte an einen Patienten ist, dass es aufgrund von nicht aufeinander angepassten Startwerten und Beatmungsdrücken zu Beginn der Simulation zu nicht abgestimmten Luftflüssen kommen und so z.B. die Trachea einen negativen Inhalt haben kann. Zusätzlich wird der Startwert des unteren Teils der Trachea auf den gemessenen seriellen Totraum gesetzt.

Blutgase stabilisieren. In dieser Phase wird solange simuliert bis der P_aO_2 stabil ist. Dabei wird bei der Messung der Stabilität der Blutgase folgendermaßen vorgegangen: Die Simulation des physiologischen Modells erfolgt im Sekunden-Takt. Diese Werte werden auf die Atemzugsebene aggregiert und diese aggregierten Werte wiederum zu „Minuten-Werten“ zusammengefasst. Die beiden Zeitaggregationsstufen liefern ein Maß für die Stabilität des Modells: MZ bezeichnet die Anzahl der Minuten, die im physiologischen Modell simuliert werden. P_aO_2^j entspricht dem über die j -te Minute gemittelten arteriellen Sauerstoffpartialdruck. $\text{P}_a\text{O}_2^j(i)$ ist der arterielle Sauerstoffpartialdruck des i -ten Atemzugs in der j -ten Minute. Als Fehlermaße für die Stabilität dienen die Kennzahlen err_s und err_1 bei einer Frequenz von f Atemzügen pro Minute:

$$\text{err}_s = \sum_{i=\lfloor \frac{f}{2} \rfloor}^f \sqrt{(\text{P}_a\text{O}_2^{\text{MZ}} - \text{P}_a\text{O}_2^{\text{MZ}(i)})^2}, \quad (8.58)$$

$$\text{err}_1 = \|\text{P}_a\text{O}_2^{\text{MZ}} - \text{P}_a\text{O}_2^{\text{MZ}-1}\|. \quad (8.59)$$

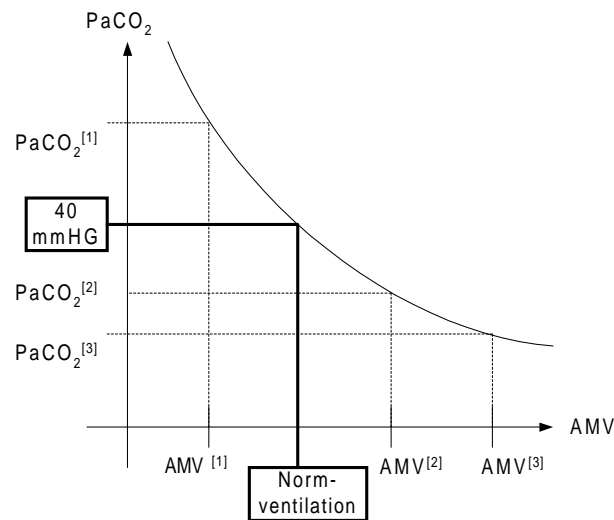


Abbildung 8.11: Das Vorgehen bei der Berechnung der Normventilation

Das Modell wird als nicht stabil angesehen, wenn sowohl das kurzfristige err_s als auch das langfristige Maß err_l größer als eine vorgegebene Konstante c , die letztendlich willkürlich auf $c = 5$ gesetzt worden ist. Der Grund für die Einführung zweier Maße liegt in der Abhängigkeit des P_aO_2 eines Atemzugs von dem angelegten Inspirationsdruck. Bei Unterschieden zwischen dem P_{ASB} und dem P_{Insp} kann der P_aO_2 von Atemzug zu Atemzug so stark schwanken, dass das kurzfristige Maß nicht eingehalten werden konnte, obwohl das Modell nach der langfristigen Betrachtung stabil ist. Andererseits kann sich das Modell erst während der Simulation der letzten Minute stabilisiert haben, so dass das langfristige Maß eine falsche Einschätzung liefern würde.

8.9.5 Berechnung der Normventilation

In Kapitel 7.5 ist die Normventilation als diejenige alveolare Ventilation definiert worden, die zu einem Partialdruck des Kohlendioxids von 40 mmHG in den Arterien führt. Der Algorithmus zur Berechnung der Normventilation beruht darauf, dass verschiedene Atemminuten-Volumina verschiedene P_aCO_2 -Werten ergeben. Führt man von einer gegebenen Ausgangssituation Simulationen mit verschiedenen Beatmungsdrücken durch, resultieren verschiedene Paare aus Atemminuten-Volumen und arteriellem Kohlendioxidpartialdruck. Da man nicht davon ausgehen kann, dass der Zusammenhang zwischen Ventilation und P_aCO_2 -Werten linear ist, müssen mindestens drei Simulationen durchgeführt werden. Bestimmt man die zu den Paaren aus Atemminuten-Volumina und Partialdrücken des Kohlendioxids gehörende Parabel, kann man die Normventilation als das Urbild von 40 mmHG bestimmen, wie es Diagramm 8.11 zeigt. Da der P_aCO_2 mit steigender Ventilation abnimmt, ist das Urbild aus dem fallenden Schenkel der Parabel der richtige der beiden möglichen Werte.

Da der Algorithmus zur Bestimmung der Normventilation im Verlauf einer Entwöhnung aufgerufen wird, ist es sinnvoll, einen der drei „Messpunkte“ aus der durch den Patienten gegebenen Ausgangssituation zu bestimmen. Um nach der Berechnung der Normventila-

tion keine weitere Laufzeit zu benötigen, wird diese Messung als letzte durchgeführt und kann so auch als „normale“ Simulation dienen. Für die Beatmungsdrücke der Simulationen ist zu beachten, dass in jedem Fall unterschiedliche Atemminuten-Volumina zustande kommen. Daher werden sowohl der P_{Insp} als auch der P_{ASB} verändert, und es muss gesichert werden, dass der Inspirationsdruck höher ist als das PEEP-Niveau und so überhaupt Atmung zustande kommt. Dies ist mit den folgenden Formeln realisiert worden:

$$P_{\text{Insp}}^{\text{neu}} = \max\left(\text{PEEP} + \frac{P_{\text{Insp}}}{2}, P_{\text{Insp}} \pm 10\right),$$

$$P_{\text{ASB}}^{\text{neu}} = \max\left(\text{PEEP} + \frac{P_{\text{ASB}}}{2}, P_{\text{ASB}} \pm 10\right).$$

Bei Simulationen mit diesen Inspirationsdrücken wird jedes Mal mit den aktuellen Startwerten gearbeitet, ohne die Zeit fortzuschreiben. Dies findet lediglich nach der letzten Simulation mit dem aktuellen Inspirationsdruck statt.

8.9.6 Die Kalibrierung der Blutgase

Die Blutgas-Kalibrierungsaufgabe besteht nach Abschnitt 8.1 darin, die Kalibrierungsparameter auszuwählen und deren Wert so einzustellen, dass erstens die gemessenen und simulierten Blutgase in einem vorgegebenen Toleranzbereich übereinstimmen und zweitens die vorhergesagten Blutgaswerte gültig ist. Im Folgenden wird beschrieben, wie Konzepte der Kalibrierung, die in Abschnitt 5.3 entwickelt worden sind, auf das physiologische Modell angewandt werden.

Das Vorgehen. Der Algorithmus besteht aus einem zweistufigen Verfahren. Zunächst wird mit Hilfe eines neuronalen Netzes initiale Näherung der Kalibrierungsparameter gesucht. Es ist durch Data Mining auf Modelldaten entstanden und arbeitet entsprechend Abbildung 5.15 als Kalibrierungsmodell: Mit einer geeigneten Teilmenge von Systemmesswerten als Eingabe erzeugt das Kalibrierungsmodell als Ausgabe die Werte der Kalibrierungsparameter. Im zweiten Schritt wird ein Iterationsalgorithmus angewandt, der sicherstellt, dass die schließlich gefundene Lösung im gewünschten Bereich ist.

Die initiale Näherung. Im Fall der Blutgase dienen die gemessenen Werte des Atemminuten-Volumen, $P_a\text{O}_2$, $P_a\text{CO}_2$, $F_1\text{O}_2$ und das aus dem Körpergewicht geschätzte Herz-Zeit-Volumen als Eingaben für das neuronale Netz. Die Kalibrierungsparameter sind die Shunt- und Totraumfraktion PZ_S und PZ_T . Durch die Auswahl Kalibrierungsparameter und der Modellstruktur als ein Dreikompartimente-Modell hat man über $PZ_N = 1 - PZ_S - PZ_T$ die Größe des Normal-Kompartimentes bestimmt. Abbildung B.4 im Anhang zeigt dies. Dort ist darüber hinaus noch die genaue Struktur des Netzes zu sehen. Es handelt es sich um ein dreischichtiges Feedforward-Netz, dessen Details im Anhang B.5 dargestellt werden.

| Konstante | Wert |
|---------------------|------|
| min. Normal-Anteil | 0,05 |
| max. Normal-Anteil | 0,95 |
| min. Shunt-Anteil | 0,01 |
| max. Shunt-Anteil | 0,7 |
| min. Totraum-Anteil | 0,01 |

Tabelle 8.1: Extremwerte der relativen Größe der Kompartimente

Der Iterationsalgorithmus. Der Iterationsalgorithmus zum Einstellen der Blutgase ist in zwei prinzipiell gleiche Teilalgorithmen aufgespaltet worden. Zunächst wird der $P_a\text{CO}_2$, anschließend der $P_a\text{O}_2$ eingestellt. Der Algorithmus zur Bestimmung des Partialdruckes des Kohlendioxids beruht darauf, dass dieser im Wesentlichen von der alveolaren Ventilation \dot{V}_A und diese wiederum direkt von der Größe des „Normal“-Kompartimentes PZ_N bestimmt wird. Dabei ist der Zusammenhang folgender:

$$PZ_N \uparrow \Rightarrow \dot{V}_A \uparrow \Rightarrow P_a\text{CO}_2 \downarrow \quad (8.60)$$

Je größer das Normal-Kompartiment, desto höher ist die alveolare Ventilation. Dadurch wird auch mehr Sauerstoff in die Alveolen übertragen und mehr Kohlendioxid aus ihnen eliminiert, so dass der Druck des Kohlendioxids in den Arterien sinkt. Umgekehrt steigt mit fallendem Normal-Anteil auch der arterielle Partialdruck des Kohlendioxids. Dabei werden die Änderungen des Normal-Kompartimentes durch entsprechende Veränderungen des Totraums bei konstantem Anteil des Shunt-Kompartimentes erreicht.

Hat man die Größe des Normal-Kompartimentes festgelegt, so folgt eine entsprechende Prozedur für die Kalibrierung des Sauerstoff-Partialdruckes. Hier ist der Zusammenhang über Shunt und Totraum gegeben:

$$PZ_S \uparrow \Rightarrow \text{venöses Blut in Arterien} \uparrow \Rightarrow P_a\text{O}_2 \downarrow \quad (8.61)$$

Je größer der Shunt ist, desto mehr venöses Blut fließt zurück in die Arterien, so dass dort der Anteil des Sauerstoffs sinkt und so auch der Partialdruck des Sauerstoffs fällt. Wieder werden die Änderungen am Shunt durch entsprechende Änderungen am Totraum-Kompartiment erreicht.

Der Effekt, den diese Änderung auf den Kohlendioxidpartialdruck hat, ist nur sehr gering. Daher arbeiten die beiden Algorithmen unabhängig voneinander. Schließlich ist das Iterationsverfahren derart implementiert, dass die gefundenen Fraktionen der Kompartimente in den durch die Tabelle 8.9.6 gegebenen Grenzen liegen müssen.

Kapitel 9

Aspekte der Validierung

Bei der Darstellung des Falsifikationsprinzips in Abschnitt 4.1 ist darauf hingewiesen worden, dass Validierung im Grunde nur eine negative Schlussfolgerung zulässt. Wenn sich das Modell in einem gegebenen Fall oder Szenario nicht wie erwartet verhält, ist in einer Phase der Modellbildung ein Fehler unterlaufen. Daher kann eine positive Validierung höchstens vorläufig sein.

9.1 Methodik und Grenzen der Validierung

Ausgangspunkt einer Validierung ist die Verhaltensgültigkeit, wie sie in den Abschnitten 6.1.1 und 6.2 eingeführt worden ist. Sie wird in einen qualitativen, quantitativen und kontextuellen Bereich sowie eine Überprüfung der Kalibrierung unterteilt.

Das physiologische Modell ist Teil des EWS-Projektes. Das Modell interagiert mit dessen Steuerung und einem Expertensystem. Auf dieses Gesamtprojekt ist bisher nur am Rande eingegangen worden und auch hier wird auf eine Prüfung der Interaktion der Komponenten verzichtet, da sie innerhalb der Steuerung des EWS organisiert wird. Der Kontext des physiologischen Modells wird in die Gültigkeitsprüfung nur insofern eingehen, als die Größen auch hier im Mittelpunkt stehen, die gemäß dem Modellzweck an das EWS ausgegeben werden.

Quantitativ ist das Modell in einer informatischen Arbeit nur in einem sehr eingeschränkten Sinn zu prüfen. Dazu muss man sich klarmachen, dass es als Prognosemodell eingesetzt wird. Eine quantitative Validierung würde beispielsweise in einer Aussage der Art „*Das Modell prognostiziert die arteriellen Partialdrücke von Kohlendioxid und Sauerstoff in der Klasse X von Patienten, die im Modus BIPAP-ASB beatmet werden, auf Y Prozent genau.*“ kulminieren. Es ist klar, dass derartige Aussagen nur in einer medizinischen Arbeit überprüft werden können.

Es bleibt an dieser Stelle, das Modell aufgrund von Verhaltensphänomenen qualitativ zu überprüfen. Hierzu ist selbstverständlich eine geeignete Implementation notwendig. Im ersten Abschnitt wird gezeigt, wie die im zweiten Teil in Abschnitt 5.2.1 entwickelte generische Struktur für modulare L/R-Modelle in der Atmungsphysiologie eingesetzt wird. Zur Beschreibung wird die im zweiten Teil (Abschnitt 5.1.2) als Hilfsmittel zur Analyse eingeführte Sprache UML verwendet.

Davon ausgehend kann im nächsten Abschnitt das Modell beispielhaft entlang den Daten eines Patienten anhand ausgewählter Phänomene überprüft werden. Gründe für dieses Vorgehen liegen darin, dass sich fiktive (plausible) Szenarien nur schwer unabhängig von Messwerten finden lassen, da das System und mithin das Modell komplex ist. Ferner würden weitere Simulationen keine qualitativ anderen Erkenntnisse liefern.

Im Zuge der Validierung darf mit aller Vorsicht¹ auch auf Strukturgültigkeit geschlossen werden: Da bei der Modellbildung im Sinne einer Whitebox-Modellierung vorgegangen worden ist und die Operationalisierung als verifiziert vorausgesetzt wird, kann man aufgrund von im Modell reproduzierbaren Systemphänomenen auf Strukturgültigkeit schließen.

Die Validierung schließt damit, dass anhand eines ausgesuchten Aspektes der Modellierung das Koordinatensystem der Modellbildung als geeigneter Lebenszyklus der Modellbildung dargestellt wird.

9.2 Die Realisierung in einer generischen Struktur

Aus der Darstellung im zweiten Teil in Abschnitt 5.2.1 ist bekannt, dass grundsätzlich zwischen der Problemstruktur und einem steuernden Experten unterschieden werden muss. In der Klasse `TPMExpert` sind — mit Ausnahme der Simulation — die Algorithmen, die in Abschnitt 8.9 für das physiologische Modell entwickelt worden sind, implementiert. Im Problembereich werden ausgehend von der Klasse `TPMProblemcomponent` in die Modellsteuerung im Sinne von Signalen (`TEnvironment`) und die internen Komponenten `TInternal` unterschieden. Die Modellsteuerung ist in der Klasse `TPMControl` realisiert und besteht aus der Modellierung des Beatmungsmodus BIPAP-ASB, wie er in den Abschnitten 7.1.2 und 8.3 eingeführt und modelliert worden ist. Im „Innern“ des Systems werden die Problemkomponenten entwickelt. Die Darstellung folgt den Teilen Ventilation (vgl. Abschnitte 7.2 und 8.4 bis 8.6), Perfusion (Abschnitt 7.3 und 8.7) und Diffusion (Abschnitt 7.4 und 8.8). Abbildung 9.1 fasst die benutzten Klasse ausgehend von `TPMProblemcomponent` zusammen.

9.2.1 Das Klassenmodell der Ventilation

Innerhalb des Ventilationsteils des physiologischen Modells sind die Trachea, die Kompartimente und die Lunge zu betrachten.

Trachea

Es werden zwei verschiedene Typen von Tracheen eingesetzt. Zunächst ist für das Volumenmodell eine Klasse `TPMTrachea` implementiert worden. Sie realisiert den Fluss zwischen der Trachea und der Systemumwelt. Zusätzlich sind hier das expiratorische PEEP-Ventil, sowie Compliance und Resistance der Trachea integriert. Eine Instanz dieser Klasse wird benutzt, um für den übrigen Teil des Modells festzulegen, welches Volumen das Modell in- bzw. expiriert.

¹Noch steht die Überprüfung des Modells in einer medizinischen Arbeit aus.

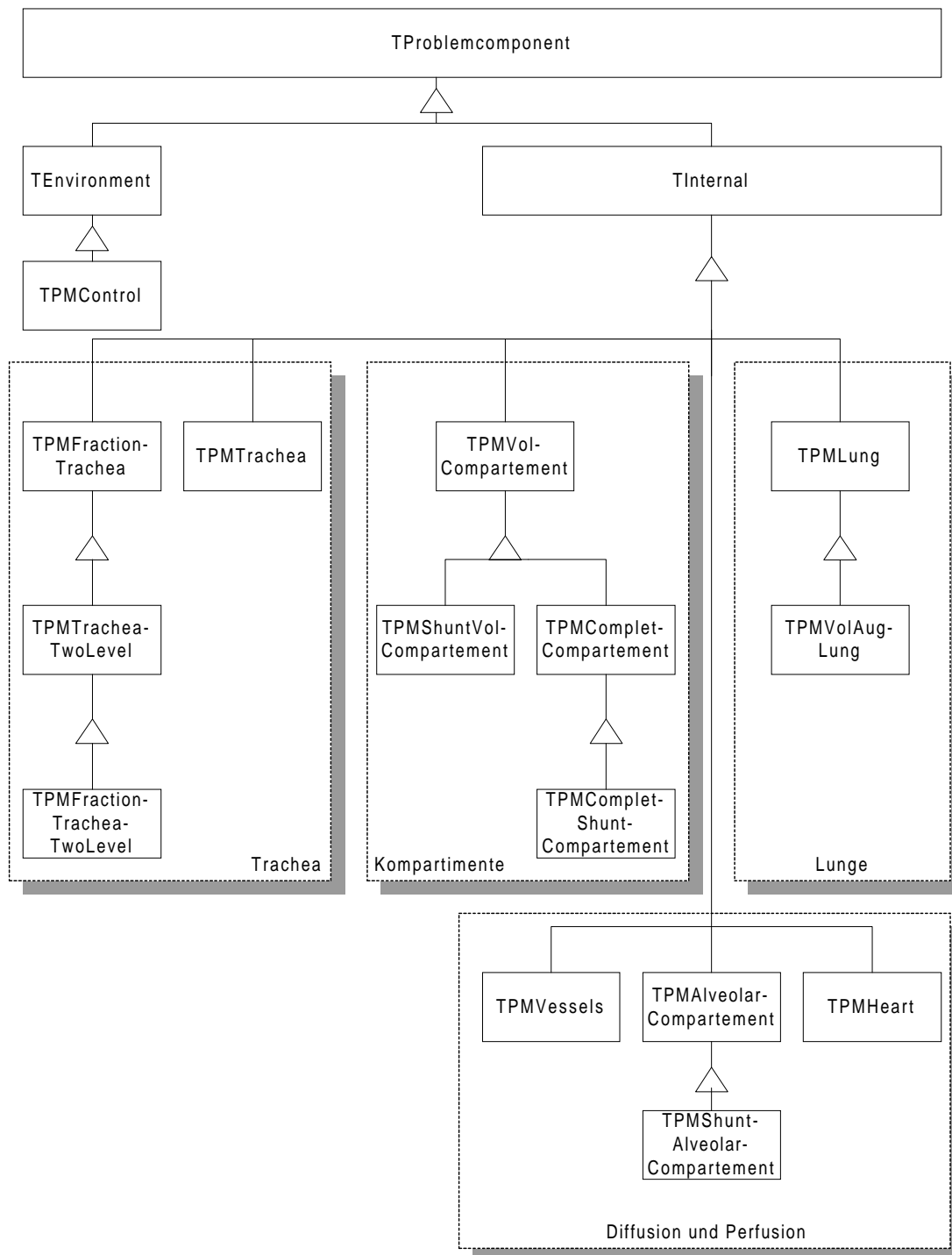


Abbildung 9.1: Die Vererbungsstruktur des physiologischen Modells

Dieser Wert wird in der zweiten Modellierung der Trachea benutzt. Deren Ziel ist, für das obere und untere Ende der Trachea verschiedene Sauerstoff- bzw. Kohlendioxidfraktionen zu erzeugen. Daher sind aus einer abstrakten Klasse `TPMFractionTrachea` die konkreten `TPMTracheaTwoLevel` und `TPMFractionTracheaTwoLevel` abgeleitet worden. Die beiden letzten Klassen realisieren den Volumenfluss durch zwei Trachea-Level (vgl. Abschnitt 8.5.2) und daraus abgeleitet den Teilchenfluss von Sauerstoff und Kohlendioxid, so dass schließlich Fraktionen gebildet werden können.

Die abstrakte Basisklasse ist daher eingeführt worden, damit bei einer möglichen Modellrevision eine weitere Klasse von ihr abgeleitet werden kann, ohne im übrigen Modell etwas ändern zu müssen. Denn als „Schnittstellen“ sind nur das obere und untere Ende von Belang: Wie der Weg dazwischen modelliert wird, bleibt verborgen. In Abbildung 9.1 ist die entsprechende Klassenstruktur links zu sehen.

Die Kompartimente

Bei der Modellierung der Kompartimente sind zwei Punkte zu beachten: Es muss zwischen den belüfteten Kompartimenten und dem nichtbelüfteten Shunt unterschieden werden. Zweitens ist, ähnlich der Modellierung der Fraktionen-Trachea, für die Modellierung der Fraktionen Voraussetzung, dass das Volumen bereits modelliert ist. Daher ist von `TInternal` eine Vererbungsstruktur, wie sie in Abbildung 9.1 in der Mitte zu sehen ist, aufgebaut worden: Ausgangspunkt ist die Klasse `TPMVolCompartment`. Sie implementiert ein Kompartiment mit dem druckgesteuerten Volumenfluss aus der Trachea und den damit verbundenen Parametern Resistance und Compliance. Die abgeleitete Klasse `TPMShuntVolCompartment` überschreibt die Methode zur Flussberechnung derart, dass das Ergebnis stets Null ist. Die vollständigen Kompartimente `TPMCompleatCompartment` enthalten die jeweiligen Mengen von Sauerstoff und Kohlendioxid sowie die Fraktionen. Aufgrund der Vererbung steht ihnen das Gesamtvolumen zur Bildung von Fraktionen zur Verfügung. Wiederum ist für den Shunt eine eigene Klasse abgeleitet worden.

Die Lunge

Aus der Darstellung der Ventilation in Abschnitt 8.4 ergibt sich, dass die Lunge des Modells im Wesentlichen zwei Funktionen erfüllt: Zunächst ist sie ein „Informationslager“ über den gesamten Atmungsapparat: Zeitkonstante, Compliance, Resistance, Thorax-Compliance und Atemzugvolumen sind die hervorragenden Beispiele. Zweitens spielt sie bei der Modellierung der Spontanaktivität eine wichtige Rolle. Es ist ihre Aufgabe, einen Druckgenerator gemäß Gleichung (8.29) zu liefern. Abbildung 9.1 zeigt rechts die geeignete Vererbungsstruktur: Die Klasse `TPMLung` enthält die wesentlichen Parameter und eine virtuelle Methode zur Berechnung des Druckgenerators. Hier ist dieser noch konstant Null und nicht zur Spontanaktivität fähig. Erst mit der vererbten Klasse `TPMVolAugLung` wird der Generator gemäß der obigen Darstellung eingefügt. Dabei ist der Volumengenerator als Methodenzeiger realisiert, so dass im Prinzip auch andere zeitabhängigen Funktionen implementiert werden können. Darüber hinaus lässt sich auch ein ganz anderes Konzept in anderen Nachfolgeklassen von `TPMLung` verfolgen, solange diese den dortigen Druckgenerator konkretisieren.

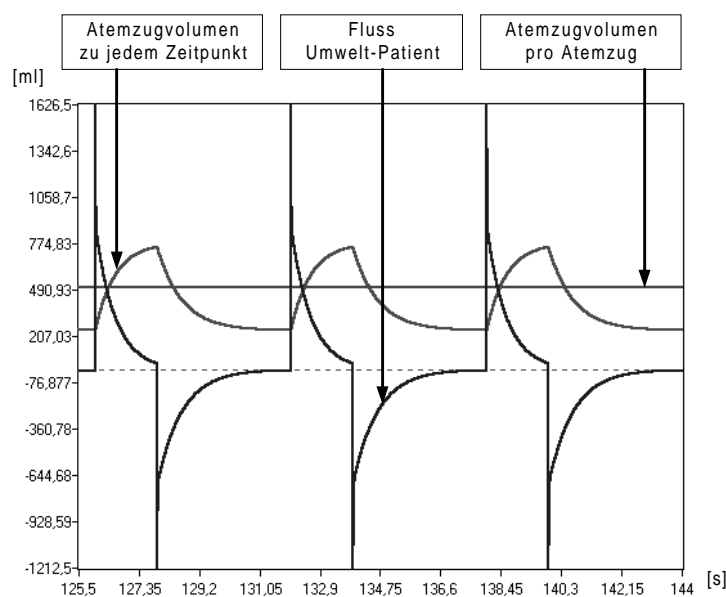


Abbildung 9.3: Das Atemzugvolumen

durch eine paarweise 1-1-Assoziation erreicht. Das *Herz* versorgt den übrigen Körper mit Blut. Es ist auf einen „Gesamt-Blutpool“ verzichtet worden, so dass das gesamte Herz-Zeit-Volumen Q im *Arterienpool* und den *Venenpool* zur Verfügung steht. Diese sind mit dem *Gefäß* assoziiert, in dem der zweite Diffusionsprozess modelliert wird.

9.3 Validierung durch ein Fallbeispiel

Anhand eines kontrolliert beatmeten Patienten werden die grundsätzlichen Verhaltensphänomene des physiologischen Modells dargestellt.

9.3.1 Ventilation

Für die Ventilation und das Rekonstruieren des gemessenen Atemminuten-Volumens von 5,2 Litern pro Minute sind die Beatmungsparameter aus der folgenden Tabelle verantwortlich. Die Compliance des Patienten beträgt 30,2 ml/mbar.

| f_{Masch} [1/min] | $T_{\text{I-Masch}}$ [s] | P_{Insp} [mbar] | PEEP [mbar] |
|----------------------------|--------------------------|--------------------------|-------------|
| 10 | 1,89 | 23 | 4 |

Fluss-Volumen-Diagramm

Aus einer Simulation mit diesen Einstellungen sind drei Atemzüge ausgewählt worden. Man erkennt in Abbildung 9.3, wie zu Beginn der Inspiration der Fluss sprunghaft ansteigt, um bis zum Ende der Inspiration fast auf Null zu sinken. Mit Beginn der Expiration führt ein entsprechender negativer Fluss zum Abatmen der Gase. Die horizontale Linie stellt mit 511 ml das Atemzugvolumen dar. Es ergibt sich aus der Atemzugvolumen-Kurve, indem die Differenz zum Ende der Inspiration und zum Ende der Expiration gebildet wird.

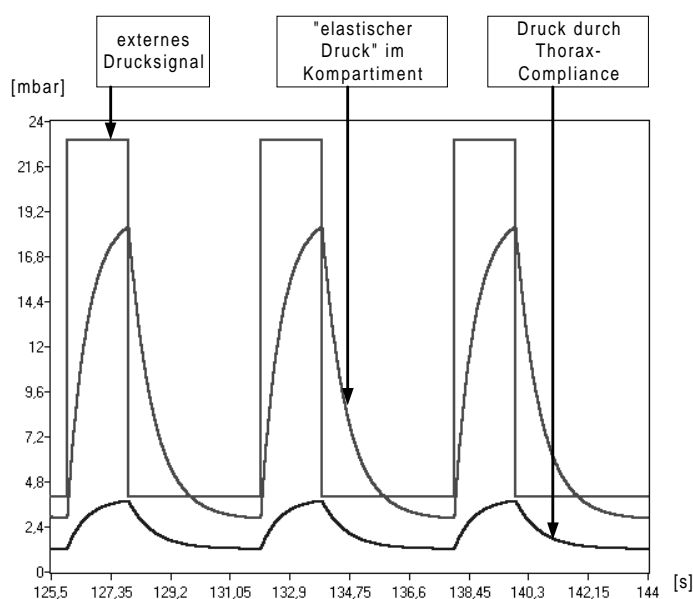


Abbildung 9.4: Drucksignal und Druckkurven

Druckkurven ohne Augmentierung

Der Fluss im Diagramm 9.3 wird durch ein externes Drucksignal und den Druck in der Lunge gesteuert. Abbildung 9.4 zeigt die Zusammenhänge. Der Druck in der Lunge setzt sich aus einem „elastischen“ Anteil aufgrund der Compliance im Kompartiment und einem Anteil aufgrund der Thorax-Compliance zusammen. Bis zum Ende der Inspiration steigt die Summe der beiden Drücke fast auf das Niveau des P_{Insp} . Die Inspiration ist nicht „perfekt“, d.h. es kommt nicht zum völligen Druckausgleich. Daher fällt der Fluss im Diagramm 9.3 zum Ende der Inspiration nicht ganz auf Null.

Druckkurven mit Augmentierung

Unter den gleichen Voraussetzungen wie in den vorherigen Simulationen wird eine Augmentierung von 2000 ml bei einer Thorax-Compliance von 200 ml/mbar eingestellt. Dies entspricht zusammen einer Augmentierung von 10 mbar. Man erkennt in Abbildung 9.5, dass der Druck durch die Thorax-Compliance unverändert bleibt. Durch die Augmentierung steigt der Druck in den Kompartimenten über den Druck in der Umwelt. Dennoch fließt Luft in die Lunge, so dass das Atemminuten-Volumen durch die Augmentierung von 5,11 auf 7,71 steigt. Umgekehrt kann bei einem vorgegebenen Minuten-Volumen des Patienten der Algorithmus `Augmentierung_Berechnen` per Iteration die Augmentierung ermitteln, die zum gewünschten Volumen im Modell führt.

9.3.2 Diffusion und Perfusion

Nach der Einstellung der Atmungsmechanik erfolgt die Untersuchung der Blutgaswerte im physiologischen Modell. Hierzu wird zunächst das Modell mit der Methode `Einschwingen` in einen stabilen Zustand überführt.

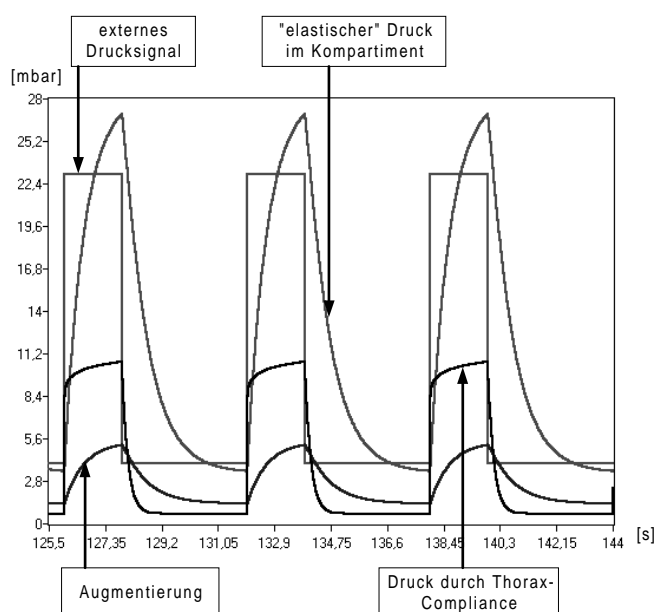


Abbildung 9.5: Druckkurven mit Augmentierung

Arterielle Partialdrücke

Im Zuge des Einschwingens werden in diesem Beispiel zunächst die Kalibrierungsparameter der Shunt- und Totraumfraktion PZ_S , PZ_T und daraus die Normal-Fraktion PZ_N mit Hilfe des neuronalen Netzes gesetzt. Die anschließende Simulation ergibt folgende Resultate:

| PZ_S [%/100] | PZ_T [%/100] | PZ_N [%/100] | P_{aO_2} [mmHG] | P_{aCO_2} [mmHG] |
|----------------|----------------|----------------|-------------------|--------------------|
| 0,061 | 0,222 | 0,717 | 158,7 | 34,2 |

Zu den gemessenen Blutgasen von einem P_{aO_2} von 153 mmHG und P_{aCO_2} von 37,1 mmHG bei einem Atemminuten-Volumen von 5,2 l/min, bei einer F_{IO_2} von 0,4 und einem Herz-Zeit-Volumen von 4200 ml/min liefert das neuronale Netz die Verteilung der Kompartimente. Die zugehörige Simulation während des Einschwingens ergibt die in obiger Tabelle angegebenen simulierten Blutgaswerte. Die Abbildungen 9.6 und 9.7 zeigen, wie diese zustande gekommen sind. Hier wird exemplarisch deutlich, weshalb im physiologischen Modell drei verschiedene Zeitskalen zu verwenden sind. Weder die Werte zu jedem Zeitpunkt noch diejenigen auf Atemzugsebene liefern ein stabiles Bild vom Patienten. Erst wenn die Werte auf Minutenebene aggregiert sind, können die Ergebnisse interpretiert werden.

Die beim Einschwingen gefundene Näherung für P_{aO_2} ist zunächst mit einem Fehler von 3,7% und einer Toleranzschwelle von 5% hinreichend genau. Der Fehler beim Kohlendioxidpartialdruck liegt mit 7,8% über der Toleranzschwelle von 3%. Der Iterationsalgorithmus aus Abschnitt 8.9.6 senkt gemäß der formulierten Regel den Anteil des Normal-Kompartimentes um 5%, so dass die alveolare Ventilation verringert, damit das Abatmen des Kohlendioxids vermindert und so der arterielle Kohlendioxidpartialdruck erhöht wird. Es ergeben sich die folgende Werte:

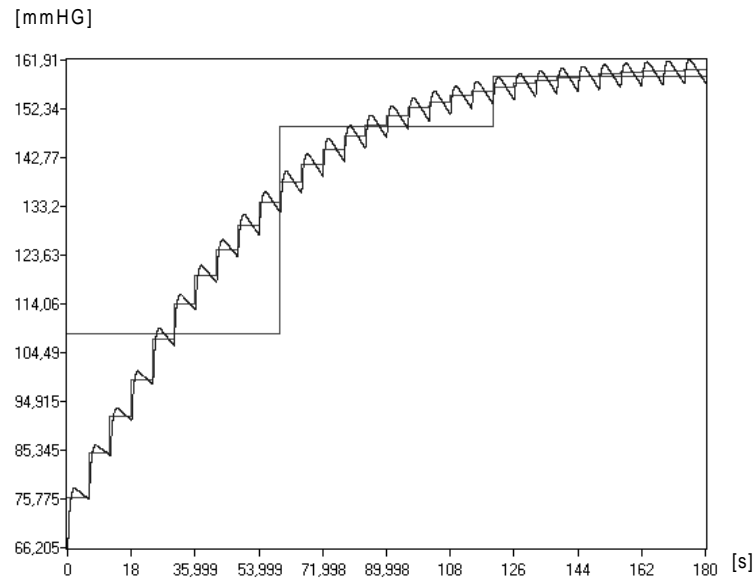


Abbildung 9.6: Arterieller Sauerstoffpartialdruck auf den verschiedenen Zeitskalen

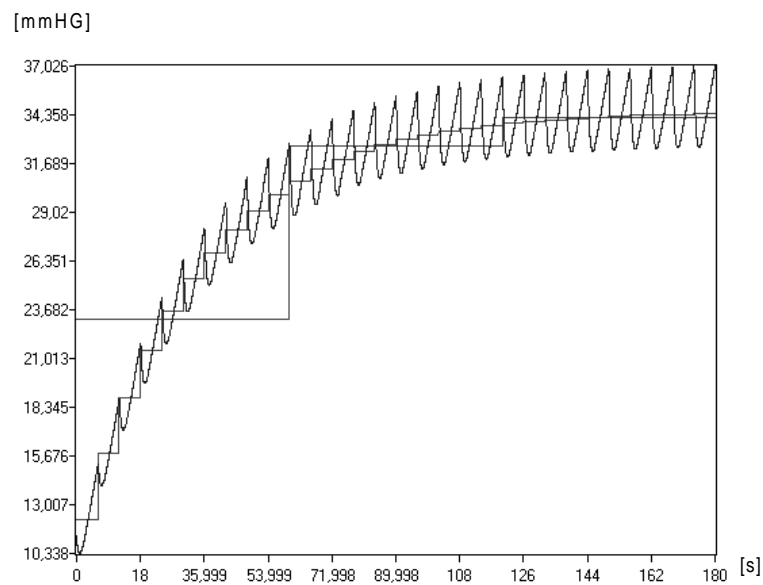


Abbildung 9.7: Arterieller Kohlendioxidpartialdruck auf den verschiedenen Zeitskalen

| PZ _S [%/100] | PZ _T [%/100] | PZ _N [%/100] | P _a O ₂ [mmHG] | P _a CO ₂ [mmHG] |
|-------------------------|-------------------------|-------------------------|--------------------------------------|---------------------------------------|
| 0,061 | 0,272 | 0,667 | 152,7 | 37,2 |

Dieser eine Simulationsschritt genügt, um die Blutgase in den gewünschten Bereich zu bringen. Man erkennt hier, dass die Änderung der alveolaren Ventilation auch Einfluss auf den arteriellen Sauerstoffpartialdruck hat. In diesem Fall geht die Änderung mit einer Verringerung des P_aO₂ in die gewünschte Richtung einher, und keine weitere Anpassung des P_aO₂ ist nötig. Diese würde man ggf. durch eine Verringerung Shuntfraktion auf Kosten der Totraumfraktion bei konstantem Anteil des Normal-Kompartimentes erreichen. Dabei bleibt der P_aCO₂ im Wesentlichen unverändert, wie die folgende Betrachtung der möglichen Extremsituationen zeigt. Aufgrund des Konzepts des Algorithmus, die Blutgase über die Anteile der Kompartimente zu regeln, ergeben sich mögliche Extremsituationen, die in der folgenden Tabelle dargestellt sind:

| PZ _S [%/100] | PZ _T [%/100] | PZ _N [%/100] | P _a O ₂ [mmHG] | P _a CO ₂ [mmHG] |
|-------------------------|-------------------------|-------------------------|--------------------------------------|---------------------------------------|
| 0,061 | 0,01 | 0,929 | 185,0 | 28,1 |
| 0,061 | 0,889 | 0,005 | 13,7 | 110 |
| 0,01 | 0,323 | 0,667 | 223,0 | 37,5 |
| 0,323 | 0,01 | 0,667 | 50,9 | 30,7 |

Es ergibt sich zunächst, dass der Algorithmus bei gegebenem Shunt-Anteil von 6,1% keinen arteriellen Kohlendioxidpartialdruck außerhalb des Intervalls [28, 1; 110] mmHG erreichen kann. Dabei ist der Partialdruck von 110 mmHG ein Extremwert der aufgrund der Implementation nicht überschritten wird.

Aus den beiden letzten Zeilen erkennt man, dass bei einem Normal-Anteil von 66,7% nur ein arterieller Sauerstoffpartialdruck im Intervall [50, 9; 223] mmHG erreicht werden kann. Ferner ist zu sehen, dass die extreme Variation des Shunt-Anteils nur einen geringen Einfluss auf den Kohlendioxidpartialdruck hat. Er schwankt zwischen 30,7 und 37,5 mmHG.

Schließlich wird beispielhaft gezeigt, wie man das Modell zur Prognose einsetzen würde: In einer weiteren Simulation ist der P_{Insp} um 5 auf 28 mbar erhöht worden. Dies führt zu einer Erhöhung der Ventilation auf 6,43 l/min. Damit prognostiziert das Modell eine Erhöhung des P_aO₂ auf 161,2 mmHG und eine Verringerung des P_aCO₂ auf 28,5 mmHG. Tendenziell gehen die Änderungen in die erwarteten Richtungen: Eine erhöhte Ventilation führt zu einem erhöhten Partialdruck des Sauerstoff und zum vermehrten Abatmen des CO₂, so dass dessen Partialdruck sinkt. Somit wäre eine Erhöhung des Inspirationsdrucks keine gute Entscheidung, da dadurch tendenziell Eigenaktivität des Patienten verhindert wird.

Normventilation

Der Algorithmus zur Berechnung der Normventilation über drei Simulationen und einen quadratischen Zusammenhang von Atemminuten-Volumen und arteriellem Kohlendioxidpartialdruck liefert eine Normventilation von 4822 ml/min. Dieses Ergebnis lässt sich nur modellintern überprüfen: Durch systematisches Variieren des Inspirationsdrucks erhält man folgende Ergebnisse:

| P_{Insp} [mbar] | AMV [$\frac{\text{ml}}{\text{min}}$] | P_{aCO_2} [mmHG] |
|--------------------------|--|---------------------------|
| 23 | 5093 | 36,6 |
| 20 | 4285 | 45,2 |
| 21,5 | 4684 | 41,3 |
| 22 | 4817 | 39,8 |

Es zeigt sich, dass eine Parabel zur Bestimmung der Normventilation zumindest im Modell ein geeignetes Modell liefert. Mit einem Inspirationsdruck von knapp 22 mbar wird man eine Ventilation von 4820 ml/min erreichen, was zu einem P_{aCO_2} von 40 mmHG führt.

Trachea und Expirogramme

Der Test, ob die Modellierung der Trachea mit Hilfe zweier Level sinnvoll ist, erfolgt in zwei Schritten. Zunächst zeigt Abbildung 9.8, dass der grundsätzliche Verlauf der Kurve mit demjenigen aus Abbildung 7.6 übereinstimmt:

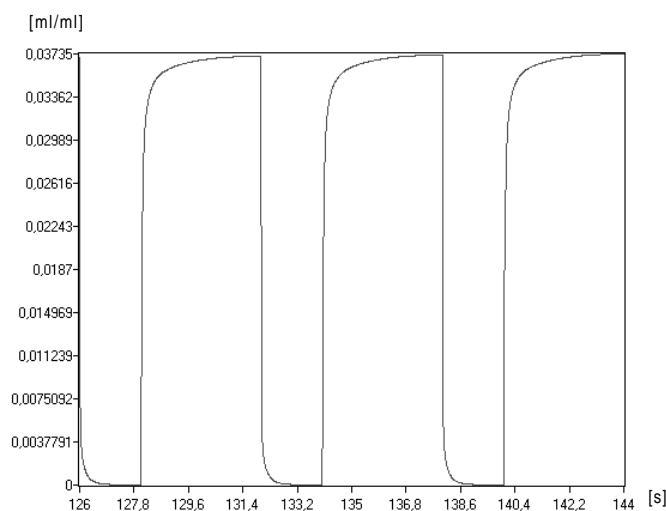


Abbildung 9.8: Ein Expirogramm aus Modelldaten

Aufgrund der groben Auflösung kann man allerdings nicht entscheiden, ob auch im Modell zu Anfang der Inspiration das Volumen des seriellen Totraums eingeatmet wird. Hierzu werden in Abbildung 9.9 und 9.10 der Ausschnitt auf einen Wechsel in der Atemphase beschränkt. In der Abbildung 9.9 ist der bisherige Fall eines seriellen Totraums von 100 ml zu sehen. Dagegen zeigt Abbildung 9.10 die Ergebnisse einer weiteren Simulation mit einem Totraum von 300 ml. Es wird deutlich, dass auch im Modell über das Inspirationsende zum Zeitpunkt 127,89 hinaus zunächst noch die Kohlendioxidfraktion fast Null ist, und dass der serielle Totraum Einfluss auf das Modellverhalten hat: Je größer der Wert des seriellen Totraums ist, desto später beginnt die expirierte Kohlendioxidfraktion zu steigen.

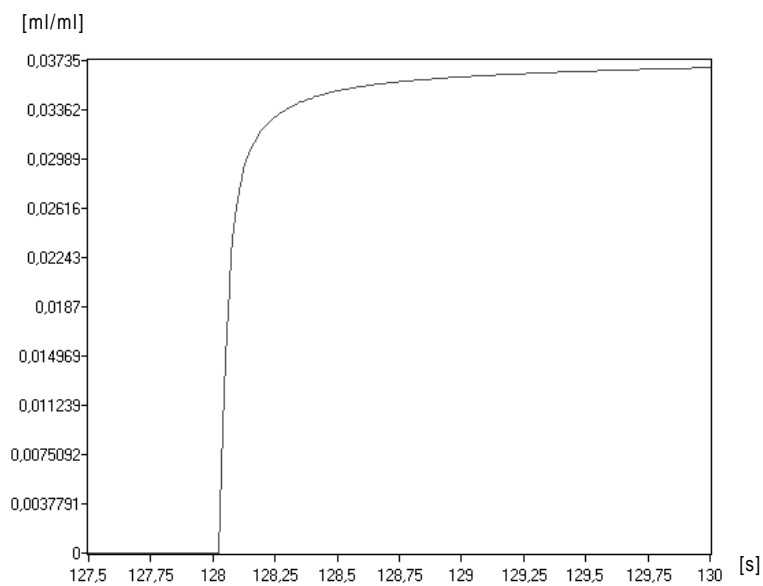


Abbildung 9.9: Expirogramme beim Wechsel der Atemphase mit einem $V_{d_s} = 100$ ml

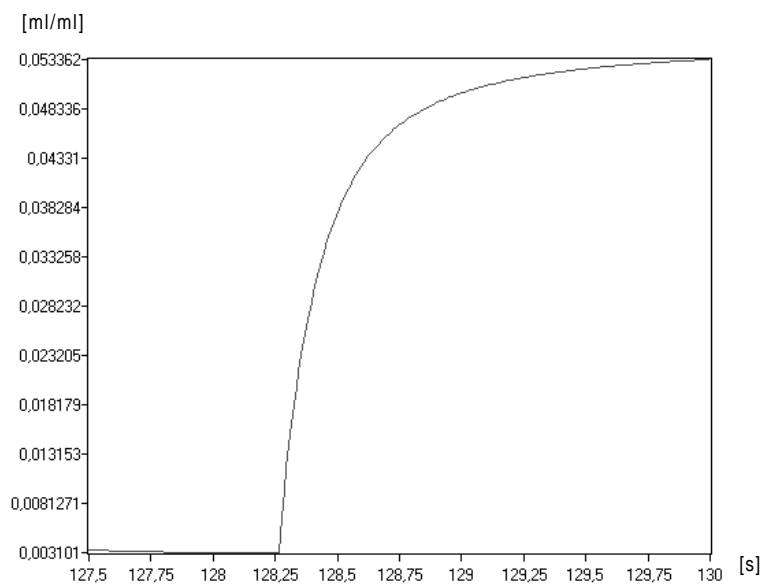


Abbildung 9.10: Expirogramme beim Wechsel der Atemphase mit einem $V_{d_s} = 300$ ml

9.4 Validierung des Koordinatensystems als Lebenszyklus

Am Beispiel der Modellierung der Spontanaktivität wird der Nutzen des Koordinatensystems als Lebenszyklus der Modellbildung dokumentiert. In diesem Fall konkretisiert sich das Koordinatensystem wie in Abbildung 5.1. Auf Systemseite hat der Experte ein informales Systemmodell vom Systemobjekt. Die Analyse durch Wissensakquisition führt beim Modellingenieur zu einem formalen Systemmodell, das operationalisiert und schließlich mit dem Experten überprüft werden muss. Die im Kapitel 6 gefundenen Validitätsarten stellen sich folgendermaßen dar:

1. **Experimentvalidität:** Hier ist die Frage zu beantworten, ob der Experte ein passendes Systemmodell kennt oder er für das aktuelle Problem und das zugehörige Objekt die richtige Modellkomponente benutzt.
2. **Strukturvalidität:** Bei fehlender Strukturvalidität liegt ein Kommunikationsproblem vor. Zwar will der Experte die „richtige“ Modellkomponente beschreiben. Dies scheitert jedoch daran, dass ihm dies nicht gelingt oder der Modellingenieur es falsch versteht.
3. **Operationalisierungsvalidität:** Neben der obligatorischen Frage nach einer verifizierten Operationalisierung kann ein Problem dadurch entstehen, dass der Modellingenieur durch einen nur scheinbar richtigen Prototypen die Modellbildung in eine falsche Richtung lenkt.
4. **Verhaltensvalidität:** Wie im Grundkonzept ist die Verhaltensgültigkeit Ausgangspunkt der Überprüfungen. Im Modellierungsprozess wird man hier im Dialog zwischen Experten und Modellingenieur auch die im Sinne einer Anforderungsdefinition festzulegenden Kriterien überprüfen bzw. erst bestimmen können.

Im Verlauf der Modellierung der Spontanaktivität haben sich diese Zusammenhänge folgendermaßen geäußert. Hierzu wird zunächst der Prozess der Modellierung hier nochmals rekapituliert:

Ausgangspunkt der Modellierung war ein völlig kontrolliert beatmeter Patient. In diesem Stadium wurde der Druck in den Kompartimenten wie in Gleichung (8.5) aus Volumen und Compliance bestimmt. Als im nächsten Schritt die Spontanaktivität modelliert werden sollte, wurde diese derart festgelegt, dass es sich dabei um den „Druck handelt, den der Patient zusätzlich zum Kompartimentsdruck aufbringt.“ Damit erschien eine Modellierung wie in Gleichung (8.29) mit einem zusätzlichen Druck $P_{G_{Aug}}(t)$ sinnvoll. Ferner hat sich in der Medizin die Sprechweise von „einer Spontanaktivität von z.B. 5 mbar“ eingebürgert. Es erschien also angebracht, die Spontanaktivität mit $P_{Aug}(t) = \text{const}$ für $0 \leq t < T_I$ zu modellieren. Als in der Folge negative Kompartiment-Drücke auftraten, folgerten wir, dass der zusätzliche Druck, den der Patient aufbringt, zum Ende der Inspiration gegen Null gehen muss. Damit begann eine lange Suche nach geeigneten Funktionen, die das Gewünschte leisten sollten. Sie führten auch in immer mehr Fällen zu sinnvollen Ergebnissen, warfen aber immer wieder das Problem von negativen Drücken oder nicht erreichtem Zielvolumen auf.

Unterdessen hatten der Experte und der Modellingenieur als gemeinsames Bild für die Modellierung der Spontanaktivität eine (Unterdruck-)Pumpe eingeführt, die nach unten gezogen wird, damit Unterdruck in der Lunge erzeugt und insgesamt dem Zwerchfell und dem Thorax entspricht. Der entscheidende Schritt gelang, als die eigentlich offensichtlich Tatsache klar wurde, dass man am meisten Kraft bei der Pumpe aufbringen muss, wenn sie ganz nach unten gezogen ist und mit Luft gefüllt ist, um diese im System zu halten. Wenn man die Pumpe loslässt, entweicht die Luft. Bis dahin hatten alle Modellierungen darauf aufgebaut, dass am Ende der Inspiration, wenn die Lunge mit Luft gefüllt ist, die Eigenaktivität Null ist, während man jedoch die meiste Kraft (und damit am meisten Druck) aufbringen muss, um den Atem in der Lunge zu halten. In physiologischer Terminologie heißt das, dass im physiologischen Modell der intrapleurale Druck modelliert worden war, obwohl nicht dieser sondern der intrapulmonale Druck zum Druckgradienten beiträgt (vgl. Abschnitt 7.2.1).

Dieser Vorgang ist insofern lehrreich, als bei der Modellierung etliche typische Fehler gemacht worden sind. Erstens hat der Modellingenieur — durch die frühe Vorgabe der Formel — das Modell mit dem Prototypen einer konstanten Eigenaktivität in eine falsche Richtung geführt (Operationalisierungsvalidität!). Zweitens fehlte Strukturvalidität, da das Wissen über das System mit der Unterscheidung zwischen intrapleuralem und intrapulmonalem Druck nicht adäquat kommuniziert worden ist. Es sollte schließlich nicht der zusätzliche, sondern der gesamte resultierende Druck modelliert werden. Damit hat der Experte die falsche Komponente seines Systemmodell verwendet hat und es fehlte Experimentvalidität.

Diese Modellierung hat unter anderem deshalb so viele Schwierigkeiten bereitet, weil die Überprüfung der Verhaltensvalidität am Modellverhalten in vielen Fällen zu qualitativ und meist auch quantitativ richtigen Ergebnissen geführt hat. Frühere Falsifikationen hätten vielleicht eine genauere Modellierung gefördert. An diesem Beispiel sollte klar geworden sein, dass man im Zuge der Modellbildung stets das Koordinatensystem auf die aktuelle Situation anpassen und damit sich im Falle von falschem Modellverhalten nach den möglichen Fehlerquellen fragen sollte.

Schlussbetrachtung

Ergebnisse

Ausgangspunkt der Arbeit und Gegenstand der abschließenden Überprüfungen im dritten Teil ist das physiologische Modell im Rahmen des Evita-Weaning-Systems EWS. Daher stellt zunächst der gesamte dritte Teil mit der Entwicklung des physiologischen Modells ein wichtiges Ergebnis dar. Wie in der Einleitung dargelegt worden ist, sind die Ideen dieser Arbeit in einem bottom-up Prozess entstanden. Insofern führen die Darstellungen in den ersten beiden Teilen auf das physiologische Modell hin.

Das EWS regelt den Prozess der Entwöhnung vom komplett kontrolliert beatmeten bis hin zum völlig spontan atmenden Patienten. Im Zusammenhang der Entwöhnung wird das Modell zur Entscheidungshilfe im Expertensystem benutzt. Unter Verwendung der vom Modell simulierten arteriellen Partialdrücke des Kohlendioxids und Sauerstoffs und der Normventilation werden vom Expertensystem neue Beatmungsparameter vorgeschlagen.

Zu diesem Zweck ist das physiologische Modell derart aufgebaut worden, dass es von einer Druckkurve, die den Beatmungsmodus BIPAP-ASB darstellt, gesteuert wird und als Ausgaben die genannten Blutgaswerte liefert. Hierzu sind die Ventilation, die Perfusion und die Diffusion in das Modell integriert. Es handelt sich um ein Dreikompartimente-Modell. Dementsprechend sind neben einem „idealen“ Normal-Kompartiment auch der Shunt und der Totraum vorhanden. Ihr Einfluss auf die betrachteten Prozessen wird maßgeblich von der relativen Größe der Kompartimente zur gesamten Lunge, den sog. Shunt-, Totraum- und Normal-Fractionen bestimmt. Diese Fractionen werden auch in den übrigen Teilen des Modells verwendet. Sie regulieren die Verteilung des Herz-Zeit-Volumens auf die Kompartimente. Somit haben sie auch maßgeblichen Anteil auf die modellierten Diffusionsprozesse in der Lunge. In diesem Teilmodell sind physiologische Standardverfahren worden: Die Kompartimente werden mit Blut versorgt, der Gasaustausch findet ohne Diffusionswiderstand zwischen Alveole und Kapillar statt und anschließend mischen sie Anteile am Herz-Zeit-Volumen zum arteriellen Blut.

Da das physiologische Modell den gesamten Prozess der Entwöhnung unterstützen soll, ist es notwendig auch die Spontanaktivität des Patienten zu abbilden. Dies ist durch einen zeitabhängigen Druckgenerator in der Lunge geschehen, so dass das Modell stets von zwei Signalen gesteuert wird. In den Extremfällen von völlig kontrollierter und völlig spontaner Atmung ist das jeweils andere Drucksignal konstant. In den Mischformen arbeiten die Signale synchron. Es wird jeder Atemzug durch das gleiche Maß an Eigenaktivität unterstützt.

Schließlich ist im Modell eine Modellierung der Trachea enthalten, die es ermöglicht, Expirogramme zu reproduzieren. Insbesondere kann der Effekt nachgebildet werden, dass bei der Expiration zunächst „frische“ Luft aus der Umwelt abgeatmet wird.

Insgesamt ist keine Größe im physiologischen Modell enthalten, die im Sinne einer Blackbox-Modellierung nicht physiologisch interpretierbar wäre.

Welche allgemeinen Konzepte der Modellbildung zur Operationalisierung — nicht nur im Hinblick auf das physiologische Modell — entwickelt worden sind, wird im zweiten Teil der Arbeit beschrieben. Das physiologische Modell ist zunächst in einem funktionalen Prototypen realisiert worden. So konnte der Aufbau des Modells und die Interaktion mit den übrigen Komponenten wie dem Expertensystem vorangetrieben werden. Anschließend ist eine generische Struktur zur Simulation von modularen Level-Raten-Modellen entwickelt worden, die im Zuge des Projektes konkret für die Anforderungen des physiologischen Modells implementiert worden ist. Damit ist es mit begrenztem Aufwand möglich, das Modell um weitere Komponenten zu erweitern, oder ähnliche Modelle in anderen Zusammenhängen zu entwickeln.

Des Weiteren sind für das physiologische Modell Algorithmen für dessen Kalibrierung entwickelt worden. Im Fall der Kalibrierung der Blutgase wird hier ein neuronales Netz verwendet, das als Ausgabe die Shunt- und die Totraumfraktion liefert. Das Netz ist mit Daten, die durch das physiologische Modell selbst generiert worden sind, trainiert worden. Dies ist die Anwendung eines in dieser Arbeit entwickelten Konzepts zur Kalibrierung, das auf Data Mining von Modelldaten beruht.

Bevor allgemeine Konzepte der Modellbildung und die spezielle Anwendung in der Atmungsphysiologie entwickelt werden können, ist eine Analysephase notwendig. Dies leistet der erste Teil der Arbeit. Ausgehend von einem formalen Zugang über zeitorientierte Systemdefinitionen werden die im weiteren Verlauf notwendigen Fakten über Modelle, Modellentwicklung und Simulation genannt. Im dritten Kapitel wird ein graphentheoretischer Zugang zu Level-Raten-Modellen gegeben. Hier werden die Begriffe aus dem ersten Kapitel über dynamische Systeme aufgegriffen und in einen zur Simulation tauglichen Modellbegriff übertragen. Er ist im zweiten Teil als generische Struktur implementiert worden.

Somit gliedert sich die Arbeit in drei Teile:

1. Die Analyse der Grundlagen,
2. Der Prozess der Operationalisierung,
3. Die Validierung am Beispiel der Atmungsphysiologie.

Dieser Aufbau stellt somit selbst ein Beleg für die Gültigkeit des ebenfalls in dieser Arbeit entwickelten Lebenszyklus dar. Modellbildung wird von einem doppelten Dualismus bestimmt: Erstens muss zwischen dem System und dem Modell und zweitens bei beiden wieder zwischen einer abstrakten und einer konkreten Ebene unterschieden werden. Dieser Zusammenhang kann in einem Koordinatensystem dargestellt werden, woraus sich

die Arbeitsschritte experimentieren, analysieren, operationalisieren und validieren ableiten lassen. Das Koordinatensystem wird im ersten Kapitel des zweiten Teils aus den gängigen Zyklen der Modellbildung und des Software Engineerings, wie sie im ersten Teil der Arbeit vorgestellt worden sind, entwickelt. Darüber hinaus sind im Verlauf der Arbeit noch Zyklen aus anderen Bereichen der Informatik wie der Wissensakquise und dem Data Mining aufgetaucht. Diese lassen sich im Rahmen des Koordinatensystems der Modellbildung als dessen Spezialisierungen interpretieren. Damit liefert das Koordinatensystem eine allgemeine Schablone für informatische Modellbildung und es können die sich im Koordinatensystem ergebenden Arten von Validität auf verschiedene Problembereiche übertragen werden. Kapitel Sechs zeigt insbesondere, dass die gängigen Begriffe der Validität in das Koordinatensystem integriert werden können. Somit geben die verschiedenen Arten der Validität Hinweise auf mögliche Fehlerquellen im Zuge einer Modellbildung.

Ausblick

Die Arbeit liefert Ergebnisse auf drei Ebenen. Mit dem Koordinatensystem der Modellbildung ist ein Beitrag zur Theoriebildung geleistet. Inhaltlich stellen die beiden ersten Teile der Arbeit praktische Erkenntnisse einer informatischen Modellbildung dar. Schließlich ist der dritte Teil mit dem physiologischen Modell eine eigenständige Anwendung im Bereich der Medizin-Informatik.

Reizvoll für die Anwendung des Koordinatensystems ist es, den dadurch induzierten Lebenszyklus in anderen Projekten anzuwenden, und so weitere Anwendungsfelder zu erschließen, aber auch dessen Grenzen aufzuzeigen. Insbesondere ist die Tragfähigkeit der beschriebenen Validitätsbegriffe stets neu zu prüfen. Dieser Test des Koordinatensystems ist nicht nur prospektiv als Handlungsanweisung sondern auch retrospektiv möglich. Es könnte nämlich ein bereits abgeschlossenes Projekt mit diesem Raster untersucht werden.

Im Bereich der praktischen Modellbildung können vor allem die generische Struktur für modulare L/R-Modelle und das Konzept der Kalibrierung durch Data Mining mit Modelldaten Gegenstand weiterer Forschungen sein. Insbesondere wäre zu überlegen, wie die Interaktion einer steuernden Klasse eines Modells, wie sie mit `TPMExpert` für das physiologische Modell eingeführt worden ist, generisch gestaltet werden könnte.

Die Idee, ein Modell durch Data Mining auf Modelldaten zu kalibrieren, ist bisher nur prototypisch für die konkrete Anwendung des physiologischen Modells implementiert. Es ist zu überprüfen, wie dies flexibler und insbesondere unabhängig vom gegebenen Modell realisiert werden kann. Hierzu müsste zunächst ein Werkzeug entworfen werden, dem man ein Modell in einem festzulegenden Format übergibt. Parameter dieses Modells müssten dann variiert und die Ergebnisse gespeichert werden. Dies ist aufwändig und könnte ggf. auch verteilt realisiert werden. Nach einer Datenaufbereitung kann der eigentliche Data Mining-Schritt erfolgen und das Ergebnis auf ein System angewandt werden. Dies sind Tätigkeiten auf zwei Ebenen: Erstens ist ein Programm zu entwerfen, das den Data Mining-Zyklus aus Abbildung 5.14 für unterschiedliche Modelle automatisiert, und zweitens ist die Anwendbarkeit des Konzeptes auf andere Modelle zu prüfen.

Einen Ausblick auf weitere Anwendungen der Modellierung der Atmungsphysiologie kann in dieser informatischen Arbeit nur sehr bedingt erfolgen. Aber es sind grundsätzlich zwei verschiedene Richtungen denkbar: Erstens könnte überprüft werden, wo und gegebenenfalls wie der Aufbau des Modells ergänzt und verbessert werden könnte. Ideen hierfür sind, ein Modellierung des Gehirns einzuführen, das die Eigenaktivität steuert, oder mehr als drei Kompartimente zu benutzen. Zweitens könnten für das physiologische Modell weitere Anwendungsfelder erschlossen werden. Erste Versuche, es als Unterrichtsprogramm in Fortbildungen einzusetzen, haben schon stattgefunden. Außerdem könnte es als Monitoringsystem von Patienten genutzt werden.

Anhang A

Nummerische Integration und Runge-Kutta-Verfahren

A.1 Problemstellung

Wir betrachten ein System von Differenzialgleichungen: Gegeben sei ein System von N Differenzialgleichungen

$$\frac{dy_i(t)}{dt} = f_i(t, y_1, \dots, y_N) \quad i = 1, \dots, N, \quad (\text{A.1})$$

mit den Anfangswerten $y_i(t_0)$ für $i = 1, \dots, N$. Gesucht sind für $i = 1, \dots, N$ die Werte $y_i(t^*)$ der Funktionen y_i zu einem Zeitpunkt t^* . Die Idee des Verfahrens besteht darin, die Gleichungen (A.1) zu diskretisieren, indem dy und dt durch Δy und Δt ersetzt werden. Multipliziert man noch auf beiden Seiten mit Δt , ergibt sich die diskrete Fassung des Systems von Differenzialgleichungen.

$$\Delta y_i = \Delta t * f_i(t, y_1, \dots, y_N), \quad i = 1, \dots, N. \quad (\text{A.2})$$

A.2 Cauchy-Euler-Integration und Level-Raten-Modelle

Ausgehend von Gleichung (A.2) erhält man sofort die Formel für die CAUCHY-EULER-INTEGRATION. Im Fall einer einzigen diskretisierten Differenzialgleichung erhält man

$$\Delta y = \Delta t * f(t, y) \quad y(t_0) = y_0 \quad (\text{A.3})$$

Dies führt auf eine iterative Berechnungsvorschrift:

$$y_{n+1} = y_n + \Delta t * f(t_n, y_n), n \in \mathbb{N}_0. \quad (\text{A.4})$$

Gleichungen dieser Form sind formal mit den Levelgleichungen (3.3) aus Kapitel 3 identisch. Dabei nimmt die Differenz von Zu- und Abflüssen in Level-Raten-Modellen die Rolle der Ableitungsfunktion f ein.

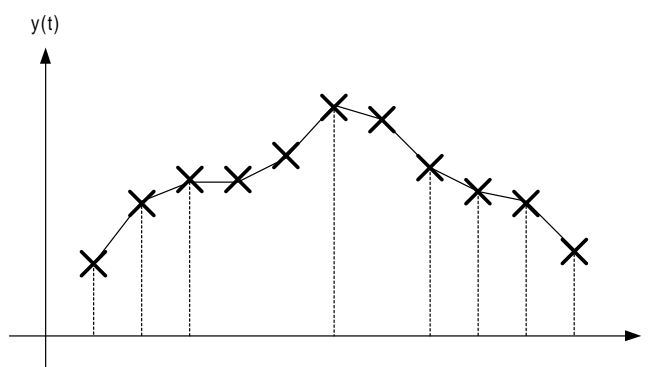


Abbildung A.1: Diskretisierung im Cauchy-Euler-Verfahren

In Abbildung A.1 wird das Vorgehen bei der Cauchy-Euler-Integration nochmals veranschaulicht: Die gesuchte Funktion y wird durch einen Polygonzug angenähert. Zur Berechnung der Lösung zum Zeitpunkt $t + \Delta t$ wird lediglich die Information zum Zeitpunkt t verwendet. Der Verlauf der Funktion dazwischen bleibt unberücksichtigt. Schon diese Bemerkung deutet an, dass das Verfahren nicht übermäßig genau ist. Es ist bekanntlich ein Verfahren erster Ordnung mit einem Fehler von $O(\Delta t^2)$.

A.3 Runge-Kutta-Verfahren

Die Klasse der Runge-Kutta-Verfahren kann wesentlich bessere Lösungen liefern.

A.3.1 Die Idee des Verfahrens

Ausgangspunkt ist der Gedanke, zur Lösung auch Information aus dem gesamten Lösungsintervall zu nutzen. Benutzt man beispielsweise die Annäherung an die Lösungsfunktion in der Intervallmitte, so folgt für $n \in \mathbb{N}_0$:

$$k_1 := f(t_n, y_n), \quad (\text{A.5})$$

$$k_2 := f\left(t_n + \frac{1}{2}\Delta t, y_n + \frac{1}{2}\Delta t k_1\right), \quad (\text{A.6})$$

$$y_{n+1} = y_n + \Delta t k_2. \quad (\text{A.7})$$

Dabei wird zur Berechnung von k_2 der Wert der Funktion $y(t_n + \frac{1}{2}\Delta t)$ aus dem Wert von y für t_n als $y_n + \frac{1}{2}\Delta t k_1$ geschätzt. Dieses Verfahren wird als RUNGE-KUTTA-VERFAHREN ZWEITER ORDNUNG bezeichnet. Am weitesten verbreitet ist „das“ Runge-Kutta-Verfahren vierter Ordnung. Es ist ein Spezialfall eines allgemeinen Schemas.

A.3.2 Das allgemeine Runge-Kutta-Verfahren

Als Runge-Kutta-Verfahren wird eine ganze Klasse von Integrationsverfahren für gewöhnliche Differenzialgleichungen bezeichnet. Ein RUNGE-KUTTA VERFAHREN S-TER ORDNUNG wird als Schema dargestellt. Dabei sei für $n \in \mathbb{N}_0$:

| | | | | | |
|----------|----------|----------|----------|-------------|-------|
| 0 | | | | | |
| c_2 | a_{21} | | | | |
| c_3 | a_{31} | a_{32} | | | |
| \vdots | \vdots | \vdots | \ddots | | |
| c_s | a_{s1} | a_{s2} | \dots | $a_{s,s-1}$ | |
| | b_1 | b_2 | \dots | b_{s-1} | b_s |

Tabelle A.1: Die Koeffizienten des Runge-Kutta-Verfahrens s-ter Ordnung

$$\begin{aligned}
k_1 &:= f(t_n, y_n), \\
k_2 &= f(t_n + c_2 \Delta t, y_n + \Delta t (a_{21} * k_1)), \\
k_3 &= f(t_n + c_3 \Delta t, y_n + \Delta t (a_{31} * k_1 + a_{32} * k_2)), \\
&\dots \\
k_s &= f(t_n + c_s \Delta t, y_n + \Delta t (a_{s1} * k_1 + a_{s2} * k_2 + \dots + a_{s,s-1} * k_{s-1})), \\
y_{n+1} &= y_n + \Delta t (b_1 * k_1 + b_2 * k_2 + \dots + b_s * k_s).
\end{aligned}$$

Dabei muss gelten:

$$c_i = \sum_j a_{ij}$$

Aus diesem Schema ergibt sich aus einer initialen Näherung $y_0 := y(t_0)$ eine Iterationsvorschrift für $y_n = y(t_n)$. Vereinfachend stellt man dieses Schema wie in Tabelle A.3.2 dar:

A.3.3 Das Runge-Kutta-Verfahren vierter Ordnung

Aus der großen Anzahl möglicher Verfahren wird hier das gängigste beschrieben. Es handelt sich um ein Verfahren vierter Ordnung und die Funktion wird zweimal in der Mitte des Lösungsintervalls Δt angenähert. Für $n \in \mathbb{N}_0$ sei

$$\begin{aligned}
k_1 &= f(t_n, y_n), \\
k_2 &= f\left(t_n + \frac{1}{2} \Delta t, y_n + \Delta t \left(\frac{1}{2} * k_1\right)\right), \\
k_3 &= f\left(t_n + \frac{1}{2} \Delta t, y_n + \Delta t \left(0 * k_1 + \frac{1}{2} * k_2\right)\right), \\
k_4 &= f(t_n + \Delta t, y_n + \Delta t (0 * k_1 + 0 * k_2 + 1 * k_3)), \\
y_{n+1} &= y_n + \Delta t \left(\frac{1}{6} k_1 + \frac{2}{6} k_2 + \frac{2}{6} k_3 + \frac{1}{6} k_4\right).
\end{aligned}$$

Als Schema notiert besitzt das Verfahren die Gestalt aus Tabelle A.3.3.

| | | | | |
|---------------|---------------|---------------|---------------|---------------|
| 0 | | | | |
| $\frac{1}{2}$ | $\frac{1}{2}$ | | | |
| $\frac{1}{2}$ | 0 | $\frac{1}{2}$ | | |
| 1 | 0 | 0 | 1 | |
| | $\frac{1}{6}$ | $\frac{2}{6}$ | $\frac{2}{6}$ | $\frac{1}{6}$ |

Tabelle A.2: Die Koeffizienten des Runge-Kutta-Verfahrens vierter Ordnung

A.3.4 Die Implementation des Verfahrens

Für Runge-Kutta-Verfahren ist eine Klasse implementiert worden. Dazu werden die Koeffizienten aus A.3.3 benutzt. Diese werden in einer Textdatei gespeichert und bei der Initialisierung eingelesen. Die Benennung der zugehörigen Felder der Klasse folgt der aus Tabelle A.3.2. Da die Größe der Tabelle von der Ordnung s des Verfahren abhängt, wird sie am Anfang der Textdatei eingelesen.

Schließlich benötigt das Verfahren eine Methode, mit der die Funktion f mit Hilfe des Schemas ausgewertet werden kann. Hierzu enthält die Klasse einen Methodenzeiger vom Typ

```
procedure (Zustaende : array of double;
          var NeueZustaende : array of double) of object;
```

Zur Laufzeit wird diesem Zeiger die passende Funktion zugeordnet. Sie wird derart angewandt, dass im ersten Feld die aktuellen Funktionswerte übergeben werden und im zweiten Feld die veränderten zurückgegeben werden. Im Fall der generischen Struktur für L/R-Modelle ist dies die Methode `NewRates` der Klasse `TPMModel`, die im Abschnitt 5.2.1 beschrieben worden ist.

Anhang B

Backpropagation-Netze

Im Verlauf der Arbeit ist mit den Backpropagation-Netzen ein weiterer Typ von Modellen verwendet worden, die bei der Kalibrierung von Modellen (vgl. 5.3) eingesetzt worden sind. Dort ist im Zusammenhang mit Kalibrierungsmodellen nur von Interesse, dass sie sich in Lernphase an gegebene Ein- und Ausgangsdaten anpassen und in der Nutzungsphase zu den gegebenen Eingängen die zugehörigen Ausgänge berechnen. (vgl. Abbildung 5.15). Hier nun sollen diese Modelle insoweit betrachtet werden, wie es für die vorgenommene Implementation notwendig ist. Für genauere Betrachtungen wird auf [Hay94] und [PU92] verwiesen.

B.1 Überblick

Bei der Verwendung von neuronalen Netzen kann man grundsätzlich zwei verschiedene Arten unterscheiden. Hierzu wird auf den relationalen Systembegriff aus Abschnitt 1.4 zurückgegriffen. Dort wird ein System über seine Eingänge und Ausgänge definiert. Ist die Datenlage des Systems derart, dass man zu gegebenen Eingängen die Ausgänge kennt, so kann man einen Algorithmus mit ÜBERWACHTEM LERNEN einsetzen, der versucht, die an dieser Stelle noch unbekannte Struktur des Netzes so anzupassen, dass dessen Ausgabe den gegebenen Daten so gut wie möglich entspricht. Andererseits kann es Datenlagen geben, bei denen die Ausgabe nicht bekannt sind. Hier sind Verfahren des UNÜBERWACHTEN LERNENS entwickelt worden. Für die Kalibrierung von Modellen durch Data Mining wird lediglich das überwachte Lernen benötigt. Die Darstellung beschränkt sich daher auf diesen Punkt.

Die grundsätzliche Handhabung eines neuronalen Netzes teilt sich in zwei Phasen auf. Zunächst wird in der aufwändigen LERNPHASE das Netz generiert. Hierzu sind LERNMUSTER notwendig; das sind Datensätze, die das durch das neuronale Netze zu modellierende System beschreiben. In der zweiten NUTZUNGSPHASE wird dem neuronalen Netz lediglich das aktuelle Eingangsmuster gezeigt, und dieses reagiert mit der jeweiligen Ausgabe. Man spricht in diesem Zusammenhang von der Fähigkeit neuronaler Netze zu GENERALISIEREN: Sie können von den in der Lernphase „gesehenen“ auf bisher unbekannte Muster verallgemeinern, die dem Netz erst in der Nutzungsphase gezeigt werden.

Die Darstellung geht vom statischen Modell eines neuronalen Netzes aus und stellt in einem zweiten Schritt dessen Dynamik vor. Sie wird auch in der Lernphase, deren Algorithmus im dritten Abschnitt beschrieben wird, benutzt. An diesen Stellen wird demonstriert, wie die im Kapitel 3 eingeführten S/T-Netze im Zuge der Modellbildung eingesetzt werden können. Dieser Anhang schließt mit der Anwendung eines Backpropagation-Netzes im physiologischen Modell.

B.2 Netzstruktur

In einer Folge von Definitionen werden alle notwendigen Begriffe entwickelt, um die Struktur der verwendeten neuronalen Netze zu erläutern. Schließlich werden die Begriffe in dem in diesem Kontext üblichen Darstellungsmodell veranschaulicht und es wird ein Hinweis auf eine einfache Realisierung gegeben. Damit finden sich auch hier die drei Teile von Modellen System-, Simulations- und Darstellungsmodell, wie es im Abschnitt 2.3.1 über den Standardentwurf gesagt worden ist.

Definition B.1 (Neuron). *Ein Neuron $n = (A, f, F)$ ist definiert durch*

1. eine ZUSTANDSMENGE $A \subseteq \mathbb{R}$
2. eine AUSGABEFUNKTION $f : A \rightarrow \mathbb{R}$
3. und eine AKTIVIERUNGSFUNKTION $F : A \times \mathbb{R}^m \rightarrow \mathbb{R}$, wobei m die Anzahl der Eingänge des Neurons ist.

Als Kandidaten für Ausgabefunktionen haben sich so genannte SIGMOID-FUNKTIONEN wie

$$\phi_a(x) = \frac{1}{1 + \exp(-ax)}, \quad a \in \mathbb{R}, \quad (\text{B.1})$$

$$\phi(x) = \frac{1 - \exp(-x)}{1 + \exp(-x)} \quad (\text{B.2})$$

durchgesetzt. Sie erleichtern die Theoriebildung und liefern die gewünschten Ergebnisse, indem sie die reellen Zahlen in einen gewissen Wertebereich transformieren und darüber hinaus noch unendlich oft differenzierbar sind.

Die einzelnen Neuronen werden über VERBINDUNGEN verschaltet.

Definition B.2 (Verbindung, Gewicht der Verbindung). *Sei $e_i^{(k)}$ der k -te Eingang des Neurons i , o_j der Ausgang des Neurons j und $w_{ij}^{(k)}$ das GEWICHT DER VERBINDUNG. Dann bezeichnet das Tripel $(e_i^{(k)}, o_j, w_{ij}^{(k)})$ die VERBINDUNG zwischen dem Ausgang von j und dem k -ten Eingang von i .*

Mit diesen Begriffen kann ein NEURONALEN NETZES definiert werden.

Definition B.3 (neuronales Netz). *Ein Tripel $N = (\mathcal{N}, \mathcal{W}, I)$ heißt ein NEURONALES NETZ N , falls*

1. I eine Indexmenge ist,

2. $\mathcal{N} = \{n_i \mid i \in I\}$ eine Menge von Neuronen n darstellt und
3. eine Menge \mathcal{W} von gewichteten Verbindungen zwischen den Neuronen gegeben ist.

Diese Struktur wird mit Hilfe des Begriffs der Verbindungen genauer beschrieben.

Definition B.4 (Netzeingänge und -ausgänge). Ein Eingang $e_i^{(k)}$ heißt NETZEINGANG, falls $e_i^{(k)}$ keinen Vorgänger¹ hat. Falls alle Eingänge von i Netzeingänge sind, so nennt man i auch EINGANGSNEURON. Entsprechend nennt man den Ausgang von j NETZ-AUSGANG, falls j keine Nachfolger hat.

Definition B.5 (Feedforward-Netz). Ein FEED-FORWARD-NETZ ist ein neuronales Netz ohne Schleifen.²

Definition B.6 (Schicht). Eine Teilmenge von nicht verbundenen Neuronen bezeichnet man als SCHICHT. Ist die Neuronenmenge partitioniert, so spricht man von einer Zerlegung in Schichten.

Im Allgemeinen wird man die Schichten funktional aufteilen. Eingangsneuronen und Ausgangsneuronen bilden jeweils eine Schicht, sowie dazwischenliegende versteckte Schichten. Neuronale Netze mit drei Schichten können jede quadratisch-integrierbare Funktion $f : [0, 1]^m \rightarrow \mathbb{R}^n$ beliebig genau approximieren können.³ Die folgenden Beispiele beschränkten sich auch auf dreischichtige Netze, da sie auch in der Anwendung in der Atmungsphysiologie verwendet worden sind.

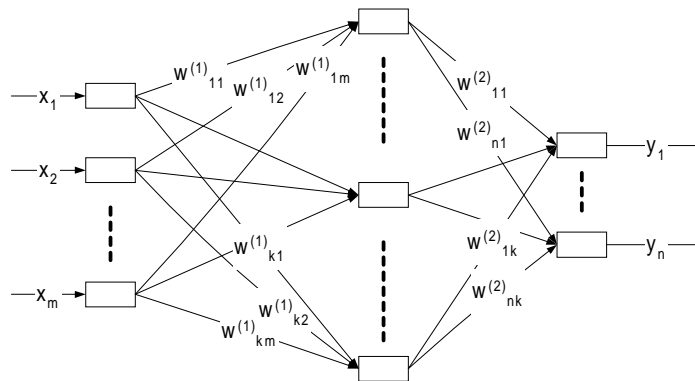


Abbildung B.1: Ein Feedforward-Netz mit drei Schichten

Die Darstellungsmodelle von neuronalen Netzen sind gerichtete Graphen. An die Kanten werden die Gewichte der Verbindungen geschrieben. Abbildung B.1 zeigt ein dreischichtiges Netz mit m Eingangsneuronen, einer verdeckten Zwischenschicht mit k und der Ausgangsschicht mit n Neuronen.

Für die Realisierung eines Netzes ist es günstig, die Gewichte zwischen den Schichten in Matrizen darzustellen. In dem Beispiel sind die Gewichte in den beiden Matrizen $W^{1,2}$ und $W^{2,3}$ abgebildet.

¹Dabei werden die Begriff Vorgänger und Nachfolger in der gewohnten Weise definiert.

²Dabei bezeichnet man mit einer Schleife einen geschlossener Pfad von Verbindungen der Neuronen.

³Vgl. [HN91] S. 132.

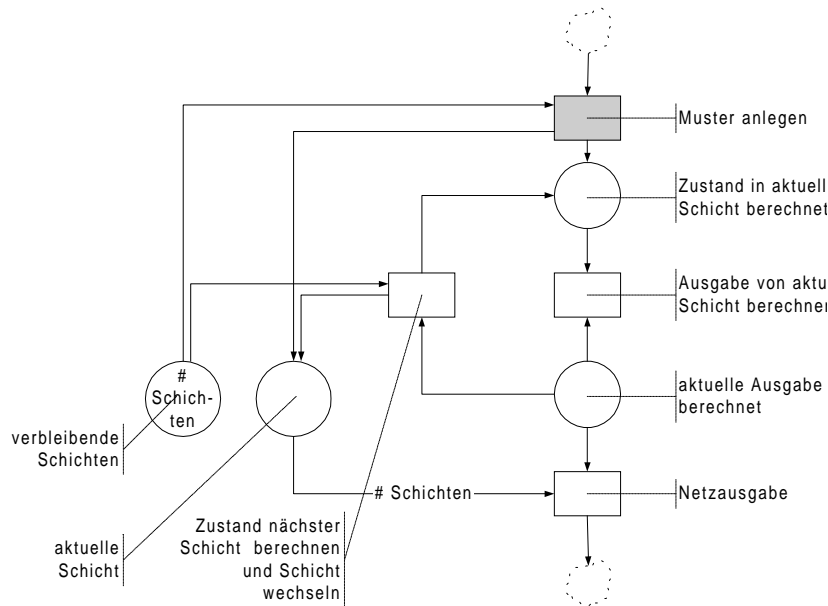


Abbildung B.2: Berechnung der Netzausgabe in einem geschichteten Netz

$$W^{1,2} = \begin{pmatrix} w_{11}^{(1)} & w_{12}^{(1)} & \dots & w_{1m}^{(1)} \\ \vdots & \vdots & \vdots & \vdots \\ w_{k1}^{(1)} & w_{k2}^{(1)} & \dots & w_{km}^{(1)} \end{pmatrix}, \tag{B.3}$$

$$W^{2,3} = \begin{pmatrix} w_{11}^{(2)} & w_{12}^{(2)} & \dots & w_{1k}^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ w_{n1}^{(2)} & w_{n2}^{(2)} & \dots & w_{nk}^{(2)} \end{pmatrix}.$$

Durch diese Netzdarstellung in Matrizen kann deren Dynamik auf Matrizenmultiplikationen zurückgeführt werden.

B.3 Netzdynamik

Die Netzdynamik wird für ein in obigem Sinne in Schichten unterteiltes Feed-Forward-Netz mit m Ein- und n Ausgängen beschrieben. Dynamik entsteht dadurch, dass an die Eingangsschicht Werte, sog. MUSTER, angelegt werden. Diese werden gemäß des zu beschreibenden Algorithmus in ein Ausgangsmuster verwandelt, das an den Ausgangsneuronen abgelesen werden kann. In dieser Sichtweise ist ein neuronales Netz eine Abbildung $N : \mathbb{R}^m \rightarrow \mathbb{R}^n$.

Anhand der Dynamik von neuronalen Netze wird beispielhaft gezeigt, wie Stellen-Transitions-Netze aus Abschnitt 3.2 zur Modellierung benutzt werden. Die Stellen symbolisieren erreichte Zustände. Zur Beschreibung der Dynamik werden Ereignisse verwendet. Abbildung B.2 zeigt dies für die Berechnung der Netzausgabe. Dabei werden Kanten nur

beschriftet, wenn sie Gewichte ungleich 1 haben. Insgesamt sind folgende Ereignisse zu definieren:

1. **Muster anlegen:** Das Eingangsmuster $x = (x_1, \dots, x_m)$ liefert die Zustände der Eingangsschicht $v_i^{(1)} = x_i$ für $1 \leq i \leq m$.
2. **Ausgabe berechnen:** Ist der Zustand $v_j^{(k)}$ eines Neurons j in der Schicht k bekannt, so ergibt sich dessen Ausgabe aus der Ausgabefunktion $o_j^{(k)} = f(v_j^{(k)})$.
3. **Zustand nächster Schicht berechnen:** Solange nicht alle Schichten durchlaufen sind, ergeben sich die Zustände der Neuronen $v_i^{(k+1)}$ der folgenden Schicht $k+1$ aus:⁴

$$v_i^{(k+1)} = \sum_{j \in P(i)} w_{ij}^{(k)} o_j^{(k)}$$

In Matrixschreibweise heißt dies $\vec{v}^{(k+1)} = W^{k,k+1} * \vec{o}^{(k)}$.

4. **Netzausgabe:** Die Netzausgabe ergibt sich aus den Ausgabefunktionen der letzten Schicht L : $y_i^{(L)} = o_i^{(L)}$ für $1 \leq i \leq n$.

Insgesamt ergibt sich für einen Eingangsvektor \vec{x} folgende Ausgabe \vec{y} in Matrixschreibweise:

$$\vec{y} = \phi \left(W^{2,3} * \phi \left(\left(W^{1,2} * \vec{x} \right) \right) \right)$$

Dabei wird die sigmoide Funktion ϕ hier auf jede Komponente des Vektors angewandt.

B.4 Die Lernphase

Neuronale Netze müssen bei überwachtem Lernen aus Paaren von Ein- und Ausgängen ihre innere Struktur, die durch die einzelnen Gewichte gegeben ist, anpassen. Dies geschieht dadurch, dass dem Netz iterativ über t Generationen die gegebenen Lernmuster präsentiert werden und es seine Gewichte jeweils an die Lernmuster adaptiert. Ein bewährter Algorithmus hierfür ist der Backpropagation-Algorithmus, der hier beschrieben wird. Dabei folgt die Darstellung dem Diagramm B.3, in dem folgende Ereignisse dargestellt sind:

1. **Vorwärtsphase:** Sie erfolgt nach dem im Diagramm B.2 vorgestellten Prozess.
2. **Ausgabefehler berechnen:** Für die n Ausgangsneuronen berechnen sich die Fehler als Differenzen aus den gewünschten Ausgängen d_j tatsächlichen Ausgängen o_j .

$$e_j(n) = d_j(t) - o_j(t), \quad j = 1, \dots, n.$$

⁴Dabei bezeichnet $P(i)$ die Menge der Vorgänger des Neurons i .

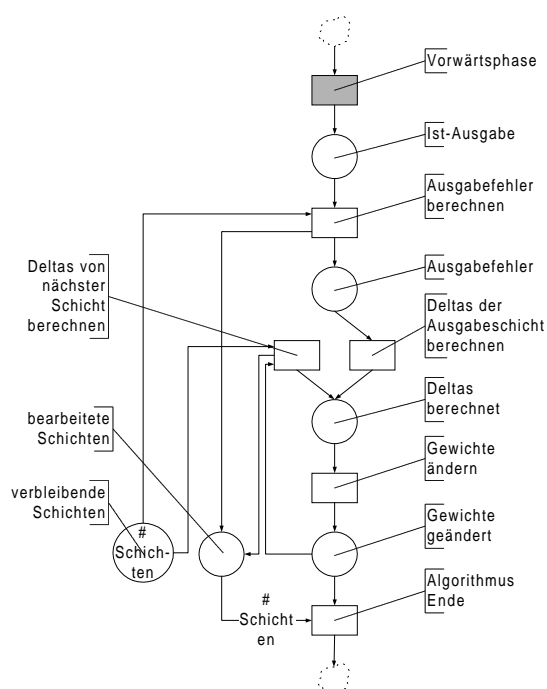


Abbildung B.3: Backpropagation in einem neuronalen Netz

3. **Deltas der Ausgangsschicht berechnen:** Die lokalen Gradienten δ_j der n Neuronen der Ausgangsschicht L ergeben sich als:⁵

$$\delta_j^{(L)}(t) = e_j^{(L)}(t) o_j(t) (1 - o_j(t)), \quad j = 1, \dots, n.$$

4. **Gewichte ändern:** Aus diesen Informationen ergeben sich die neuen Gewichte der Verbindungen $w_{ji}^{(l)}$ vom i -ten Neuron der $l - 1$ Schicht, das mit dem Neuron j der l -ten Schicht als:

$$w_{ji}^{(l)}(t + 1) = w_{ji}^{(l)}(t) + \eta \delta_j^{(l)} y_i^{(l-1)}(t)$$

Dabei durchlaufen die Indizes j und i die jeweilige Zahl der Neuronen der Schichten.

5. **Deltas der nächsten Schicht berechnen:** Die lokalen Gradienten δ_j der Neuronen j übrigen Schichten l ergeben sich jeweils als:

$$\delta_j^{(l)}(t) = y_j^{(l)}(t) (1 - y_j^{(l)}(t)) \sum_k \delta_k^{(l+1)} w_{kj}^{(l+1)}(t)$$

⁵Beim Backpropagation-Algorithmus handelt es sich um ein Gradientenabstiegsverfahren. Zur Herleitung vergleiche man etwa [Hay94] S. 142f.

B.5 Anwendung in der Atmungsphysiologie

Im folgenden Abschnitt wird beschrieben, welches neuronale Netz im Rahmen der Kalibrierung der Blutgase im physiologischen Modell verwendet wird. Die Eingangsgrößen in das Netz sind:

1. das gemessene Atemminuten-Volumen,
2. der arterielle Sauerstoffpartialdruck P_aO_2 des Patienten,
3. der arterielle Kohlendioxidpartialdruck P_aCO_2 des Patienten,
4. die inspiratorische Sauerstofffraktion F_{IO_2} ,
5. und das aus dem Körpergewicht geschätzte Herz-Zeit-Volumen des Patienten.

Die sich ergebenden Ausgaben der Kalibrierungsparameter sind die relativen Kompartimentgrößen:

1. Totraumfraktion,
2. Shuntfraktion.

Bei dem Backpropagation-Netz handelt es sich um ein dreischichtiges Feedforward-Netz. Neben den 5 Dateneingängen hat es noch Bias-Eingang, der konstant gleich -1 ist.⁶ Die Zwischenschicht enthält 10 Neuronen und die Ausgangsschicht die 2 Neuronen für die Datenausgänge, wie es Figur B.4 zeigt. Dieses Netz ist über die zwei Matrizen W_1 und W_2 in Gleichung (B.4) implementiert. Als Ausgabefunktion der Neuronen wird die sigmoide Funktion $\phi_1(x) = 1/(1 + \exp(-x))$ mit Wertebereich zwischen 0 und 1 benutzt. Die Informationen über die Netzarchitektur sind in einer Konfigurationsdatei gespeichert. Mit diesen Informationen kann sich das Netz selbst erzeugen und die Gewichte aus einer Datei einlesen. Dadurch können später anders konfigurierte Netze in das physiologische Modell integriert werden.

Schließlich sind die Eingangs- und Ausgangsdaten geeignet zu transformieren. Die Eingangsdaten werden derart standardisiert, dass sie einen Mittelwert von 0 und eine Streuung von 1 haben, während bis auf einen Offset die Ausgangsdaten in den Wertebereich der sigmoiden Funktion transformiert worden sind. Hierzu werden die in einer weiteren Initialisierungsdatei gespeicherten Statistiken der in der Lernphase benutzten Daten verwendet. Im praktischen Einsatz werden die am Patienten gemessenen Eingangsdaten zunächst standardisiert, dann dem Netz präsentiert, das hierzu Ausgangsdaten liefert, die schließlich zurück transformiert werden.

⁶Zu der Funktion von Biasneuronen vergleiche man etwa [Hay94] S. 8.

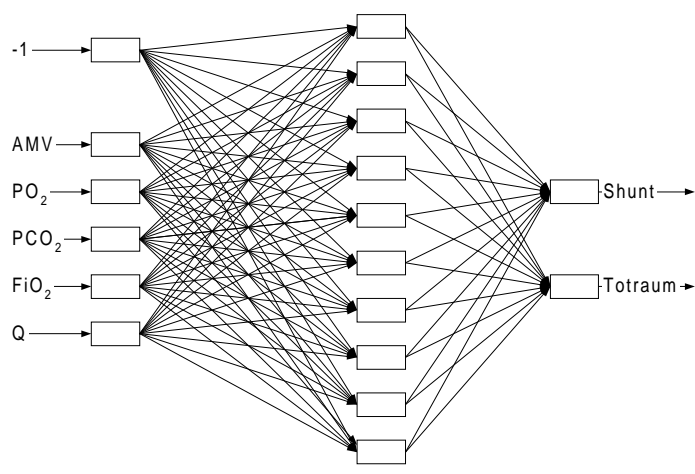


Abbildung B.4: Die Struktur des neuronalen Netzes zur Kalibrierung der Blutgase

$$W^{1,2} = \begin{pmatrix} -9,43501 & -0,09631 & 14,06857 & 0,59930 & 1,03631 & -0,20688 \\ 5,26377 & -0,45138 & -3,63388 & -1,04934 & 12,83631 & -0,13321 \\ 9,39154 & -2,85138 & -3,95871 & -4,79426 & 2,61910 & 3,15890 \\ -3,55064 & -4,17851 & -2,51448 & 4,51893 & 1,24921 & 1,19060 \\ 6,77371 & 2,96228 & -0,94968 & 4,31339 & -5,00901 & -2,22556 \\ -7,05158 & 3,70169 & -1,05497 & -4,77653 & 4,19332 & 0,90843 \\ -16,04408 & -1,74261 & 2,40725 & 0,79384 & -0,86508 & 4,72066 \\ 20,53250 & -4,46906 & -1,61480 & -24,10514 & 0,52197 & -1,77376 \\ -22,14504 & 9,03362 & -0,10610 & 1,35120 & 0,51358 & 2,58174 \\ 10,44476 & -13,08674 & 1,87783 & -1,12348 & -5,11380 & 1,62333 \end{pmatrix}, \tag{B.4}$$

$$W^{2,3} = \begin{pmatrix} -7,65833 & 7,01998 & 3,61367 & 1,40719 & -2,14005 & -0,83432 & -2,98818 & 6,46638 & 1,57516 & 1,019739 \\ -2,05705 & -0,84119 & -1,25845 & 1,49178 & 2,60319 & 3,065678 & 1,73827 & -11,76699 & -1,38684 & -2,80809 \end{pmatrix}.$$

Anhang C

Physiologisches Modell: Einheiten und Parameter

Es wird im Zuge einer relationalen Beschreibung des Modells angegeben, mit welchen Parametern das physiologische Modell vom EWS gesteuert wird und welche Werte es an das Expertensystem zurückgibt. Zuvor werden aber die im Modell benutzten Einheiten benannt.

C.1 Die Einheiten

In der folgenden Tabelle werden die im physiologischen Modell benutzten Einheiten angegeben. Dabei wird die Anzahl der Teilchen mit dem Volumen, das sie einnehmen identifiziert, so dass Volumeneinheiten angegeben werden.

| Größe | Einheit |
|----------------|-------------|
| Zeit | s |
| „Gas“-Druck | mbar |
| Frequenz | 1/min |
| Resistance | s * mbar/ml |
| Compliance | ml/mbar |
| Zeitkonstante | s |
| Volumen | ml |
| Volumenfluss | ml/s |
| Teilchen | ml |
| Teilchenfluss | ml/s |
| Fraktion | %/100 |
| Gehalt | ml/ml |
| Partialdruck | mmHG |
| Minutenvolumen | ml/min |

C.2 Die Parameter

Das physiologische Modell wird vom EWS gesteuert. Die Kommunikation der Modellparameter erfolgt über zwei Methoden, bei denen die Parameter über einen String `name` identifiziert werden.

```
procedure SetParam (name : String; value : Double);
function GetParam (name : String) : Double;
```

Gemäß der Unterscheidung in 5.1 sind im physiologischen Modell folgende Arten von Parametern zu unterscheiden:

1. Sitzungsparameter
2. Simulationsparameter
3. Kalibrierungsparameter
4. Modellkonstanten

Sitzungsparameter: Eine Sitzung beim physiologischen Modell wird von einem Patienten bestimmt. Hier sind also die Werte zu setzen, die sich nur ändern können, wenn ein anderer Patient vorliegt. Im Falle des physiologischen Modells beschränken sich die Sitzungsparameter auf das Körpergewicht.

| Name | String | Einheiten | Default |
|---------|------------|-----------|---------|
| Gewicht | Bodyweight | kg | 75 |

Simulationsparameter: Sie müssen vor jeder Simulation vom EWS gesetzt werden. Dabei handelt es sich um die Parameter zur Modellierung des Beatmungssignales und vorhandene Messwerte, wie folgende Tabelle zeigt:

| Name | String | Einheiten | Default |
|-----------------------------|----------------|---------------------|---------|
| P_{Insp} | P_Insp | mbar | 30 |
| P_{ASB} | P_Asb | mbar | 20 |
| PEEP | PEEP | mbar | 10 |
| $T_{\text{I Mand}}$ | T_Insp | s | 3 |
| $T_{\text{I Spn}}$ | T_spn | s | 1 |
| f_{mand} | F_contr | 1/min | 10 |
| f_{spn} | F_Spn | 1/min | 0 |
| V_{ds} | V_ds | ml | 200 |
| $F_{\text{I O}_2}$ | FiO2 | %/100 | 0.2 |
| CO ₂ -Produktion | CO2-Production | ml/min ¹ | 220 |

¹Dies ist die Einheit in der die CO₂-Produktion vom EWS gesetzt wird. Intern wird in ml/s gerechnet.

Kalibrierung

Im Zuge der Kalibrierung ist die Mechanik der Ventilation und die Kalibrierungsaufgabe der Blutgase zu lösen.

Zielwerte der Mechanik-Kalibrierung. Die Zielwerte für die Kalibrierung der Mechanik sind die gemessenen Atemminuten-Volumina:

| Name | String | Einheiten | Default | Bemerkung |
|--------------------|---------------|-----------|---------|-----------------------|
| MV_{mand} | MV_contr_Goal | ml/min | 7600 | masch. Minutenvolumen |
| MV_{spn} | MV_Spn_Goal | ml/min | 0 | spont. Minutenvolumen |

Kalibrierungsparameter für die Mechanik. Die Kalibrierungsparameter zur Kalibrierung der Mechanik werden bis auf die Augmentierung im EWS bestimmt und an das physiologische Modell übergeben:

| Name | String | Einheiten | Default | Bemerkung |
|------------------|---------------|-----------|---------|---------------|
| C | Compliance | ml/mbar | 40 | Compliance |
| τ | Tau | s | 1 | Zeitkonstante |
| V_{Aug} | Augm.-Volumen | ml | 0 | Augmentierung |

Zielwerte der Blutgas-Kalibrierung. In einer Blutgasanalyse werden die arteriellen Partialdrücke des Kohlendioxids und des Sauerstoffs bestimmt, die als Zielwerte der Kalibrierung dienen:

| Name | String | Einheiten | Default | Bemerkung |
|--------------------|-----------|-----------|---------|---------------------|
| P_{aO_2} | PO2_Goal | mmHG | 100 | arterielle Blutgase |
| P_{aCO_2} | PCO2_Goal | mmHG | 40 | arterielle Blutgase |

Kalibrierungsparameter für die Blutgase. Als Kalibrierungsparameter dienen die Größe der Kompartimente.

| Name | String | Einheiten | Default | Bemerkung |
|--------|-----------|-----------|---------|-----------------|
| PZ_S | %-Shunt | %/100 | 0,7 | Shuntfraktion |
| PZ_N | %-Normal | %/100 | 0,2 | Normal-Fraktion |
| PZ_T | %-Totraum | %/100 | 0,1 | Totraumfraktion |

Weitere Parameter der Blutgas-Kalibrierung. Im Zuge einer Blutgasanalyse werden auch weitere Parameter bestimmt, die für die Modellierung der Atmungsphysiologie verwendet werden:

| Name | String | Einheiten | Default | Bemerkung |
|------|----------|-----------|---------|-----------------|
| Hb | Hb | g/ml | 0,15 | Hämoglobin |
| BE | BE | mmol/l | 0 | Basenüberschuss |
| pH | pH-Value | - | 7,4 | pH-Wert |

Ausgaben

Zu einer relationalen Beschreibung des Modells gehören auch dessen Ausgaben.

Ausgaben im Rahmen des EWS. Die folgenden Werte benutzt das EWS zur Entscheidungshilfe und/oder zur Abschätzung, ob das Modell den Patienten noch adäquat abbildet.

| Name | String | Einheiten | Bemerkung |
|------------------|----------------|-----------|-----------------------------|
| P_{aO_2sim} | PAO2-Min | mmHG | arterielle Blutgase |
| P_{aCO_2sim} | PACO2-Min | mmHG | arterielle Blutgase |
| $MV_{mand-sim}$ | MV_contr | ml/min | maschinelles Minutenvolumen |
| $MV_{spn-sim}$ | MV_Spn | ml/min | spontanes Minutenvolumen |
| \dot{V}_{Norm} | — ² | ml/min | Normventilation |

Weitere Ein- und Ausgaben. Neben den benutzten Ausgabegrößen und den eingestellten Kalibrierungsparametern werden derzeit noch andere Parameter und simulierte Werte an das EWS übergeben, die zwar nicht benutzt, aber in der folgenden Tabelle der Vollständigkeit halber angegeben werden. Sie werden ggf. in späteren Versionen und in anderen Zusammenhängen benutzt werden.

| Name | String | Einheiten | Bemerkung |
|-------------------------------|----------------|-------------|--------------------------------------|
| C_{Gem} | Compliance | ml/mbar | übergebene Compliance |
| R_{Ges} | Resistance | s * mbar/ml | Gesamt-Resistance |
| HZV | Q | ml/s | Herz-Zeit-Volumen |
| RQ | RQ | - | Resp. Quotient |
| etCO ₂ | etCO2-Min | %/100 | endtidales CO ₂ |
| S _a O ₂ | SAO2-Min | % | arterielle O ₂ -Sättigung |
| Hb | Hb | g/ml | Hämoglobin |
| BE | BE | mmol/l | Basenüberschuss |
| pH | pH-Value | - | pH-Wert |
| CO ₂ -Produktion | CO2-Production | ml/min | CO ₂ -Produktion |
| V _{ds} | V_ds | ml | serieller Totraum |

²Die Normventilation wird über die Funktion `Get_NormVentilation` abgefragt und nicht über die `GetParam`-Methode, so dass auch kein String benötigt wird.

Anhang D

Ein wenig Elektrotechnik

Im Hauptteil der Arbeit ist die Parallele zwischen der Elektrotechnik und der Atmungsphysiologie gezogen worden, um die Struktur des Modells aus Kompartimenten besser zu verstehen. Hierzu sind Stromkreise aus einer Spannungsquelle— dem Beatmungsgerät—, Widerständen(Resistance) und Kondensatoren mit ihrer Kapazität oder Compliance genutzt worden. Dies setzt eine etwas genauere Beschäftigung mit den physikalischen Gegebenheiten voraus, die hier in einem Anhang zusammengefasst werden. Dabei orientiert sich die Darstellung an [GKV82] S. 269 ff.

So wie beim beatmeten Patienten Druck am Beatmungsgerät angelegt wird, legt man an einer Quelle im Stromkreis Spannung an. Die Folge davon ist, dass sich Ladungen Q bewegen, also Strom fließt. Dies führt zur Definition der Stromstärke I :

$$I = \frac{dQ}{dt}. \quad (\text{D.1})$$

Ein Maß für die „Menge“ des fließenden Stromes ist die bewegte Ladungsmenge dQ pro Zeiteinheit dt oder kurz, die Ableitung der Ladung nach der Zeit. Dies ist im Fall des Stromkreises die kontinuierliche Form der Ratengleichung für den Volumenfluss im Modell der Ventilation.

In einem Stromkreis mit Gleichstrom gehorcht die Stromstärke einem einfachen Gesetz:

Regel D.1 (Kirchhoffsche Knotenregel). *An jedem Verzweigungspunkt (Knoten) in einer Schaltung muss ebensoviel Ladung zu- wie abfließen. Die Summe aller Ströme in den einzelnen Zweigen, die in den Knoten münden, ist Null:*

$$\sum I_i = 0. \quad (\text{D.2})$$

Dabei ist selbstverständlich auf das Vorzeichen, die Richtung des Flusses, zu achten. Die Regel setzt Gleichstrom mit konstanter Spannung voraus. Ferner gilt, dass innerhalb eines geschlossenen Stromkreises keine Ladung abgezogen wird und so die Stromstärke überall die selbe ist.

Betrachtet man die Spannungsabfälle innerhalb eines Stromkreises, so führt dies auf folgende Regel:

Regel D.2 (Kirchhoffsche Maschenregel). Die Gesamtspannung längs einer geschlossenen Masche einer Schaltung, d.h. die Summe aller Spannungsabfälle an den einzelnen Elementen, aus denen die Masche besteht, ist Null.

$$\sum U_i = 0 \quad (\text{D.3})$$

In vielen wichtigen Fällen ist nach dem Ohmschen Gesetz die angelegte Spannung U proportional zum fließenden Strom I . Man nennt die Proportionalitätskonstante Widerstand R :

$$U = R * I. \quad (\text{D.4})$$

Dieses Gesetz ist in Kapitel 7.2.1 zur Definition der Resistance benutzt worden.

In einem Stromkreis mit Gleichstrom gelten für die Widerstände und der Kondensatoren folgende Regeln:

Regel D.3 (Widerstandsregel). Für n Widerstände in einem Stromkreis gilt:

1. Sind die Widerstände in Reihe geschaltet, so addieren sich die einzelnen Widerstände zum Gesamtwiderstand:

$$R_{\text{Ges}} = \sum_{i=1}^n R_i. \quad (\text{D.5})$$

2. Sind die Widerstände parallel geschaltet, so addieren sich die Kehrwerte zum Kehrwert des Gesamtwiderstandes:

$$\frac{1}{R_{\text{Ges}}} = \sum_{i=1}^n \frac{1}{R_i}. \quad (\text{D.6})$$

Ähnliche Regeln gelten für die Kapazitäten:

Regel D.4 (Kapazitätsregeln). Für n Kondensatoren in einem Stromkreis gilt:

1. Sind die Kondensatoren parallel geschaltet, so addieren sich die einzelnen Kapazitäten zur Gesamtkapazität.

$$C_{\text{Ges}} = \sum_{i=1}^n C_i \quad (\text{D.7})$$

2. Sind sie Reihe geschaltet, so addieren sich die Kehrwerte zum Kehrwert der Gesamtkapazität.

$$\frac{1}{C_{\text{Ges}}} = \sum_{i=1}^n \frac{1}{C_i} \quad (\text{D.8})$$

Die Tatsache, dass zwischen der Spannung U und der Ladung Q an einem Kondensator ein proportionaler Zusammenhang mit der Kapazität C als Proportionalitätskonstanten herrscht, bedeutet in der Atmungsphysiologie folgendes: In Abschnitt 7.2.1 die Compliance als Ableitung des Volumens nach dem Druck eingeführt und für den physiologischen Normalfall vereinfachend als konstant angenommen worden.

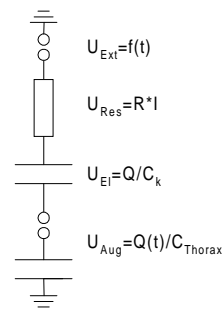


Abbildung D.1: Das Einkompartiment-Modell als Stromkreis

Abschließend zeigt Abbildung D.1 das Einkompartiment-Modell aus Abbildung 7.3 in seiner Darstellung als Stromkreis. Er wird durch die beiden Erden geschlossen. Nimmt man an, dass an der externen Spannungsquelle U_{Ext} Gleichstrom angelegt ist, sind die Spannungsabfälle im Stromkreis nach der Maschenregel genauso groß. Daher folgt hier entsprechend der Gleichung (7.5)

$$U_{\text{Ext}} = U_{\text{Res}} + U_{\text{El}} + U_{\text{Aug}}, \quad (\text{D.9})$$

$$U_{\text{Ext}} = R * I + \frac{Q_k}{C_k} + \frac{Q(t)}{C_{\text{Thorax}}}. \quad (\text{D.10})$$

Anhang E

Aspekte der Realisierung

E.1 Die generische Struktur: Das Beispiel eines Kompartimentes

Am Beispiel eines belüfteten Kompartimentes wird gezeigt, wie eine Komponente innerhalb der generischen Struktur für modulare L/R-Modelle im Zusammenhang des physiologischen Modells entwickelt wird.

E.1.1 Die Deklaration der Klasse

Zunächst zeigt der folgende Abschnitt die Deklaration der Klasse `TPMVolCompartment`. Sie modelliert die Zusammenhänge im Kontext des Gesamtvolumens.

```
TPMVolCompartment = class(TPM_Component)
  private

  protected

  FWithAugmentation          : boolean;
  FCompliance, FPercentage,
  FVolume                    : double;

  FTracheaPointer           : TPMTrachea;
  FLungPointer              : TPMLung;
  FAlveole                  : TPMAlveolarCompartment;
  {
    wird erst in Complet-Compartment gesetzt
    aber hier wegen Gasaustausch gebraucht
  }

  function GetPCompartment   : double;           virtual;
  function GetFlwVolCompartment : Double;       virtual;
  {
```

```

    Größen werden in eigenen geschützten Funktionen implementiert}
}

public

    procedure Build;                                override;
    procedure InitModel_ConnectComponents;          override;

    procedure InitSimulation_SetParam;              override;
    procedure SetDefaultStartvalues;                override;

    procedure CalculateRates;                        override;
    procedure FlowAddition;                          override;
end;
```

Im öffentlichen Teil der Deklaration finden sich die Methoden, die in Tabelle 5.2 angegeben sind. Diese werden im Implementationsteil überschrieben. Die privaten Deklarationen werden bei der Darstellung der Implementation erläutert.

E.1.2 Die Implementation der Klasse

Die Routinen zum Erzeugen eines Objektes der Klasse, der Simulationsinitialisierung und der Simulationsdurchführung sind zu implementieren.

Das Erzeugen der Klasse

Die Routinen `build` und `InitModel_ConnectComponents` werden im Zusammenhang mit dem Erzeugen des Modells implementiert. In der Methode `build` werden die Komponenten erzeugt und die Parameter gesetzt:

```

procedure TPMVolCompartment.build;
begin
    FParams.SetParams([SCCompCompliance+self.CompartmentName,
                      SCCompResistance+Self.CompartmentName,
                      SCCompPercentage+Self.CompartmentName],
                      [dflt_CompCompliance,
                      dflt_CompResistance,
                      dflt_CompPercentage],
                      [SCUnitCompliance,SCUnitResistance,SCUnitPercentage]);

    FBoolParams.SetParams([SCWithAugmentation],
                           [dflt_WithAugmentation],
                           [SCUnitBoolean]);
```

```

FDependentValues.setParams([SCCompPressure+self.CompartementName],
                           [GetPCompartement],
                           [SCUnitPressure]);

```

```

FLevels.SetParams([SCVolCompartement+self.CompartementName],
                 [dflt_StvCompartement],
                 [SCUnitVolume]);

```

```

FStartValues.SetParams([SCVolCompartement+self.CompartementName],
                      [dflt_StvCompartement],
                      [SCUnitVolume]);

```

```

FRates.SetParams([SCFlwVolCompartement+Self.CompartementName],
                 [NoFlow],
                 [SCUnitVolFlw]);

```

```

{
  Werte, die aggregiert werden sollen
}

```

```

FBreathAggregateValues.SetValueLists(
    [SCCompPressure+self.CompartementName,
    SCVolCompartement+self.CompartementName],
    [SCUnitPressure,SCUnitVolume],
    [AGG_Mean,AGG_Mean]);

```

```

FMinAggregateValues.SetValueLists(
    [SCCompPressure+self.CompartementName,
    SCVolCompartement+self.CompartementName],
    [SCUnitPressure,SCUnitVolume],
    [AGG_Mean,AGG_Mean]);

```

```

{
  Werte, die gespeichert werden sollen
}

```

```

FStoredValues.SetValueLists(
    [SCVolCompartement+self.CompartementName,
    SCFlwVolCompartement+self.CompartementName],
    [SCUnitVolume,SCUnitVolFlw]);

```

```

{
  Schnittstellenabbildung
}

```

```

    FAlveole      := nil;
    FLungPointer  := nil;
    FTracheaPointer := nil;
end;

```

Die `InitModel_ConnectComponents`-Methode dient der Implementation der Schnittstellenabbildung aus der Definition eines modularen L/R-Modells 3.13. Es werden alle Zeiger auf verwendete Modellkomponenten gesetzt. Hier handelt es sich um Trachea und Lunge.

```

procedure TPMVolCompartment.InitModel_ConnectComponents;
begin
    FTracheaPointer
        :=(FMain.PhysModel.Component[SCVolTrachea] as TPMTrachea);

    if assigned(FMain.PhysModel.Component[SCPMLung])
        then FLungPointer
            := (FMain.PhysModel.Component[SCPMLung] as TPMLung)
        else BoolParams.Value[SCWithAugmentation] := false;
end;

```

Die Simulationsinitialisierung

Zur Initialisierung der Simulation sind die Standard-Startwerte und die für eine Simulation konstanten übrigen Parameter zu setzen. Ersteres geschieht durch `SetDefaultStartValues`. In diesem Fall ist der Standard-Startwert gerade das theoretische Mindestvolumen im Kompartiment aus dem Produkt von Compliance und PEEP. Die übrigen Parameter werden mit Hilfe der Methode `InitSimulation_SetParam` festgelegt.

```

procedure TPMVolCompartment.SetDefaultStartValues;
begin
    StartValues.Value[SCVolCompartment +
        self.CompartementName]
        :=self.FParams.Value[SCCompCompliance +Self.CompartementName]*
        FMain.PhysModel.ComponentByVariable[SCPEEP].Params.Value[SCPEEP];
end;

```

```

procedure TPMVolCompartment.InitSimulation_SetParam;
begin
    FWithAugmentation := BoolParams.Value[SCWithAugmentation];

```

```

    FCompliance      := self.FParams.Value[SCCompCompliance +
                                           Self.CompartementName];
    FPercentage      := self.FParams.Value[SCCompPercentage +
                                           self.CompartementName];
end;

```

Die Simulationsdurchführung

Zunächst werden die verwendeten privaten Routinen der Klasse beschrieben. Aus dem aktuellen Zustand und der Compliance muss der elastische Druck berechnet werden.

```

function TPMVolCompartement.GetPCompartement : Double;
begin
    result := FLevels.ValueByIndex[0]/FCompliance;
end;

```

Aus der Druckdifferenz wird anschließend der Fluss berechnet.

```

function TPMVolCompartement.GetFlwVolCompartement : Double;
var
    PressureIn, PressureLocal,ResistanceLocal : Double;
begin
    pressureIN      := FTracheaPointer.DependentValues.Value[SCPTrachea];
    pressureLocal   := GetPCompartement;
    if FWithAugmentation
    then PressureLocal := PressureLocal -
         FLungPointer.DependentValues.Value[SCDeltaPAugmentation];

    resistanceLocal := self.FParams.Value
                     [SCCompResistance+Self.CompartementName];

    result := PressureFlow(ResistanceLocal,PressureIn,PressureLocal);
end;

```

Die zugehörigen öffentlichen Routinen

Diese Methoden werden innerhalb der öffentlichen FlowAddition-Routine verwendet. Sie bestimmt die Veränderungen der Zustände der Level. Dabei dient die Funktion TotalFlow der Berechnung der Differenz aus den beiden übergebenen Feldern. Man beachte, dass hier die Modellierung des Sauerstoffs und des Kohlendioxids noch nicht stattgefunden hat, so dass deren Einflüsse noch nicht berücksichtigt sind.

```

procedure TPMVolCompartement.FlowAddition;
begin
    DeltaStates[0] := TotalFlow([FRates.ValueByIndex[0]], [NoFlow]);
end;

```

Die `CalculateRates`-Routine berechnet die aktuellen Werte der in der Klasse vorhandenen Raten. In diesem Fall besteht es nur darin, den Fluss des Volumens ins Kompartiment zu bestimmen.

```
procedure TPMVolCompartment.CalculateRates;
begin
  FRates.ValuebyIndex[0] := GetFlwVolCompartment;
end;
```

E.1.3 Die Anwendung der Klasse

Angewandt wird die Klasse, indem ein entsprechendes Objekt in der `build`-Routine der Klasse `TPMModel` erzeugt wird und entsprechende globale Parameter, wie etwa die relative Größe des Kompartimentes, gesetzt werden.

Im folgenden Auszug aus dem Quellcode werden die Kompartimente „Normal“ und „Traum“ erzeugt und ihre Parameter gesetzt. Bei den Kompartimenten handelt es sich um Instanzen der Klasse `TPMCompletCompartment`, in die die Modellierung des Sauerstoffs und des Kohlendioxids integriert sind und die von der Klasse `TPMVolCompartment` abgeleitet sind.

```
For i:=0 to 1 do begin
  LokCompName := FPMDesigner.CBVolCompNames.items[i];

  ComponentLocal := TPMCompletCompartment.create(EmptyStr);
  with ComponentLocal do begin
    Name := SCVentCompartment+LokcompName;
    CompartmentName :=LokCompName;

    FCompartmentNames[0] := LokCompName;
    Build;

    if LokCompName=SCCompDeadspace then begin
      StartValues.Value[SCVolCompartment+CompartmentName] :=
        dflt_StvVolDeadspace;
      StartValues.Value[SCO2Compartment +CompartmentName] :=
        dflt_StvO2Deadspace;
      StartValues.Value[SCCO2Compartment+CompartmentName] :=
        dflt_StvCO2Deadspace;

      Params.Value[SCCompPercentage+ComponentLocal.CompartmentName]
        :=dflt_DeadspacePercentage;
    end;

    if LokCompName=SCCompNormal then begin
      StartValues.Value[SCVolCompartment+CompartmentName] :=
```

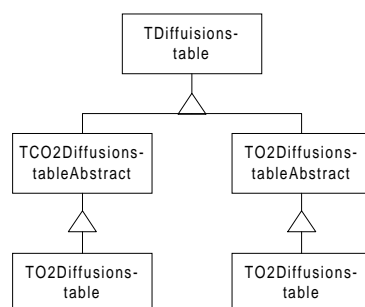


Abbildung E.1: Vererbungsstruktur für die Bindungskurven

```

        dflt_StvVolNormal;
    StartValues.Value[SCO2Compartment +CompartmentName] :=
        dflt_StvO2Normal;
    StartValues.Value[SCCO2Compartment+CompartmentName] :=
        dflt_StvCO2Normal;
    Params.Value[SCCompPercentage+ComponentLocal.CompartmentName]
        :=dflt_NormalPercentage;
    end;
end;
FComponents.add(ComponentLocal);
end; // for i vent-compartmentes

```

In ähnlicher Weise können weitere Modellkomponenten entwickelt und in das physiologische Modell integriert werden.

E.2 Die Bindungskurven

Abbildung E.1 zeigt die erstellte Vererbungsstruktur. Da alle Bindungskurven den Gehalt aus dem Druck und umgekehrt berechnen können müssen, ist eine zugehörige abstrakte Basisklasse implementiert worden. Weitere abstrakte Klassen unterscheiden zwischen dem Kohlendioxid und dem Sauerstoff. Die Klasse für die Sauerstoff-Bindungskurve enthält zusätzlich Methoden, um den Zusammenhang zwischen Sättigung und Druck zu berechnen. Schließlich werden konkrete Bindungskurven implementiert.

E.2.1 Kohlendioxid-Bindungskurve

Im Fall des Kohlendioxids sind im funktionalen Prototypen (vgl. Abschnitt 5.2.1) zwei verschiedene Bindungskurven implementiert worden: Es ist dies zunächst eine Tabelle, die auf der aus [Nun87] S. 213 entnommenen Abbildung E.2 beruht. Darüber hinaus ist ein Algorithmus von Kelman realisiert worden, der hier nach [WW77] zitiert wird. Es hat sich gezeigt, dass die Ergebnisse der beiden Methoden sich kaum unterscheiden, so dass der einfacheren Tabellen-Funktion, die auf physiologischen Erfahrungswerten beruht, der Vorzug gegeben worden ist.

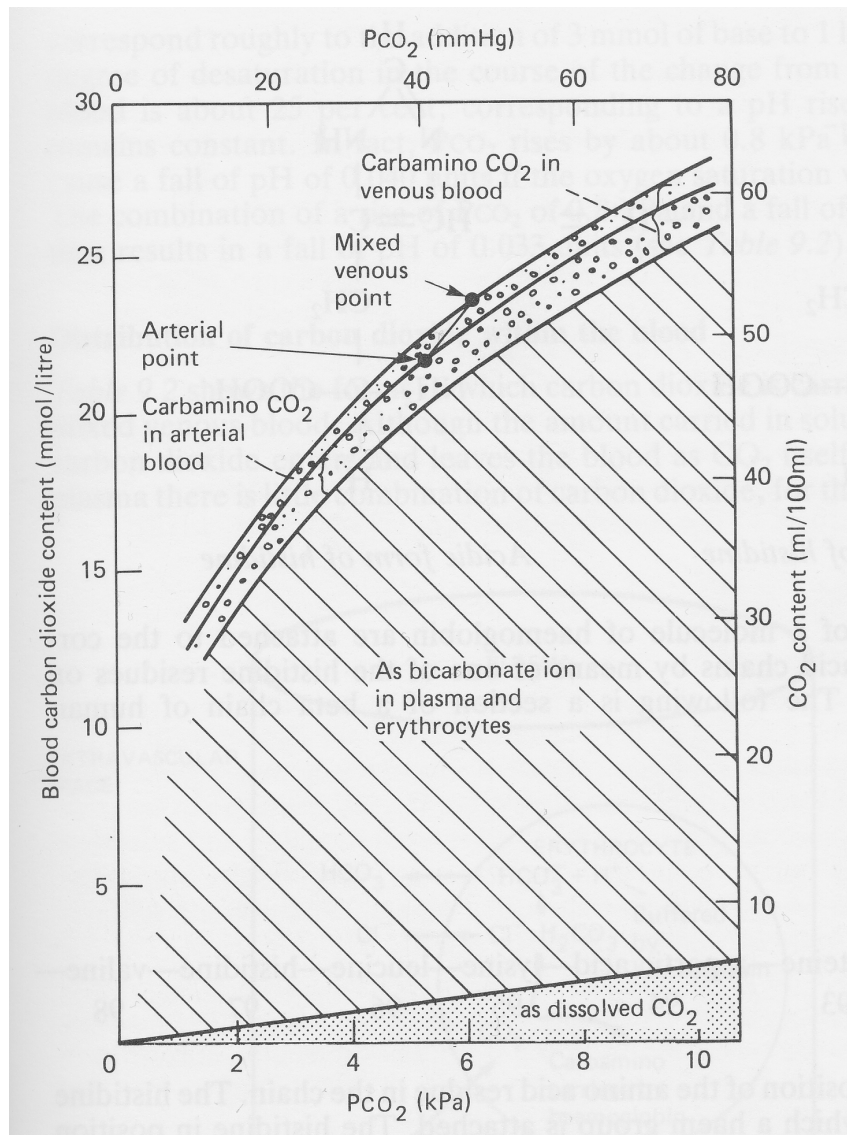


Abbildung E.2: Kohlendioxid-Bindungskurve

Tabellen-Funktion. In der folgenden Tabelle über den Zusammenhang des arteriellen Partialdruck des Kohlendioxids $P_a\text{CO}_2$ und des zugehörigen Gehalts $C_a\text{CO}_2$ sind die Stützstellen aus Abbildung E.2 abgelesen worden. Verlässt der Gehalt den Bereich der Tabelle, so wird auf den entsprechenden Extremwert abgeschnitten. Zwischen den Stützstellen wird linear interpoliert.

| $P_a\text{CO}_2$ [mmHG] | $C_a\text{CO}_2$ [ml/ml] |
|-------------------------|--------------------------|
| 10 | 0,282 |
| 20 | 0,366 |
| 30 | 0,435 |
| 40 | 0,482 |
| 50 | 0,521 |
| 60 | 0,559 |
| 70 | 0,588 |
| 80 | 0,612 |
| 90 | 0,621 |
| 100 | 0,639 |
| 110 | 0,654 |

Tabelle E.1: Kohlendioxid-Bindungskurve als Tabelle

Formel nach West. Der Algorithmus von Kelman berechnet den Kohlendioxid-Gehalt C_{CO_2} aus dem Partialdruck $P_a\text{CO}_2$, dem pH-Wert des Plasmas, der jeweilige Sauerstoffsättigung S_{O_2} , dem Hämatokritwert $Haema$ und der Temperatur $Temp$. Als Näherung für den Hämatokritwert wird das dreifache des bei der Blutgasanalyse festgestellten Hämoglobinwerts verwendet. Die Temperatur wird mit 37 Grad Celsius abgeschätzt.

1. **pK-Wert** pK :

$$pK = 6.086 + 0,042(7,4 - \text{pH}) + (38 - \text{Temp}) \left(0,0047 + 0,0014(7,4 - \text{pH}) \right)$$

2. **CO₂-Löslichkeit** $SOL_{C_{\text{CO}_2}}$:

$$SOL_{C_{\text{CO}_2}} = 0,0307 + 0,00057(37 - \text{Temp}) + 0,00002(37 - \text{Temp})^2$$

3. **CO₂-Gehalt im Plasma** $C_{\text{CO}_2}^{\text{Plasma}}$:

$$C_{\text{CO}_2}^{\text{Plasma}} = SOL_{C_{\text{CO}_2}} * P_a\text{CO}_2 \left(1 + 10^{(\text{pH} - pK)} \right)$$

4. **Verhältnis von CO₂-Gehalt in Zellen und Plasma** D_{mix}

Hierzu wird dieses Verhältnis zunächst für komplett oxigeniertes Blut (D_{oxy}), anschließend für völlig reduziertes Blut (D_{red}) bestimmt, um schließlich in Abhängigkeit von der Sättigung $S_a\text{O}_2$ zwischen beiden linear zu interpolieren und das gewünschte Verhältnis D_{mix} zu erhalten:

- (a) $D_{\text{oxy}} = 0,59 + 0,2913(7,4 - \text{pH}) - 0,0844(7,4 - \text{pH})^2$
 (b) $D_{\text{red}} = 0,664 + 0,2275(7,4 - \text{pH}) - 0,0938(7,4 - \text{pH})^2$
 (c) $D_{\text{mix}} = D_{\text{oxy}} + (D_{\text{red}} - D_{\text{oxy}}) \left(1 - \frac{S_{\text{aO}_2}}{100}\right)$

5. **CO₂-Gehalt in den Zellen** $C_{\text{CO}_2}^{\text{Zellen}}$:

$$C_{\text{CO}_2}^{\text{Zellen}} = D_{\text{mix}} * C_{\text{CO}_2}^{\text{Plasma}}$$

6. **CO₂-Gehalt im Blut** C_{CO_2} :

$$C_{\text{CO}_2} = \left(\text{Haema} * C_{\text{CO}_2}^{\text{Zellen}} + (1 - \text{Haema}) C_{\text{CO}_2}^{\text{Plasma}} \right) * 2,22$$

Der Faktor 2,22 dient der Umrechnung der Einheit mmol/l in ml/100 ml.

Der beschriebene Algorithmus wird in den Kapillargefäßen der Kompartimente verwendet, wo aus dem herrschenden Partialdruck auf den Gehalt geschlossen werden muss. In den Arterien benötigt man dessen Umkehrung, da sich dort der Gehalt C_{CO_2} aus der Simulation ergibt und auf den herrschenden Partialdruck geschlossen werden muss. Zunächst werden dazu, wie zuvor beschrieben, die Variablen D_{oxy} , D_{red} und D_{mix} bestimmt. In diese Berechnung geht die Sauerstoffsättigung ein. Diese wird für die Arterien berechnet unter Vernachlässigung des physikalisch gelösten Sauerstoffs näherungsweise als:

$$\widetilde{S_{\text{aO}_2}} = \frac{C_{\text{aO}_2}}{\text{Hb} * 0,0131}.$$

Anschließend werden die Löslichkeit des Kohlendioxids im Plasma, der pK-Wert, der Gehalt des Kohlendioxids im Plasma und schließlich daraus der Partialdruck bestimmt:

$$\begin{aligned} \text{Loes}_{\text{Plasma}}^{CO_2} &= 0,0307 + 0,00057 * (37 - \text{Temp}) + 0,00002 * (37 - \text{Temp})^2, \\ \text{pK} &= 0,086 + 0,042 * (7,4 - \text{pH}) + \\ &\quad (38 - \text{Temp}) * (0,0047 + 0,0014 * (7,4 - \text{pH})), \\ C_{\text{Plasma}}^{CO_2} &= \frac{100 * C_{\text{CO}_2}}{(\text{Haema} * D_{\text{mix}} + (1 - \text{Haema})) * 2,22}, \\ P_{\text{aCO}_2} &= \frac{C^{CO_2}}{\text{Loes}_{\text{Plasma}}^{CO_2} * (1 + 10^{\text{pH} - \text{pK}})}. \end{aligned}$$

E.2.2 Sauerstoff-Bindungskurve

Bei der Implementation der Sauerstoff-Bindungskurve handelt es sich um eine Operationalisierung der Tabelle E.3 aus [Nun87] S. 511. Diese wird folgendermaßen verwendet:

1. **Berechnung des virtuellen PO₂**: Aus Tabelle E.2 werden zu gegebener Temperatur, pH-Wert und Basenüberschuss BE die zugehörigen Faktoren ermittelt und miteinander multipliziert. (Zwischen zwei Stützstellen wird linear interpoliert.) Der sich daraus ergebende Korrekturfaktor wird mit dem gemessenen PO₂ multipliziert, so dass ein *virtueller* PO₂ entsteht.

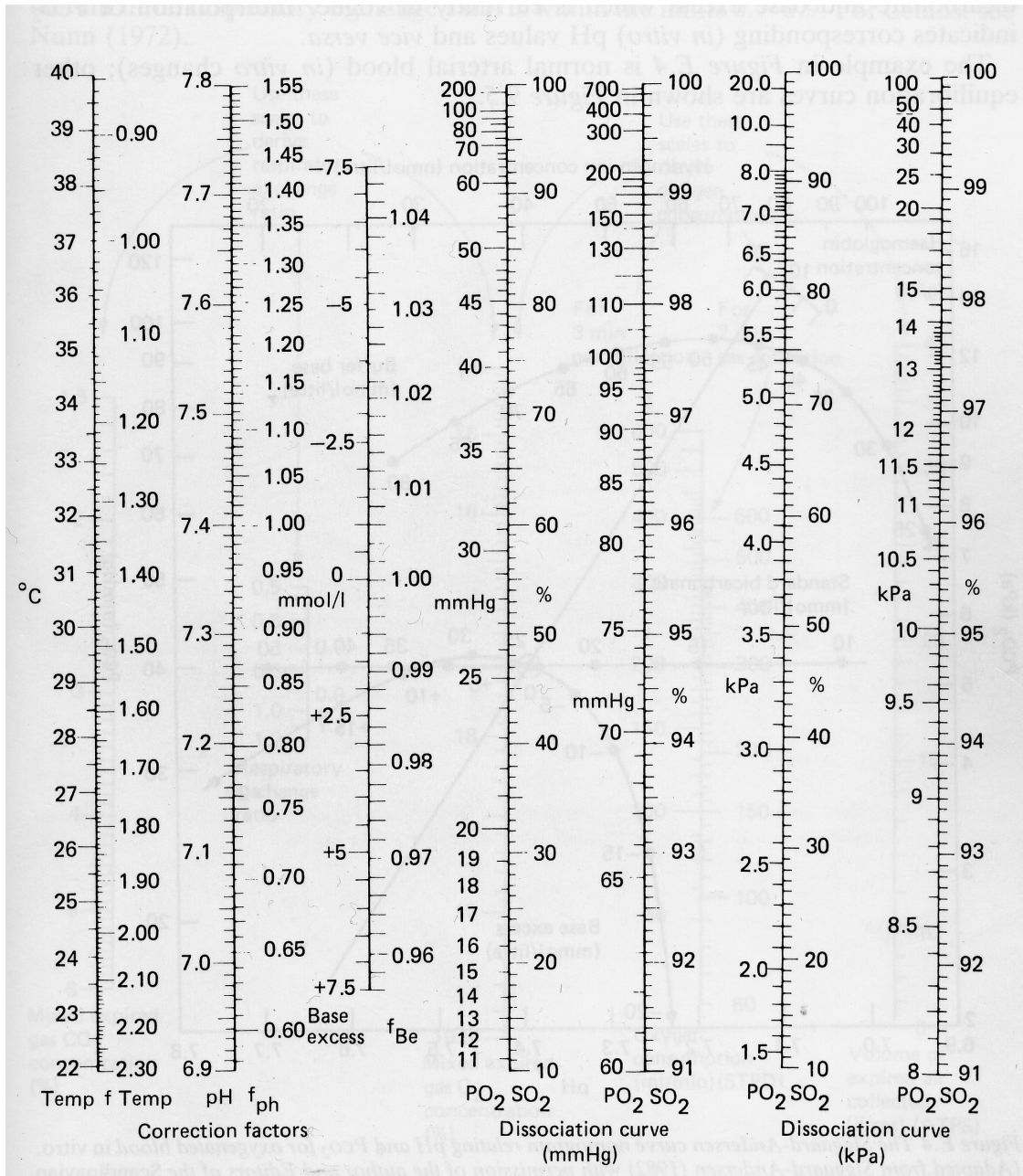


Abbildung E.3: Sauerstoff-Bindungskurve als mehrdimensionale Tabelle

| Temperatur | Faktor | pH-Wert | Faktor | BE | Faktor |
|------------|--------|---------|--------|------|--------|
| 22 | 2,29 | 6,9 | 0,58 | -7,5 | 1,046 |
| 23 | 2,17 | 6,95 | 0,61 | -5 | 1,031 |
| 24 | 2,05 | 7 | 0,64 | -2,5 | 1,015 |
| 25 | 1,94 | 7,05 | 0,68 | 0 | 1 |
| 26 | 1,84 | 7,1 | 0,72 | 2,5 | 0,986 |
| 27 | 1,74 | 7,15 | 0,76 | 5 | 0,971 |
| 28 | 1,65 | 7,2 | 0,8 | 7,5 | 0,964 |
| 29 | 1,55 | 7,25 | 0,85 | | |
| 30 | 1,47 | 7,35 | 0,9 | | |
| 31 | 1,39 | 7,4 | 0,94 | | |
| 32 | 1,32 | 7,45 | 1 | | |
| 33 | 1,25 | 7,5 | 1,06 | | |
| 34 | 1,18 | 7,55 | 1,12 | | |
| 35 | 1,12 | 7,6 | 1,18 | | |
| 36 | 1,06 | 7,65 | 1,25 | | |
| 37 | 1,00 | 7,7 | 1,39 | | |
| 38 | 0,95 | 7,75 | 1,47 | | |
| 39 | 0,91 | 7,8 | 1,55 | | |
| 40 | 0,84 | | | | |

Tabelle E.2: Die Sauerstoff-Bindungskurve beeinflussende Faktoren

2. **Ablesen der Sättigung:** Zu dem berechneten virtuellen PO_2 wird in der Tabelle E.3 die Sättigung abgelesen.
3. **Berechnung des Gehalts:** Der Gehalt des Sauerstoffs ergibt sich nach der Formel:

$$C_{aO_2} = [Hb] * S_{aO_2} * 0,131 + 0,0000317 * P_{aO_2}. \quad (E.1)$$

Für die Umkehrung wird ein iterativer Algorithmus benötigt, da der Gehalt sowohl von der Sättigung als auch vom Partialdruck abhängt, wie Gleichung (E.1) zeigt.

| PO ₂ | SO ₂ | PO ₂ | SO ₂ | PO ₂ | SO ₂ |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| 11 | 11 | 43 | 78,1 | 80 | 95,85 |
| 12 | 13 | 44 | 79,2 | 81 | 95,99 |
| 13 | 15 | 45 | 80,2 | 82 | 96,1 |
| 14 | 17 | 46 | 81,3 | 83 | 96,21 |
| 15 | 19 | 47 | 82,2 | 84 | 96,31 |
| 16 | 21.5 | 48 | 83,1 | 85 | 96,4 |
| 17 | 24 | 49 | 84 | 86 | 96,5 |
| 18 | 26.5 | 50 | 85 | 87 | 96,6 |
| 19 | 29 | 52 | 86,7 | 88 | 96,68 |
| 20 | 32 | 54 | 87,8 | 89 | 96,78 |
| 21 | 35 | 56 | 88,8 | 90 | 96,88 |
| 22 | 38 | 58 | 90 | 91 | 96,95 |
| 23 | 40.5 | 60 | 91 | 92 | 97,02 |
| 24 | 43.2 | 61 | 91,35 | 93 | 97,09 |
| 25 | 46 | 62 | 91,72 | 94 | 97,16 |
| 26 | 48.5 | 63 | 92,1 | 95 | 97,23 |
| 27 | 51 | 64 | 92,43 | 96 | 97,31 |
| 28 | 53.2 | 65 | 92,79 | 97 | 97,37 |
| 29 | 55.5 | 66 | 93,08 | 98 | 97,42 |
| 30 | 58 | 67 | 93,35 | 99 | 97,48 |
| 31 | 59.5 | 68 | 93,65 | 100 | 97,54 |
| 32 | 61.5 | 69 | 93,9 | 105 | 97,78 |
| 33 | 63.2 | 70 | 94,32 | 110 | 98 |
| 34 | 65 | 71 | 94,41 | 120 | 98,25 |
| 35 | 67 | 72 | 94,54 | 130 | 98,5 |
| 36 | 68.3 | 73 | 94,73 | 140 | 98,69 |
| 37 | 70.1 | 74 | 94,88 | 150 | 98,8 |
| 38 | 71.5 | 75 | 95,03 | 200 | 99,2 |
| 39 | 73.1 | 76 | 95,21 | 300 | 99,56 |
| 40 | 74.4 | 77 | 95,4 | 400 | 99,71 |
| 41 | 75,8 | 78 | 95,57 | 600 | 99,9 |
| 42 | 77 | 79 | 95,71 | | |

Tabelle E.3: Die Sauerstoff-Bindungskurve

Literaturverzeichnis

- [Bal96] BALZERT, HELMUT: *Lehrbuch der Software Technik: Software-Entwicklung*. Spektrum Akademischer Verlag, Heidelberg, Berlin, 1996.
- [Bal98] BALZERT, HELMUT: *Lehrbuch der Software Technik: Software-Management, Software-Qualitätssicherung, Unternehmensmodellierung*. Spektrum Akademischer Verlag, Heidelberg, Berlin, 1998.
- [Bal99] BALZERT, HEIDE: *Lehrbuch der Objektmodellierung: Analyse und Entwurf*. Spektrum Akademischer Verlag, Heidelberg, Berlin, 1999.
- [Bau90] BAUMGARTEN, B.: *Petri-Netze: Grundlagen und Anwendungen*. BI-Wissenschaftlicher Verlag, Mannheim u.a., 1990.
- [BBB⁺83] BUCHANAN, B.G., D. BARSTOW, R. BECHTAL, J. BENNETT, W. CLANCEY, C. KULIKOWSKY und T. MITCHELL und D. WATERMAN: *Constructing an Expert System*, Seiten 127 – 167. Addison Wesley, London, 1983.
- [BG93] BOMZE, I.M. und W. GROSSMANN: *Optimierung — Theorie und Algorithmen*. BI-Wissenschaftsverlag, Mannheim, Leipzig, Wien, Zürich, 1993.
- [BH99] BERTHOLD, M. und D.J. HAND: *Intelligent Data Analysis — An Introduction*. Springer, Berlin, Heidelberg, New York u.a., 1999.
- [BL97] BERRY, M.J.A. und G. LINOFF: *Data Mining Techniques: For Marketing, Sales, and Customer Support*. John Wiley & Sons, New York, Chichester, 1997.
- [Bos94] BOSSEL, H.: *Modellbildung und Simulation*. Vieweg, Braunschweig, Wiesbaden, zweite Auflage, 1994.
- [Bro86] *Brockhaus Enzyklopädie in 24 Bänden*. Mannheim, 1986. 19. Auflage.
- [BW88] BRUNNER, J.X. und G. WOLFF: *Pulmonary Function Indices in Critical Care Patients*. Springer, Berlin, Heidelberg, New York u.a., 1988.
- [CBL73] COLES, J.R., W.A. BROWN und D.G. LAMPARD: *Computer control of respiration and anaesthesia*. Medical Biological Engineering, Seiten 262–267, 1973.

- [CGU97] CHIARI, L., G.AVANZOLINI und M. URSINO: *A Comprehensive Simulator of the Human Respiratory System: Validation with Experimental and Simulated Data*. *Annals of Biomedical Engineering*, 25, 1997.
- [Cle99] CLEMEN, T.: *Zur Wavelet-gestützten Validierung von Simulationsmodellen in der Ökologie*. Doktorarbeit, Christian-Albrechts-Universität Kiel, 1999.
- [CY91a] COAD, P. und E. YOURDON: *Object oriented analysis*. NJ: Yourdon Press, Englewood Cliffs, zweite Auflage, 1991.
- [CY91b] COAD, P. und E. YOURDON: *Object oriented design*. NJ: Yourdon Press, Englewood Cliffs, 1991.
- [Dic77] DICKINSON, C.J.: *A Computer Model of Human Respiration*. MTP, London, 1977.
- [Drä97] DRÄGERWERK, AG: *Gebrauchsanweisung EVITA 4 Intensivtherapie-Ventilator*. Lübeck, 1997.
- [Fal77] FALTER, J.W.: *Zur Validierung theoretischer Konstrukte — Wissenschaftstheoretische Aspekte des Validierungskonzepts*. *Zeitschrift für Soziologie*, (Heft 4), Oktober 1977.
- [For72] FORRESTER, J.W.: *Grundsätze einer Systemtheorie*. Betriebswirtschaftlicher Verlag Dr. Th. Gabler, Wiesbaden, 1972.
- [Fre85] FREILING, M.: *Starting a Knowledge Engineering Project: A Step-by-Step Approach*. *AI Magazine*, Fall 1985.
- [FS98] FOWLER, M. und K. SCOTT: *UML konzentriert*. Addison-Wesley, Bonn, Reading, Massachusetts u.a., 1998.
- [GGB67] GRODINS, F.S., J. BUELL und A.J. BART: *Mathematical analysis and digital simulation of respiratory control system*. *Appl. Physiol.*, 22:260–276, 1967.
- [GC86] GRUBER, TH. und P. COHEN: *Principles of Knowledge System Design to Facilitate Knowledge Acquisition*. In: J. BOOSE, B. GAINES (Herausgeber): *AAAI Workshop: Knowledge Acquisition for Knowledge Based Systems*, Banff, 1986.
- [GKV82] GERTHSEN, C., KNESER und H. VOGEL: *Physik — Ein Lehrbuch zum Gebrauch in Vorlesungen*. Springer, Berlin, Heidelberg, New York, 14. Auflage, 1982.
- [Hay94] HAYKIN, S.: *Neural Networks: A Comprehensive Foundation*. Macmillan College Publishing Company, New York, 1994.
- [HJS83] H.-J. SCHRÖDER, U. KNITELIUS: *Strukturelle Analyse und Implementation interpretierter Petri-Netz-Vergrößerungen*. Diplomarbeit, Universität Osnabrück - Fachbereich 6: Mathematik, 1983.

- [HN91] HECHT-NIELSEN, R.: *Neurocomputing*. Addison-Wesley, Reading, Menlo Park, New York u.a., 1991.
- [HrW87] HAIRER, E., S.P. NØRSETT und G. WANNER: *Solving Ordinary Differential Equations I*. Springer, Berlin, Heidelberg, New York u.a., 1987.
- [KL90] KARBACH, W. und M. LINSTER: *Wissensakquisition für Expertensysteme*. Carl Hanser, München, Wien, 1990.
- [KPS83] KNITELIUS, U., J. PERL und H.-J. SCHRÖDER: *Implementierung interpretierter Netzvergrößerungen*. Osnabrücker Schriften zur Mathematik, (Heft 13), 1983.
- [MP00] MESTER, J. und J. PERL: *Grenzen der Anpassungs- und Leistungsfähigkeit des Menschen aus systematischer Sicht*. Leistungssport, 1, 2000.
- [Nol98] NOLL, O.: *Entwicklung eines Level-Raten-Metamodells und exemplarische Anwendung für die Modellierung von Leistungspotentialen*. Diplomarbeit, Johannes Gutenberg-Universität Mainz - Fachbereich 17: Mathematik, 1998.
- [Nun87] NUNN, J.F.: *Applied Respiratory Physiology*. Butterworths, London, Boston u.a., dritte Auflage, 1987.
- [OE98] ORR, G.B. und K.-R. MÜLLER (EDS.): *Neural Networks: Tricks of the Trade*. Springer, Berlin, Heidelberg, New York u.a., 1998.
- [OWA94] OCZENSKI, W., A. WERBA und H. ANDEL: *Atmen Atemhilfen*. Blackwell MZV, Wien, erste Auflage, 1994.
- [OWA96] OCZENSKI, W., A. WERBA und H. ANDEL: *Atmen Atemhilfen*. Blackwell Wissenschaft-Verlag, Berlin, Wien, zweite, erweiterte Auflage, 1996.
- [PB93] POMBERGER, G. und G. BLASCHEK: *Software Engineering: Prototyping und objektorientierte Software-Entwicklung*. Carl Hanser, München, Wien, 1993.
- [PBK⁺91] POMBERGER, G., W. BISCHOFBERGER, D. KOLB, W. PREE und H. SCHLEMM: *Prototyping-Oriented Software Development - Concepts and Tools*. Structured Programming, 12, 1991.
- [Per81] PERL, J.: *Graphentheorie: Grundlagen und Anwendungen*. Akademische Verlags-Gesellschaft, Wiesbaden, 1981.
- [Per97] PERL, J.: *Möglichkeiten und Probleme der computerunterstützten Interaktionsanalyse am Beispiel Handball*. In: PERL, J. (Herausgeber): *Sport und Informatik V: Bericht über den 5. Workshop Sport und Informatik*, 1997.
- [Per99] PERL, J.: *Interaktiver graphischer Modell-Generator mit Präsentationseinheit für on line-Simulation*. In: HOHMANN, G. (Herausgeber): *Simulationstechnik: 13. Symposium in Weimar*, 1999.

- [PFTV88] PRESS, W.H., B.P. FLANNERY, S.A. TEUKOLSKY und W.T. VETTERLING: *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, Cambridge, New York u.a., dritte Auflage, 1988.
- [PGPB96] PUPPE, F., U. GAPPA, K. POECK und S. BAMBERGER: *Wissensbasierte Diagnose- und Informationssysteme*. Springer, Berlin, Heidelberg, New York, 1996.
- [Pic75] PICHLER, F.: *Mathematische Systemtheorie*. De Gruyter Lehrbuch, Berlin, New York, 1975.
- [PLG02] PERL, J., M. LAMES und U. GLITSCH: *Modellbildung in der Sportwissenschaft*. Verlag Karl Hofmann, Schorndorf, 2002. im Druck.
- [PLM97] PERL, J., M. LAMES und W.-D. MIETHLING: *Informatik im Sport: Ein Handbuch*. Verlag Karl Hofmann, Schorndorf, 1997.
- [Pop94] POPPER, K.R.: *Logik der Forschung*. J.C.B. Mohr (Paul Siebeck), Tübingen, zehnte, verbesserte und vermehrte Auflage, 1994.
- [PU92] POLANI, D. und T. UTHMANN: *Neuronale Netze: Grundlagen und ausgewählte Aspekte*. Mainzer Informatik-Bericht, 1992.
- [Pup88] PUPPE, F.: *Einführung in Expertensysteme*. Springer, Berlin, Heidelberg, New York u.a., 1988.
- [Rei82] REISIG, W.: *Petrinetze*. Springer, Berlin, Heidelberg, New York, 1982.
- [Rei85] REISIG, W.: *Systementwurf mit Netzen*. Springer, Berlin, Heidelberg, New York, Tokyo, 1985.
- [Rum93] RUMBAUGH, J.: *Objektorientiertes Modellieren und Entwerfen*. Hanser Verlag, München u.a., 1993.
- [Sau99] SAUERBIER, T.: *Theorie und Praxis von Simulationssystemen*. Vieweg, Braunschweig, Wiesbaden, 1999.
- [SB96] STOER, J. und R. BURLISCH: *Numerische Mathematik I*. Springer, Berlin, Heidelberg, New York, zweite, erweiterte Auflage, 1996.
- [Sch97a] SCHÖNING, U.: *Algorithmen — kurz gefasst*. Spektrum Akademischer Verlag, Heidelberg, Berlin, 1997.
- [Sch97b] SCHÖNING, U.: *Theoretische Informatik — kurz gefasst*. Spektrum Akademischer Verlag, Heidelberg, Berlin, 3. Auflage, 1997.
- [SG00] SEMMEL-GRIEBELER, TH.: *EWS ein wissensbasierter „Weaning“ Assistent*. Mainzer Informatik-Bericht: Oberseminar Informatik SS 1998, WS 1998/99, SS 1999, 2000.

- [SGUW⁺99] SEMMEL-GRIEBELER, TH., TH. UTHMANN, N. WEILER, J. PERL und W. HEINRICHS: *A Knowledge-Based Weaning Assistant for Controlling an Artificial Respirator*. In: *AIMDM '99: Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making*, 1999.
- [ST90] SCHMIDT, R. und G. THEWS: *Physiologie des Menschen*. Springer-Lehrbuch, Berlin, Heidelberg, New York u.a., 24. Auflage, 1990.
- [Uth94] UTHMANN, T.: *Wissensbasierte Informationssysteme: Konzepte und Realisierungen*. unveröffentliche Habilitationsschrift, Mainz 1994.
- [Wal96] WALTER, W.: *Gewöhnliche Differentialgleichungen*. Springer, Berlin, Heidelberg, New York u.a., sechste Auflage, 1996.
- [Wei98] WEILER, N.: *Experimentelle und klinische Untersuchungen zur automatischen Entwöhnung des Patienten vom Respirator mit Hilfe der Adaptive Lung Ventilation (ALV)*. unveröffentliche Habilitationsschrift, Mainz 1998.
- [WW77] WEST, J.B und P.D. WAGNER: *Pulmonary Gas Exchange*. In: DEKKER, MARCEL (Herausgeber): *Bioengineering Aspects auf the Lung*, 1977.
- [Zei90] ZEIGLER, B.P.: *Object-Oriented Simulation with Hierarchical, modular Models*. Academic Press, Inc., Boston, San Diego, New York u.a., 1990.
- [Zel94] ZELL, A.: *Simulation neuronaler Netze*. Addison-Wesley, Bonn, Paris, Reading, Massachusetts u.a., 1994.
- [ZPK00] ZEIGLER, B.P., H. PRAEHOFER und T.G. KIM: *Theory of Modeling and Simulation*. Academic Press, San Diego, San Francisco u.a., zweite Auflage, 2000.
- [Zum77] ZUMKLEY, H.: *Klinik des Wasser-, Elektrolyt- und Säure-Basen-Haushalts*. Georg Thieme Verlag, Stuttgart, 1977.

Abbildungsverzeichnis

| | | |
|------|--|----|
| 1 | Das allgemeine Koordinatensystem der Modellbildung | 2 |
| 1.1 | Systeme als Blackbox | 11 |
| 1.2 | Systeme als Blackbox mit Experimentierfeld | 12 |
| 1.3 | Ein modulares System mit zwei Komponenten | 16 |
| 1.4 | Codierer und Decodierer zweier Systeme | 17 |
| 2.1 | Schema der zeitorientierten und ereignisorientierten Simulation | 24 |
| 2.2 | Simulation zur Prognose und Entscheidungshilfe | 25 |
| 2.3 | Der Übergang vom System zum Simulationsmodell | 26 |
| 2.4 | Der Standardentwurf übertragen auf die Modellbildung | 26 |
| 2.5 | Das Grundmodell der Modellbildung | 27 |
| 2.6 | Das Ebenenmodell der Modellbildung | 28 |
| 2.7 | Software-Lebenszyklus: Das V-Modell | 28 |
| 2.8 | Software-Lebenszyklus: Das Spiralmodell | 29 |
| 3.1 | Die in Level-Raten-Diagramme verwendeten Symbole und deren Bedeutung | 38 |
| 4.1 | Der zyklische Prozess der Theoriebildung | 44 |
| 4.2 | Abstraktion in der Theoriebildung | 45 |
| 4.3 | Das Koordinatensystem der Modellbildung als dessen Grundstruktur | 46 |
| 4.4 | Das Verhältnis von Objekt und Konstrukt nach Falter | 47 |
| 4.5 | Das Koordinatensystem für begriffliche Objekte | 48 |
| 5.1 | Expertensysteme und dynamische Systeme | 52 |
| 5.2 | Der Zyklus der prototypischen Wissensakquisition | 52 |
| 5.3 | Stadien der Modellierung in KADS | 53 |
| 5.4 | Die Assoziation zweier Klassen | 55 |
| 5.5 | Die Aggregation zweier Klassen | 55 |
| 5.6 | Die Vererbung von Klassen | 55 |
| 5.7 | Verhältnis von Gesamtsystem zu Teilsystemen | 59 |
| 5.8 | Verhältnis von Gesamtmodell, Komponenten und Problemkomponenten | 60 |
| 5.9 | Modellsteuerung durch Expertenklasse | 60 |
| 5.10 | Modellierung der Systemumwelt | 61 |
| 5.11 | Modellierung einer Komponente eines L/R-Modells | 61 |
| 5.12 | Das Koordinatensystem für die Modellierung der Zeit | 63 |

| | | |
|------|---|-----|
| 5.13 | Vergleich von System und Modell zur Begriffsbildung der Kalibrierung . . . | 65 |
| 5.14 | Ein Data Mining-Zyklus | 67 |
| 5.15 | Aufbau eines Kalibrierungsmodells | 69 |
| 6.1 | Interventionsmöglichkeiten bei fehlender Validität | 72 |
| 6.2 | Validitätsarten im Koordinatensystem | 74 |
| 7.1 | Systemkomponenten im Zusammenhang des EWS | 82 |
| 7.2 | Eine Druck-Volumen-Kurve | 86 |
| 7.3 | Schema eines Einkompartiment-Modells | 87 |
| 7.4 | Aufteilung des Lungenvolumens | 88 |
| 7.5 | Vorstellung des seriellen Totraums als Rohr | 90 |
| 7.6 | Ein typisches Expirogramm | 90 |
| 7.7 | Sauerstoff-Bindungskurven mit verschiedenen pH-Werten | 93 |
| 7.8 | Zusammenfassung der Prozesse beim Gasaustausch in den Erythrozyten . . | 95 |
| 7.9 | Prinzip des Zusammenspiels von Ventilation, Perfusion und Diffusion | 96 |
| 7.10 | Das Dreikompartimente-Modell | 99 |
| 8.1 | Eine im physiologischen Modell verwendete Druckkurve | 106 |
| 8.2 | Das Einkompartiment-Modell als Level-Raten-Modell | 107 |
| 8.3 | Das Volumenmodell ohne die Druckberechnungen | 108 |
| 8.4 | Das Dreikompartimente-Modell in Kondensatoren-Darstellung | 110 |
| 8.5 | Die Modellierung der Trachea mit Hilfe zweier Level | 112 |
| 8.6 | Schematische Darstellung der Augmentierungsfunktion | 117 |
| 8.7 | Die Modellierung der Spontanaktivität als L/R-Modell | 118 |
| 8.8 | Aufteilung des Herz-Zeit-Volumens mit Hilfe von Gefäßwiderständen | 119 |
| 8.9 | Sauerstoffdiffusion im Kompartiment | 120 |
| 8.10 | Diffusion im Volumen-Kompartiment | 124 |
| 8.11 | Das Vorgehen bei der Berechnung der Normventilation | 129 |
| 9.1 | Die Vererbungsstruktur des physiologischen Modells | 134 |
| 9.2 | Das Objektmodell des physiologischen Modells | 136 |
| 9.3 | Das Atemzugvolumen | 137 |
| 9.4 | Drucksignal und Druckkurven | 138 |
| 9.5 | Druckkurven mit Augmentierung | 139 |
| 9.6 | Arterieller Sauerstoffpartialdruck auf den verschiedenen Zeitskalen | 140 |
| 9.7 | Arterieller Kohlendioxidpartialdruck auf den verschiedenen Zeitskalen . . . | 140 |
| 9.8 | Ein Expirogramm aus Modelldaten | 142 |
| 9.9 | Expirogramme beim Wechsel der Atemphase mit einem $V_{d_s} = 100$ ml . . . | 143 |
| 9.10 | Expirogramme beim Wechsel der Atemphase mit einem $V_{d_s} = 300$ ml . . . | 143 |
| A.1 | Diskretisierung im Cauchy-Euler-Verfahren | 151 |
| B.1 | Ein Feedforward-Netz mit drei Schichten | 156 |
| B.2 | Berechnung der Netzausgabe in einem geschichteten Netz | 157 |

B.3 Backpropagation in einem neuronalen Netz 159

B.4 Die Struktur des neuronalen Netzes zur Kalibrierung der Blutgase 161

D.1 Das Einkompartiment-Modell als Stromkreis 168

E.1 Vererbungsstruktur für die Bindungskurven 175

E.2 Kohlendioxid-Bindungskurve 176

E.3 Sauerstoff-Bindungskurve als mehrdimensionale Tabelle 179

Danksagung

An dieser Stelle möchte ich mich bei allen bedanken, die die Entstehung der Arbeit wohlwollend begleitet und unterstützt haben.

Den Projektmitarbeitern an der Klinik für Anästhesiologie und den Kooperationspartnern der Firma Drägerwerk AG bin ich zu Dank verpflichtet. Dank gebührt den Betreuern und Gutachtern.

Ich danke den Korrekturlesern, meinen weiteren Freunden und schließlich meiner Familie.

Index

Symbole

C_{Thorax} , *siehe* Thorax/Compliance
 P_{ASB} , 84, 105, 129, 130
 P_{El} , *siehe* Druck/elastischer
 P_{Insp} , 84, 105, 118, 129, 130, 138, 141
 Q , *siehe* Herz-Zeit-Volumen
AZV, *siehe* Atemzugvolumen
Fl.Lunge, 110, 112
RQ, *siehe* Respiratorischer Quotient
VPV, *siehe* Ventilations-Perfusions-Verhältnis
 V_{d_s} , *siehe* Totraum/serieller
FRC, *siehe* funktionelle Residualkapazität
PEEP, 84, 85, 116, 130, 172
PEEP-Ventil, 84, 125, 126, 133
Par_Aug, 116, 128

A

Aggregation, 55, 59
ALV, 82
Alveole, 85, 90, 95
Anforderungsdefinition, 7, 29, 49, 51, 58, 71, 72, 76, 101
Anwendungsgültigkeit, 70–71
Arterialisierung, 95–96, 122
Assoziation, 54, 60, 61
Atemminuten-Volumen, 88, 104, 125, 127, 130, 137–139, 141
Atemphase, 83
Atemzug
 mandatorischer, 85, 105
 maschineller, 105
 spontaner oder ASB-, 84, 105
Atemzugsfrequenz, 105
 maschinelle, 105
 spontane, 105

Atemzugvolumen, 88, 110, 116, 117, 124, 135, 137
Attribut, 10, 61
Augmentierung, 115, 138
Augmentierungsdruck, 116
Avogadro
 Gesetz von, 89

B

B/E-Netze, 32
Backpropagation-Netz, 68, 154–161
 Ausgangsschicht, 156
 Eingangsschicht, 156
 generalisieren, 154
 in Atmungsphysiologie, 160–161
 Lernphase, 68, 154, 158–159
 Netzeingang, 156
 Neuron, 155
 Zustandsmenge, 155
 Nutzungsphase, 68
 Schicht, 156
 Verbindung
 Gewicht, 155
 versteckte Schicht, 156
Bauchatmung, 115
Beatmung, 1, 83
 assistierte, 83
 kontrollierte, 83
Beatmungsdruck, 84
Beobachtungsebene, 44, 47, 48
Beobachtungsskala, 64, 104
BIPAP-ASB, 83–85, 105, 132, 133, 146
Blackbox, 12, 23
Blutgasanalyse, 93, 121, 164
Blutgase, 99
Blutplasma, 91
Bohr-Effekt, 93

Brustatmung, 115

Bunsen-Löslichkeitskoeffizient, 92

C

Cauchy-Euler-Integration, 39, 151

Codierer, 16, 22, 105

Compliance, 86–88

Computerzeit, 64

D

Darstellungsmodell, 26

Darstellungsmodellzeit, 64

Data Mining, 67–69

mit Modelldaten, 68

Zyklus, 67

Datenhintergrund, 26

Datenlage, 22, 102, 154

Decodierer, 16, 22

Differenzgleichung, 39

Differenzialgleichung, 39, 87

Diffusion, 81, 95–98, 119–124

Diffusionsstörung, 98

Diffusionswiderstand, 98, 120

Drägerwerk AG, 1, 81–83

Dreikompartimente-Modell, 99, 106, 107,
109, 115, 126, 136, 146

Druck

elastischer, 87, 108, 115, 138, 173

intrapleuraler, 85, 115–117, 145

intrapulmonaler, 85, 115–117, 145

Drucksignal, 103

E

Ebenenmodell, 28, 49

Eigenschaft, 10

Eingangssignal, 60

Einkompartiment-Modell, 87, 106–108

Einsatzmodell, 22, 75

Empirische Gültigkeit, 70–71

Endzeitpunkt, 10

Entscheidungshilfe, 25, 101, 102

Entwöhnung, 1, 81, 83

Erythrozyten, 91, 94

Evita, 1, 82

EWS, 1, 81–85, 99, 101, 102, 125, 127,
132, 146, 163, 165

Projekt, 1

Experimentierfeld, 13

Experimentvalidität, 74, 76

Expirationszeit, 84

Expirogramm, 90, 112, 113, 142

F

Falsifikation, 43, 48, 70, 71, 75, 77, 132,
145

Fehler, 76

Ficksches Diffusionsgesetz, 96

Flussrelation, 30

funktionelle Residualkapazität, 88, 113

Abschätzung, 114

Funktionsmodell, 22

des Modells, 22, 65, 102

des Systems, 22, 65, 102

G

Gültigkeit, 43, *siehe* Validität

Gehalt, 121

Gewicht, 32

Greybox, 23

Grundmodell, 27

H

Hämatokritwert, 91, 177

Hämoglobin, 91–95, 122, 177

Haldane-Effekt, 95

Herz-Zeit-Volumen, 97, 98

Abschätzung, 119

HZV, *siehe* Herz-Zeit-Volumen

I

Indikatorenvalidität, 47, 73

Informatik, 25, 26, 28, 49, 69

Inhaltsvalidität, 73

Initialisierung, 22

Inspirationszeit, 116, 125

maschinelle, 84

spontane, 84

Integrations-schrittweite, 39, 63

Integrations-test, 77

K

KADS, 53
 Kalibrierung, 65–69, 73
 Kalibrierungsaufgabe, 65, 66, 75, 103, 109, 125, 130
 Kalibrierungsmodell, 69, 130, 154
 Kalibrierungsparameter, 65, 66, 68, 69, 71, 103, 109, 125, 126, 128, 130, 139, 160
 Kalibrierungszielwert, 66, 69, 103
 Kapazität, 109
 Kapazitätsregeln, 167
 Kapillargefäß, 81, 95–98, 120–123, 136
 Kausalnetz, 34
 Kohlendioxid
 endtidales, 113
 Kohlendioxid-Bindungskurve, 94–95, 121–122
 Kohlendioxidfraktion
 endexpiratorische, 90
 endtidale, 90
 Kohlendioxidpartialdruck, 103
 Kompartiment, 87, 97–99, 104, 106, 108, 120, 133, 135–136, 138, 139, 144, 149, 169
 Normal-, 107
 Shunt-, 107
 Totraum-, 107
 Kondensator, 109
 Konstrukt, 3, 10, 47, 48, 50, 73, 74
 Kontaktzeit, 96
 Konzeptvalidität, 47, 73, 74, 76
 Koordinatensystem
 als Lebenszyklus, 144–145
 bei Expertensystemen, 52
 der Modellbildung, 43–50
 Einführung, 2
 ffl@dq ”@prtctur modellbasierte Akquise, 56
 zur Modellierung der Zeit, 63

L

L/R-Modell, 34–38, 56
 abhängige Parameter, 36

allgemeines, 37
 Diagramme, 38, 56
 Dynamik, 35
 einfach, 35
 modulares, 37
 Schnittstellenabbildung, 38
 Signale, 36
 unabhängige Parameter, 36
 Ladung, 109
 Lebenszyklus, 27–29
 Ebenenmodell, 28
 Spiralmodell, 29
 Lernen
 überwachtes, 154
 unüberwachtes, 154
 Level, 34, 104, 106
 Zustand, 35
 Levelgleichung, 36
 Lunge, 104, 107, 109, 113, 135
 Lungenareal, 98, 107
 Lungengefäßwiderstand, 119
 Lungenvolumen, 88, 110, 114, 116, 118

M

Marken, 32
 Markierung, 32
 zeitabhängige, 35
 Markierungsfolge, 33
 Modell, 20–23
 Kontext, 22
 Modelleignung, 27, 46, 49
 Modellgeneratoren, 56–63
 Modellierung
 diskret, 39
 kontinuierlich, 39
 Modellierungsskala, 64
 Modellingenieur, 51, 56, 58, 73, 76, 104, 144, 145
 Modellkonstanten, 65
 Modellparameter, 65, 71, 163
 Modellvalidierung, 27, 28, 44, 46
 Modellverifikation, 27, 28, 46
 Modellzeit, 64

Modellzweck, 22–23, 51, 55, 70, 71, 74,
101

Morphismen

dynamische Systeme, 17
modulare Systeme, 18
relationale Systeme, 17

N

Netz, 30

Anfangsmarkierung, 32
Einbetten, 31
Ereignisknoten, 30
Folgemarkierung, 33
Prozess, 34
Verfeinern, 31
Zustandsknoten, 30

Netzausgang, 156

Netzmodelle, 30–39

Neuron

Aktivierungsfunktion, 155
Ausgabefunktion, 155

Neuronales Netz, 155

Feedforward-, 156

Normbedingungen, 89, 114

ATPS-, 89

BTPS-, 89, 122

STPD-, 89, 92

Normventilation, 99, 103, 129, 141–142,
165

Berechnung, 129–130

O

Objekt, 9–10, 43, 73

Operationalisierungsvalidität, 74, 76

Operationen, 62

Optimierung, 66

Ordnungsrelation, 10

P

Partialdruck, 91

Partialdruckdifferenz, 120

Perfusion, 81, 90–95, 119

Pflichtenheft, 22

Phasenraum, 12

physiologisches Modell, 1, 101–131

Datenstruktur-Modell, 104

Einsatzmodell, 102

Einschwingen, 128–129

Funktionsmodell, 102

Kalibrierungsaufgabe, 103

Simulation, 124–125

Zeitskala, 104

Zweck, 101–104

Polymorphie, 55, 59

Problembereich, 26, 51, 52, 59, 63, 64, 72,
76, 133

Prognosemodell, 25, 65, 66, 101, 132

Prototyping, 29, 50, 52, 56–63, 72–73,
144, 145

modellbasiertes, 72

Q

Quelle, 31

Quellenereignis, 31

Quellenraten, 36

R

Rückkopplung, 15

Rampenzeit, 85, 105

Rate, 34, 106

Ratengleichung, 35, 36, 39, 56, 166

allgemeine, 37

einfache, 35

Realisierungsraum, 12, 103

Realzeit, 64

Resistance, 86–88

Respiratorischer Quotient, 100, 123

Abschätzung, 123

S

S/T-Netz, 157

S/T-Netze, 32–34, 39

Sauerstoff-Bindungskurve, 93, 121

Sauerstoffdiffusionskapazität, 96

Sauerstofffraktion

inspiratorische, 85, 89, 111, 114, 160

Sauerstoffpartialdruck, 103

arterieller, 97

Sauerstoffsättigung, 92

Schalten, 33

asynchron, 33, 34, 39
 synchron, 36, 39
 Schaltfolge, 33
 Schnittstellenabbildung, 15
 Senke, 31
 Senkenereignis, 31
 Sensitivitätsanalyse, 68
 Shunt, 97–98
 Shuntfraktion, 99, 109, 160
 Sigmoid-Funktion, 155
 Signal, 10, 12
 Ausgangs-, 12
 Eingangs-, 12
 synchron, 12
 Signalraum, 11
 Simulation, 23–25
 ereignisorientierte, 24, 39
 zeitorientierte, 23, 39, 60, 124
 Simulationsdauer, 23
 Simulationsmodell, 23, 25, 26, 30, 45, 52,
 56, 57, 65, 72
 Simulationsparameter, 23, 65, 163
 Simulationssystem, 17
 Simulationswerkzeug, 26
 Sitzungsparameter, 65
 Spannung, 109
 Speicherfrequenz, 64
 Spontanaktivität, 87, 115–118, 144, 146
 Sportinformatik, 57
 Standardentwurf, 26, 64
 Startzeitpunkt, 10
 Startzustand, 13
 Stelle, 32
 Kapazität, 32
 Stromfluss, 109
 Stromkreise, 109–110, 166–168
 Strukturgültigkeit, 70–71, 133
 Strukturnetze, 30–32
 Strukturtreue, 18, 23
 Strukturvalidität, 74, 76
 Subjekt, 43
 System, 9–19
 Überföhrungsfunktion, 14
 Ausgabefunktion, 14

Dimension, 15
 dynamisches, 14–16
 Funktional, 14
 Komponente, 15
 modulares, 15
 Morphismen, 16–19
 relationales, 11–13
 Zustand, 14
 Systemanalyse, 54–55
 Systemanalytiker, 28
 Systemdefinition, 21
 Systemgrenze, 12
 Systemkomponente
 Interaktion, 23
 Systemmodell, 20, 26, 29
 Systemobjekt, 20, 66
 Systemumwelt, 12
 Systemzweck, 12
 Szenario, 25, 68, 132, 133

T

t-Konkatenationsprodukt, 11
 Test, 76
 Blackbox-, 76
 Whitebox-, 76
 Theorie, 43–44
 Thorax, 85, 87, 109, 115–118, 145
 Compliance, 110, 135, 138
 Totraum, 98
 alveolarer, 98
 physiologischer, 98
 serieller, 89–90, 98, 112, 113, 125,
 128, 142
 Totraumfraktion, 99, 109, 160
 Trachea, 81, 107, 110, 112–115, 125, 133–
 136, 142
 Compliance, 126
 Transition, 32
 M-aktiviert, 33
 schalten, 33

U

UML, 54, 132

V

V-Modell, *siehe* Ebenenmodell
 Validierung, 27, 29, 73
 Validität, 70–77
 Ventilation, 81, 85–90, 106–114
 Ventilations-Perfusions-Verhältnis, 98,
 99, 107, 120
 Verbindung, 155
 Vererbung, 55, 59
 Vergangenheit, 10
 Verhaltensaussage, 23
 Verhaltensgültigkeit, 70–71, 77, 132, 144
 Verhaltensgrößen, 14
 Verhaltensvalidität, 74
 kontextuelle, 75
 qualitative, 75
 quantitative, 75
 Verifikation, 27, 73
 Verzögerungszeit, 35

Zustandsabbildung, 17
 Zustandsgrößen, 14, 104

W

Weaning, 1, 81
 Whitebox-Modellierung, 23, 45, 71, 77,
 102, 133
 Widerstand, 109
 Widerstandsregeln, 167
 Wissensakquisition, 51–56, 58, 72, 76, 144
 modellorientierte, 52–54
 prototypische, 52
 Wissenserwerb
 automatischer, 53, 69
 direkter, 53
 indirekter, 53
 Wissensingenieur, 51, 53

Z

Zeitintervall, 10
 Zeitkonstante, 88
 Zeitmenge, 10
 Zeitpunkt, 10
 Zeitskala, 10, 24, 63–64, 104
 diskrete, 10
 kontinuierliche, 10
 Zukunft, 10