Objektorientierte Modellierung thermischen Gebäudeverhaltens

Vom Fachbereich Elektrotechnik und Informationstechnik
der Universität Kaiserslautern
zur Verleihung des akademischen Grades
„Doktor der Ingenieurwissenschaften (Dr.-Ing.)“
genehmigte Dissertation

Von

Dr. rer. nat. Dipl. Phys. Rolf Mathias Merz
geb. in Kaiserslautern

D 386

Kaiserslautern, September 2002

Eingereicht am: **15.4.2002**

Tag der mündlichen Prüfung: **05.9.2002**

Dekan des Fachbereiches: **Prof. Dr.-Ing. habil. R. Urbansky**

Promotionskommission

Vorsitzender: **Prof. Dr.-Ing. D. Nelles**

Berichterstattende: **Prof. Dr.-Ing. habil. L. Litz**

Prof. Dr. H. Heinrich

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Lehrstuhl für Automatisierungstechnik im Fachbereich Elektrotechnik und Informationstechnik der Universität Kaiserslautern. Die Arbeit gliedert sich in das Teilprojekt D2 („Modellbasierte Entwicklung wiederverwendbarer Regelungsalgorithmen, Arbeitsschwerpunkt: Mathematische Prozessmodelle“) des Sonderforschungsbereiches 501 („Entwicklung großer Systeme mit generischen Methoden“) des Fachbereiches Informatik der Universität Kaiserslautern ein.

Herrn Prof. Dr.-Ing. habil. L. Litz danke ich herzlich für die wissenschaftliche Betreuung der Arbeit und für die vielen thematischen Diskussionen, die wesentlich zum Erfolg dieser Arbeit beitrugen. Besonders bedanken möchte ich mich auch für die Freiheit, mit der ich das Forschungsprojekt bearbeiten konnte. Sie ist Grundlage für jedes kreative Arbeiten.

Meinen ehemaligen Kolleginnen und Kollegen danke ich für die gute Zusammenarbeit und die freundschaftliche Atmosphäre am Lehrstuhl.

Bedanken möchte ich mich an dieser Stelle auch bei allen Studenten, die als Hilfsassistenten\-Innen, Studienarbeiter, Diplomanden oder Master-Kandidaten\-Innen an den Ergebnissen dieses Forschungsprojektes Teil hatten. Besonders erwähnen möchte ich hier Frau Shanti Agustina, Herrn Rafel Cladera Bohigas, Herrn Felix Felgner und Herrn Erwin Sitompul, die wesentlich an der Entwicklung der umfangreichen Modelica-Bibliothek beteiligt waren.

Für die Unterstützung bei der Validierung der Modellbibliotheken möchte ich den beteiligten Mitarbeitern des Zentralbereiches Forschung und Entwicklung der Robert **Bosch** *GmbH und besonders Herrn Dr.-Ing. Kai Oertel danken.*

Erklärung

Die vorliegende Arbeit wurde von mir selbst angefertigt, alle benutzten Hilfsmittel sind in der Arbeit angegeben. Die vorliegende Dissertation oder Teile von ihr wurden nicht als Prüfungsarbeit für eine staatliche oder andere wissenschaftliche Prüfung eingereicht. Die gleiche oder eine andere Abhandlung wurde bei keinem anderen Fachbereich oder anderen Universität als Dissertation eingereicht.

.....Kaiserslautern den.....

(Dr. Rolf Merz)

1 Einleitung 8

2 Die Vorgehensweise der Modellerstellung 10

2.1 Begriffsbildung 10
2.2 Generische Konzepte 11
2.3 Anwendung des V-Modells 13
2.4 Top-Down- und Bottom-Up-Strategie 17
2.4.1 Top-Down-Strategie 17
2.4.2 Bottom-Up-Strategie 18
2.5 Nutzung von Modellbibliotheken 19
2.5.1 Selektion und Adaption 19
2.5.2 Komposition 20
2.6 Wiederverwendungsorientierter Regelungsentwurf 21

3 Konzepte zur Strukturierung 24

3.1 Objektorientierte mathematische Modellierung 24
3.1.1 Modularität 24
3.1.2 Klassenprinzip 24
3.1.3 Aggregationshierarchie 26
3.1.4 Vererbungshierarchie 27
3.1.5 Kapselung und Schnittstellen 28
3.2 Topologische Systembeschreibungen 29
3.2.1 Ein Illustrationsbeispiel 29
3.2.2 Signalflussverknüpfung 30
3.2.3 Energieflussverknüpfung 32
3.2.4 Phänomenologische Verknüpfung 35
3.2.5 Nicht berechnungskausale Modellierung 37

4 Charakterisierung und Auswahl einer Entwicklungsumgebung . . 39

4.1 Matlab/Simulink 39
4.1.1 Die Entwicklungsumgebung 39
4.1.2 Das Carnot-Gebäudemodell 40
4.1.2.1 Charakteristika 40
4.1.2.2 Komponentenmodelle 41

4.1.2.3 Hydraulische Anlage 43
4.1.2.4 Ein Beispiel zur Verschaltung von Rohren 43
4.1.3 Das HeatCon-Gebäudemodell 44
4.2 TRNSYS 47
4.2.1 Die Entwicklungsumgebung 47
4.2.2 Das Gebäudemodell 49
4.2.2.1 Bilanzierung der Wärmeströme 49
4.2.2.2 Wandmodell 51
4.2.2.3 Langwelliger Strahlungsaustausch 52
4.2.3 Das Umgebungsmodell 54
4.3 Dymola/Modelica 58
4.3.1 Die Entwicklungsumgebung 58
4.3.1.1 Charakteristika 58
4.3.1.2 Erstellung des physikalischen Ersatzmodells 58
4.3.1.3 Erzeugung des mathematischen Modells 60
4.3.1.4 Automatische Codegenerierung 64
4.3.2 Die Sprache Modelica 64
4.3.3 Die verfügbaren Modellbibliotheken 65
4.3.3.1 Organisation der Modelica Bibliotheken 65
4.3.3.2 Modelica Standard Library 66
4.3.3.3 Modelica Additions Library 66
4.3.3.4 Licensed Modelica Library (HyLib) 68
4.3.3.5 Beta-Versionen (ThermoHyd) 68
4.4 Andere Umgebungen 70

5 Erstellung der Modellbibliothek 71

5.1 Die Gebäude-Bibliothek 72
5.1.1 Die Begriffsbildung 72
5.1.2 Ein Beispiel 73
5.1.3 Das Wandmodell 74
5.1.4 Die thermische Zone 75
5.1.4.1 Abgrenzung und Definition 75
5.1.4.2 Bilanzierung der Wärmeströme 76
5.1.4.3 Wärmeaustausch der Innenwände 76
5.1.4.4 Implementierung 78

5.1.5 Das Fenstermodell 79
5.1.5.1 Wärmeaustausch 79
5.1.5.2 Luftaustausch 80
5.1.5.3 Solare Einstrahlung 81
5.1.6 Die ideale Heizung 83
5.2 Die Thermohydraulik-Bibliothek 83
5.2.1 Das Rohrmodell 84
5.2.1.1 Hydraulischer Anteil 85
5.2.1.2 Thermischer Anteil 86
5.2.2 Das Kesselmodell 87
5.2.3 Das Pumpenmodell 88
5.2.4 Die Ventilmodelle 88
5.3 Die Umgebungs-Bibliothek 89
5.3.1 Die atmosphärische Gegenstrahlung 89
5.3.1.1 Begriffsbildung 89
5.3.1.2 Himmelstemperatur 90
5.3.1.3 Sättigungstemperatur 90
5.3.1.4 Fiktive Himmelstemperatur 91
5.3.2 Die solare Einstrahlung 91
5.3.2.1 Berechnung der wahren Ortszeit (WOZ) 92
5.3.2.2 Berechnung der Deklination 92
5.3.2.3 Berechnung des Zenit- und Azimutwinkels 93
5.3.2.4 Projektion auf beliebig geneigte Flächen 93
5.3.2.5 Interpolation der Strahlungsdaten 94
5.3.2.6 Implementierung 95
5.3.2.7 Ein Beispiel eines vollständigen Umgebungsmodells 96
5.4 Die Algorithmen-Bibliothek 98
5.4.1 Die Reglerbausteine 98
5.4.1.1 Kontinuierlicher PID-Regler 98
5.4.1.2 Diskreter PI -Regler 99
5.4.2 Die Erweiterungen der klassischen Reglerbausteine 99
5.4.2.1 Glättungsglied 99
5.4.2.2 Nichtlineare Erweiterungen 100
5.4.2.3 Anti-wind-up-Maßnahme 101
5.4.2.4 Betriebsartensteuerung 103

5.4.3 Ein Beispiel eines vollständigen Reglers 103
5.5 Ein Beispiel eines vollständigen Gebäudemodells 104

6 Die Validierung der Modellbibliothek 106

6.1 Eine Simulation topologisch verschieden strukturierter Modelle 106
6.1.1 Die Implementierung der Signalflussverknüpfung 107
6.1.2 Die Implementierung der Energieflussverknüpfung 108
6.1.3 Die Implementierung der phänomenologischen Verknüpfung 109
6.2 Das Konzept der konzentrierten Speicher 112
6.3 Die Simulation eines Einfamilienhaus des Bestandes (EFHB) 114
6.3.1 Die Implementierung in Modelica 114
6.3.2 Die Validierung des EFHB-Modells 117
6.3.3 Der Testfall A: Stoßlüften 121
6.3.4 Der Testfall B: Zeitabhängige Raumtemperatursollwerte 126
6.4 Die Ankopplung Dymola Modelica 132

7 Zusammenfassung 134

8 Literatur 135

9 Anhang 145

9.1 Die physikalischen Grundlagen des Beuken-Modells 145
9.2 Die mathematischen Grundlagen der Transferfunktionsmethode 147
9.2.1 Die Laplace-Transformation 147
9.2.2 Die z-Transformation 149

1 Einleitung

Während der letzten Jahrzehnte war es üblich, heizungs- und klimatechnische Anlagen nach vereinfachten, meist statischen Berechnungsverfahren auszulegen und konsequenterweise überzudimensionieren [VDI-6020]. Die Verteuerung der Primärenergiekosten, der erhöhte Kostendruck auf private und öffentliche Auftraggeber, aber auch erhöhte Komfortansprüche erzwangen einen Wandel der Vorgehensweise [Sch-96]. So nimmt die dynamische Simulation des Gebäude- und Anlagenverhaltens bei der Planung und Auslegung heizungs- und klimatechnischer Anlagen einen immer höheren Stellenwert ein ([BaLe-96], [LEK-97]). Unterstützt wird dieser Effekt durch die zunehmende Rechenleistung gängiger Computer. Berechnungsmethoden, die bislang als zu aufwändig und zeitintensiv galten, wurden praktisch anwendbar und daher auch verfeinert.

Ziel der Gebäude- und Anlagensimulation ist es [Fei-94], das thermische und energetische Verhalten eines realen oder fiktiven Gebäudes und seiner technischen Anlage sowie deren Interaktion nachzubilden. Zu berücksichtigen sind hierbei die äußeren Einflüsse durch das Außenklima, das Benutzerverhalten sowie innere Lasten. Gerade eine ganzheitliche Gebäudeplanung erfordert die adäquate Beschreibung realer Prozesse aus einem breiten Spektrum mathematischer, physikalischer und ingenieurwissenschaftlicher Disziplinen.

Es existiert eine Vielzahl von Simulationsprogrammen, die sich hinsichtlich ihrer Methode, der berücksichtigten Effekte, aber auch bezüglich der Zielsetzung unterscheiden. Eine besondere Problematik stellt die Überprüfung der Ergebnisse dar. Gerade Simulationssysteme, die eine menügeführte intuitive Modellerstellung unterstützen, können vom Anwender hinsichtlich der genutzten Numerik, der berücksichtigten Effekte, der angewendeten Näherungen nicht vollständig überschaut werden. Hier wurden verschiedene Bestrebungen durchgeführt, um eine Vergleichbarkeit der Simulationssysteme herzustellen und dem Anwender eine Möglichkeit der Überprüfung an die Hand zu geben [JuNe-95, VDI-6020]. Zudem mangelt es solchen Systemen, auch wenn die Eingabeoberflächen bausteinorientiert aufgebaut erscheinen, durch die starre Implementierung der Bausteine häufig an Flexibilität. Eine gravierende Einschränkung bei regelungstechnischen Problemstellungen stellt meist die grobe und nicht skalierbare Zeitrasterung dar.

Im Rahmen dieser Arbeit wird ein anderer Weg beschritten. Die Abbildung der physikalischen Realität in eine mathematische Beschreibungsform sollte im Gegensatz zu anwendungsspezifischen Simulationsprogrammen weitgehend selbst und frei gestaltet werden können. Dies ist Grundlage zur Realisierung und Nutzung eigener methodischer Konzepte der Modellerstellung.

In der Simulationstechnik werden meist Engineering-Tools genutzt, die den Anwender von der Implementierung der numerischen Details befreien, eine Visualisierung und Analyse der Simulationsergebnisse ermöglichen und bei der Erstellung von Modellbibliotheken unterstützen.

Neuere Entwicklungen greifen zunehmend auf objektorientierte, nicht berechnungskausale mathematische Beschreibungsformen zurück. Ein neuer Standard *Modelica*, der den interdisziplinären Austausch von Modellspezifikationen ermöglichen soll, wird von der *Modelica*-Association vorgeschlagen [Mod-02, Til-01]. Gerade beim Modellentwurf großer, komplexer Systeme zeigen sich entscheidende Vorteile gegenüber klassischen blockschaltbildorientierten Simulationssystemen. An erster Stelle ist hierbei die Möglichkeit einer methodischen Einbeziehung der Wiederverwendung zur Qualitäts- und Effizienzsteigerung zu nennen.

Ziel dieser Arbeit ist es, die Anwendbarkeit generischer, wiederverwendungsorientierter Konzepte bei der Modellerstellung komplexer, mathematischer Modelle zur Simulation thermischen Gebäudeverhaltens aufzuzeigen. In der Informatik stellt die Entwicklung generischer Methoden des Systementwurfes einen aktuellen Forschungsgegenstand dar [Ave et al.-98]. Konzepte, Erfahrungen und Methoden sind prinzipiell übertragbar und auch bei der Modellerstellung nutzbar. Auch ein implementiertes mathematisches Simulationsmodell stellt schließlich Software dar.

Schon die Abbildung einer einfachen Heizungsanlage benötigt thermodynamische, fluiddynamische, mechanische, elektrische und regelungstechnische Komponenten. Basis für eine generische Modellerstellung ist zunächst die Verfügbarkeit geeigneter Komponentenbibliotheken. Durch Selektion, Adaption und Komposition von Komponentenmodellen können dann komplexe Simulationsmodelle für konkrete technische Prozesse erstellt werden.

Die Schwierigkeit liegt bei der wiederverwendungsgerechten Spezifikation der Bibliothekseinträge. Die Komplexität großer Systeme ist durch Strukturierung und Hierarchisierung aufzulösen. Gemeinsamkeiten und Variabilitäten sind zu identifizieren, um mit möglichst wenigen Komponentenmodellen eine möglichst große Bandbreite realer technischer Prozesse abzudecken. Generische Parameter sind zu definieren, welche die Variabilitäten der Komponentenmodelle spezifizieren.

Entsprechende Komponenten- bzw. Modellbibliotheken wurden entwickelt und in exemplarischen Konfigurationen simulativ sowie experimentell validiert.

2 Die Vorgehensweise der Modellerstellung

2.1 Begriffsbildung

Zunächst wird in diesem Abschnitt die etablierte Begriffsbildung bei der theoretischen Analyse eines technischen Prozesses vorgestellt. Sie folgt den Ausführungen nach Lit-01b. Ein technischer Prozess stellt die Gesamtheit aller zugehörigen Vorgänge dar, bei denen Materie und Energie transportiert, umgewandelt und gespeichert sowie Information transportiert, umgewandelt, gespeichert, erzeugt und vernichtet werden. Ein System stellt einen vom jeweiligen Anwender abgegrenzten Prozessteil dar, der über Masse-, Energie- und Informationsflüsse mit seiner Umgebung in Verbindung steht. Unter einem Modell versteht man eine geeignete Beschreibungsform für die Zusammenhänge dieser Flüsse. Ziel hierbei ist es, Aussagen über das Prozessverhalten bei beliebigen Randbedingungen machen zu können. Das Verfahren der theoretischen Analyse zur Erstellung eines Simulationsmodells ist etabliert und wird hier kurz zusammengefasst (Abb. 1).

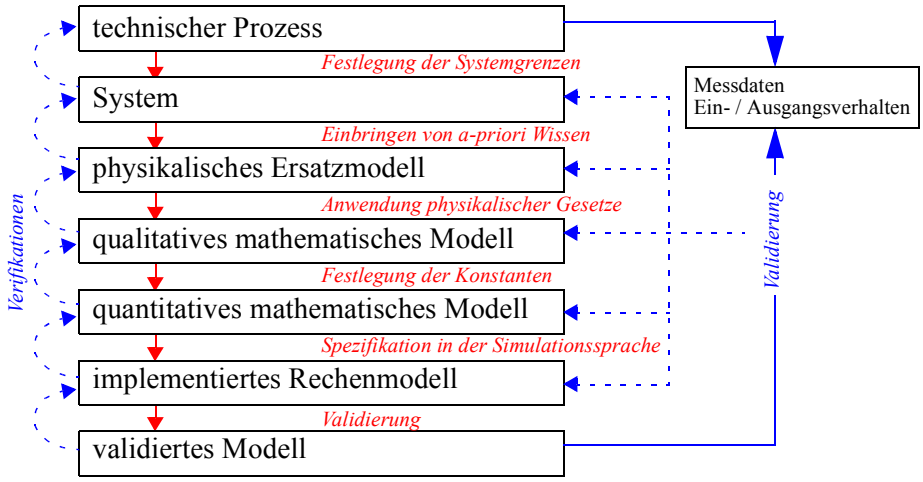


Abbildung 1: Vorgehensweise der theoretischen Analyse zur Modellerstellung nach Litz [Lit-01b]

Zunächst sind die Systemgrenzen aufgrund der konkreten Aufgabenstellung, aber auch aufgrund von Basiswissen über den Prozess zu definieren. Es folgt ein physikalisches Ersatzmodell, beispielsweise in Form eines elektrischen, mechanischen, hydraulischen oder pneumatischen Schaltbildes. Mit Hilfe physikalischer Gesetze gelangt man zu einem mathematischen Modell.

Die angewandten Gesetze basieren meist auf der Bilanzierung physikalischer Erhaltungsgrößen (*Masse, Energie, Impuls, Ladung, Drehimpuls*) oder folgen aus sogenannten phänomenologischen Gesetzen. Dies sind experimentell nachgewiesene Zusammenhänge, also Erfahrungstatsachen, die irreversible Prozesse beschreiben. Meist stellen sie einen funktionalen Zusammenhang her zwischen der zeitlichen Änderung bzw. dem Strom einer mengenartigen Größe und einer Potentialdifferenz, also der Differenz zweier nichtmengenartiger Größen.

Das mathematische Modell hat zunächst einen qualitativen und nach Festlegung aller Konstanten einen quantitativen Charakter. Hierauf basierend wird das implementierte, lauffähige Rechenmodell erstellt.

Mit Hilfe von Simulationen ist im Schritt der Validierung zunächst zu überprüfen, ob das Modell in der Lage ist, die für die Fragestellung relevanten Aspekte des technischen Prozesses geeignet zu beschreiben. Hierbei können durchaus Iterationen verbunden mit Änderungen in den vorangegangenen Zwischenschritten erforderlich sein. Als Referenz können Messdaten des technischen Prozesses oder bekanntes Prozessverhalten dienen. Letzteres kann auf unabhängigen, bereits validierten Simulationen beruhen.

Eine der anspruchsvollsten und daher auch in Zukunft nur manuell lösbaren Aufgaben ist die Erstellung des physikalischen Ersatzmodells. An diesem Punkt fließt Erfahrung, Wissen und Intuition des Modellentwicklers ein.

Zunächst wird er versuchen, das zu betrachtende System hierarchisch in kleine, überschaubare Komponenten zu untergliedern. Neben der besseren Überschaubarkeit ergibt sich auf Komponentenebene eine erhöhte Wahrscheinlichkeit der Wiederverwendbarkeit bereits entworfener Komponenten. Dies gilt sowohl für Komponentenmodelle aus dem laufenden Projekt als auch für Komponentenmodelle aus älteren, abgeschlossenen Projekten.

Die gezielte Einbeziehung der Wiederverwendung als Entwurfsmethode ist von Vorteil für die Entwurfseffizienz und zur Reduktion möglicher Fehler, da bereits getestete verifizierte Komponenten anstatt eventuell fehlerbehafteter Neuimplementierungen eingesetzt werden. Die Methode des komponentenbasierten Modellentwurfes kann Erfahrungen aus der Vorgehensweise bei der Entwicklung großer Softwaresysteme nutzen. Dies wird im folgenden beschrieben.

2.2 Generische Konzepte

Es geht hierbei darum Konzepte der Software-Technik zur Erstellung großer Softwaresysteme [Ave et al.-98] in die mathematische Modellbildung zu übertragen und zu nutzen. Auch ein mathematisches Modell stellt, zur Simulation implementiert, lauffähigen Code, also Software, dar.

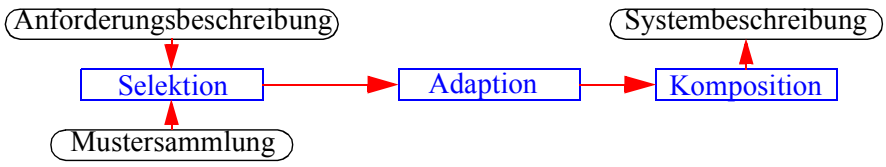


Abbildung 2: Prozessmodell generischer Softwareentwicklung nach Basili [Bas-90]

Bei grober Abstraktion kann man ein generisches Vorgehensmodell der Softwareentwicklung ähnlich Abb. 2 postulieren [Bas-90]. Aus einer Mustersammlung oder Bibliothek werden Komponenten, Erfahrungen, Vorgehensmodelle aufgrund einer Anforderungsbeschreibung selektiert, an die Problemstellung adaptiert und im Schritt der Komposition zu einer vollständigen Systembeschreibung überführt. Wieder verwendet werden hierbei die der Mustersammlung oder Bibliothek entnommenen Muster. Die Ausprägung der Muster sowie die Vorgehensweise der Selektion, Adaption und Komposition ist domänenspezifisch.

Bei der Beschreibung großer, komplexer physikalischer Systeme gelangt man zu hoch komplexen mathematischen Modellen. Ihr Anteil ist in stetigem Wachstum begriffen. Neben einer immer größeren Anzahl von Anforderungen an Systemneuentwicklungen sind auch bestehende Systeme im Hinblick auf eine verbesserte Prozessführung detaillierter zu analysieren. Auch erlaubt die steigende Leistungsfähigkeit verfügbarer Rechnersysteme die Betrachtung umfangreicher Modelle, die bislang nicht mit vertretbarem Aufwand zu simulieren waren [Pan-99].

Gerade bei der Entwicklung und Behandlung solcher Modelle kann die Anwendung generischer Konzepte von großer Bedeutung für die Entwurfseffizienz und Qualität sein. Denn neben der Einsparung des Entwicklungsaufwandes von Neuentwürfen reduziert der Einsatz bereits vorhandener, getesteter Modelle die Häufigkeit von Fehlern.

Unter generischer Erzeugung komplexer mathematischer Modelle sollen *(wie im allgemeinen Falle eines Software-Entwicklungsprozesses [Ave et al.-98])* alle Methoden, Techniken und Werkzeuge verstanden werden, die bei der Neuentwicklung eines Systems die Wiederverwendung unterstützen. Wieder verwendet werden sollen Erfahrungen aus vorangegangenen Entwicklungsprozessen, allgemeines Domänenwissen, Entwicklungsschritte und vor allem bereits existierende Produkte.

Im Falle der mathematischen Modellbildung handelt es sich bei den Produkten um mathematische Modelle. Letztere können Komponenten oder vollständige Systeme beschreiben. Sie können in Form mathematischer Spezifikationen oder in Form implementierter Codes vorliegen.

Die Nutzung generischer Konzepte bei der Modellerstellung bedeutet nicht, dass der komplette Erstellungsprozess automatisiert werden kann. Vielmehr ist es das Ziel, die Arbeit des Modellentwicklers bei der Erstellung eines physikalischen Ersatzmodells durch eine wiederverwendungsorientierte Vorgehensweise zu unterstützen. Basierend auf dem physikalischen Ersatzmodell kann dann eine automatische Codegenerierung erfolgen.

Eine Besonderheit der Anwendungsdomäne mathematische Modellbildung ist, dass ein hoher Anteil fachspezifischen Domänenwissens zur Modellbildung, Regelungstechnik und Simulation notwendig ist. Es liegt aber auch eine Vielzahl häufig wiederkehrender, ähnlicher Problemstellungen vor. Daher erscheint die Nutzung wiederverwendungsorientierter Konzepte lohnend.

Zur Realisierung eines möglichst hohen Wiederverwendungsanteils bei der Modellerstellung sind verschiedene Konzepte von Vorteil, die in den folgenden Abschnitten beschrieben werden.

Die Systemdekomposition anhand des V-Modells (Abschnitt 2.3) führt zu einer komponentenbasierten Beschreibung und elementaren, wiederverwendbaren Bausteinen [MeLi-00a]. Die Bausteine sind in einer geeigneten Bibliotheksstruktur (Abschnitt 2.5) zu hinterlegen. Darüberhinaus sind auch komplexere Bausteine in der Bibliothek zu speichern, die bereits das Resultat der Komposition elementarer Bausteine darstellen.

Die Ausprägung der Schritte Selektion, Adaption und Komposition in der Anwendungsdomäne (Abschnitt 2.5) sind Voraussetzung für das Wiederauffinden und die Wiederverwendung der Bausteine. Die Anwendung einer *Top-Down*-orientierten Vorgehensweise (Abschnitt 2.4) bewirkt, dass bei der Selektion möglichst komplexe Teilprozessmodelle der Bibliothek ausgewählt werden. Erst im Falle eines Nichtauffindens einer geeigneten Komponente ist die Granularität der Beschreibung zu erhöhen. Im schlimmsten Fall sind aus-

schließlich elementare Bausteine entsprechend dem *Bottom-Up*-Verfahren (Abschnitt 2.4) zu komponieren.

Gerade bei großen, komplexen Systemen, bei denen die Bedeutung wiederverwendungsorientierter Entwürfe sehr hoch ist, unterstützen methodische Überlegungen zur vertikalen und topologischen Systemstrukturierung (Kapitel 3) einen umfangreicheren Einsatz der Wiederverwendung, der über die alleinige Wiederverwendung elementarer Bausteine hinausgeht.

Die Auswahl einer geeigneten Simulationsumgebung (Kapitel 4) ist Grundlage zur Nutzung der theoretischen Konzepte.

2.3 Anwendung des V-Modells

Im Bereich der Softwareentwicklung wird häufig eine Vorgehensweise nach dem sogenannten V-Modell angewandt. Ausgehend von einer groben, meist unvollständigen Problembeschreibung werden beim V-Modell (Abb. 3) zunächst die an das zu erstellende System gerichteten Anforderungen aus Benutzer- und Entwicklersicht in einer Anforderungsbeschreibung dokumentiert ([BrDr-95], [RoVe-95]).

Der Systementwurf basiert im Falle komplexer Systeme auf einer hierarchischen Dekomposition in Subsysteme. Nach der Spezifikation des erforderlichen Verhaltens der Komponenten in den Komponentenanforderungen folgt der eigentliche Komponentenentwurf. Es schließt sich die Implementierung sowie die Erzeugung der lauffähigen Komponenten an. Anschließend erfolgt die Komposition zum Gesamtsystem.

Alle Entwicklungsschritte werden zum Nachweis der Korrektheit gegen die Vorgänger verifiziert. Die lauffähigen Komponenten bzw. Systeme werden gegen die jeweils zugrunde liegenden, korrespondierenden Anforderungsdokumente auf Komponentenebene bzw. Systemebene validiert. Während bei der Verifikation überprüft wird, ob ein Produkt richtig entworfen wurde, wird bei der Validierung überprüft, ob das richtige Produkt entworfen wurde.

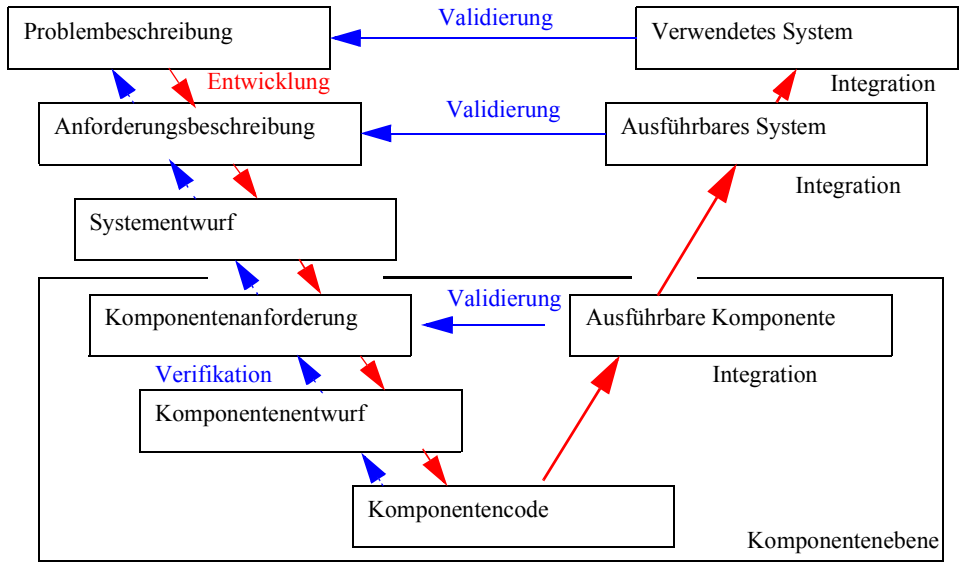


Abbildung 3: Vorgehensweise bei der Softwareentwicklung nach dem V-Modell [RoVe-95]

Charakteristisch und typisch für den Entwicklungsprozess der Anforderungsbeschreibungen ist eine sukzessive Verfeinerung und Detaillierung zunächst evtl. noch unvollständiger Dokumente geringeren Informationsgehaltes durch projektbegleitende Einbeziehung des Wissens- und Erfahrungsschatzes sowohl der Softwareentwickler als auch der beratenden Experten der aktuellen Anwendungsdomäne.

Abb. 4 zeigt, wie die Vorgehensweise des V-Modells der Softwareentwicklung auch auf die theoretische Analyse eines technischen Prozesses angewandt werden kann [MeLi-99, MeLi-00a].

Im Gegensatz zur Darstellung in Abb. 1 findet sich in Abb. 4 als erster Schritt der Systemanalyse die Dekomposition in Subsysteme bzw. als letzter Schritt die Systemintegration aller Komponentenmodelle. Auf der Komponentenebene wird dann die bereits diskutierte Vorgehensweise angewandt.

Der Vorteil der Dekomposition ergibt sich durch die alleinige Bearbeitung überschaubarer, kleiner Teilsysteme geringerer Komplexität. Auch die logische Validierung der implementierten Rechenmodelle gegen die qualitativen mathematischen Komponentenmodelle bzw. die physikalische Validierung gegen die entsprechenden Ersatzmodelle zur Überprüfung der inneren Schlüssigkeit kann zunächst auf Komponentenebene erfolgen. Gleiches gilt für die messtechnischen Validierungen.

Die Zerlegung in Komponenten ist Voraussetzung für weiterführende, generische Konzepte der Modellerstellung.

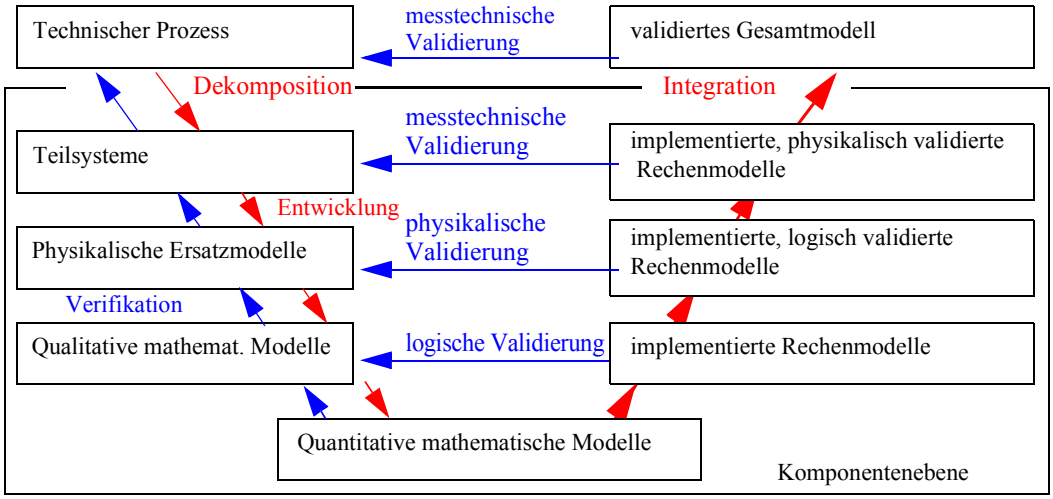


Abbildung 4: Theoretische mathematische Analyse eines technischen Prozesses ([Lit-01a], [MeLi-00a])

Man wird bestrebt sein, nicht nur vereinzelt Bausteine eines großen Entwicklungsprojektes wieder zu verwenden, sondern möglichst vollständige Strukturen. Der Idealfall ist der erneute Einsatz eines bereits validierten Gesamtmodells. Je vollständiger die entwickelte Struktur ist, desto geringer ist die Wahrscheinlichkeit eines erneuten unveränderten Einsatzes.

Einen ersten Zugang zu dieser Problematik bietet der in den Ingenieurwissenschaften etablierte Begriff der Produktfamilie, der im Rahmen des Software-Engineering zunehmend auf die Erstellung von Software-Systemen Anwendung findet [Wei-99]. Unter einer Produktfami-

lie versteht man eine Menge von Produkten mit identischem Konstruktionsprinzip. In Einzelheiten dürfen sich die Produkte der Produktfamilie unterscheiden. Wichtig für die weitere Vorgehensweise ist daher zunächst das Erkennen der Variabilitäten und Gemeinsamkeiten der entwickelten Produkte.

Hat man das gemeinsame Konstruktionsprinzip einer Produktfamilie erkannt [Gey et al.-00], kann man einen für alle Produkte der Produktfamilie anwendbaren Konstruktionsplan (Abb. 5) erstellen. Dieser ist zunächst unvollständig. Er besitzt Lücken an den Stellen, an denen die Variabilitäten der Produkte der Produktfamilie berücksichtigt werden müssen. Ein generischer Entwurfsprozess besteht darin, aus einer Komponentenbibliothek Bausteine zu entnehmen, welche die spezielle Ausprägung der Variabilität der Produktfamilie für den konkreten Anwendungsfall berücksichtigen, und mit ihnen die Lücken auszufüllen. Voraussetzung zur Anwendbarkeit dieser Methode ist die klare Definition der Schnittstellen an den Lücken des zu ergänzenden Konstruktionsplans der Produktfamilie.

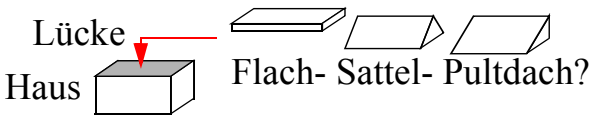


Abbildung 5: Konstruktionsplan der Produktfamilie Haus

In der Modellbildung entsprechen die zu erstellenden Produkte mathematischen Gesamtmodellen zur Beschreibung verschiedener, realer, technischer Prozesse. Die technischen Prozesse und damit auch der Konstruktionsplan der mathematischen Modelle dürfen sich unterscheiden, sollten jedoch überwiegend Gemeinsamkeiten aufweisen. Genau diese Bedingung ist jedoch im Bereich der Softwareentwicklung und mathematischen Modellbildung in den seltensten Fällen erfüllt.

Will man eine Vielzahl möglicher Gebäude mit einem gemeinsamen Konstruktionsplan beschreiben, erkennt man, dass kaum eine einzelne Komponente invariant bleibt. Die Anzahl frei zu bleibender Lücken im Konstruktionsplan sowie die mögliche Anzahl von Varianten pro Konstruktionslücke verhindern eine effektive Nutzung des Konzeptes der Produktfamilien.

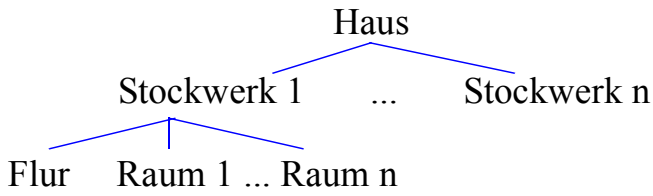


Abbildung 6: Hierarchische Gebäudestruktur

Andererseits kann das mathematische Modell eines Gebäudes in eine hierarchische Struktur untergliedert werden, die in praktisch allen Anwendungsfällen wiederkehrt. Ein Gebäude besteht aus einer variablen Anzahl von Stockwerken. Diese besitzen eine variable Anzahl von Räumen, aber nur einen gemeinsamen Flur (Abb. 6).

Im Gegensatz zu einem lückenbehafteten, starren Konstruktionsplan sollte die Variantenvielfalt der möglichen Modellbeschreibungen erfasst werden können. Hierzu sind zunächst variante und invariante Teile der Struktur zu identifizieren. Die Variabilität der varianten Teile ist

durch sogenannte generische Parameter zu beschreiben. Diese könnten im Idealfall automatischen Generatoren dazu dienen, die varianten Teile für einen konkreten Anwendungsfall zu erzeugen.

Die Erstellung eines konkreten mathematischen Modells erfolgt dann nach Festlegung des Wertes der generischen Parameter durch Instantiierung des generischen Konstruktionsplans.

Etabliert im Bereich der Modellbildung ist die Wiederverwendung auf Komponentenbasis. Hierbei muss es sich nicht ausschließlich um atomare Komponenten handeln. Gerade die Wiederverwendung komplexerer Teile eines Gesamtmodells, die einen vollständigen Teilprozess repräsentieren, ist besonders lukrativ und ist anzustreben.

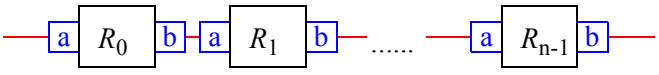
Orientiert sich die Abgrenzung der Teilprozesse an der physikalischen Anschauung, ist meist eine intuitive Zuordnung der Teilprozessmodelle zu Produktfamilien möglich. So können beispielsweise Komponentenmodelle, die Stockwerke, Räume oder Wände, technische Anlagen oder Anlagenkomponenten beschreiben, als Produktfamilie aufgefasst werden.

Ihre Variabilität ergibt sich beispielsweise durch unterschiedliche geometrische Abmessungen oder unterschiedliche Betriebsdaten. In diesen Fällen kann ihre Bandbreite durch einfache Parametrierung abgedeckt werden. Bei der Instantiierung erfolgt dann die Generierung des individuellen, unterscheidbaren mathematischen Modells.

Unterscheiden sich technische Prozesse nur in einzelnen Teilprozessen, können die Variabilitäten durch vorab definierte Lücken bzw. durch den Austausch einzelner Komponentenmodelle an festen Schnittstellen abgedeckt werden. Moderne Engineering-Tools erlauben durch graphische Editoren leicht den Austausch einzelner Komponentenbausteine in komplexen Modellen.

Weiter führende generische Konzepte werden bislang nur in Einzelfällen eingesetzt. Insbesondere strukturvariante Systeme werden meist durch mehr oder minder vollständige Dekomposition, Selektion geeigneter Komponentenbausteine und Komposition behandelt.

Es existieren in modernen mathematischen Simulationssprachen aber durchaus Konzepte, häufig wiederkehrende Strukturen (Modellbausteine) mittels einer Schleifendeklaration (Abb. 7) automatisch zu erzeugen und auch sukzessive miteinander zu verschalten. Die Anzahl der zu instantiierenden Modellkomponenten ist bei der Modellerstellung als Parameter festzulegen [Til-01].




```
for i in 1 : (n - 1) loop
  connect ( R[i].cut_a , R[ i-1].cut_b ) ;
end for;
```

Abbildung 7: Beispiel einer strukturerzeugenden Schleife in *Modelica*. Es wird eine Widerstandskette aus n Widerständen erzeugt und in Reihe geschaltet. Die Zahl n dient als generischer Parameter.

2.4 Top-Down- und Bottom-Up-Strategie

In der Literatur findet man zwei prinzipiell unterschiedliche Vorgehensweisen (Abb. 8) zur Systemanalyse. Die Darstellung der grundlegenden Prinzipien folgt Litz ([Lit-01a]). Während das Top-Down-Verfahren auf einer Zerlegung und zunehmenden Granularität der Modellbeschreibung basiert, wird das Bottom-Up-Verfahren durch Verknüpfung elementarer Bausteine charakterisiert.

Trotzdem ist ein rein bausteinorientierter Ansatz zunächst methodisch nicht ausreichend zur Realisierung der Vereinfachungen bzw. Detaillierungen der *Bottom-Up*- bzw. *Top-Down*-Strategie. Erst durch das Klassenkonzept (siehe Abschnitt 3.1.2) der objektorientierten mathematischen Modellbildung lassen sich die Konzepte realisieren.

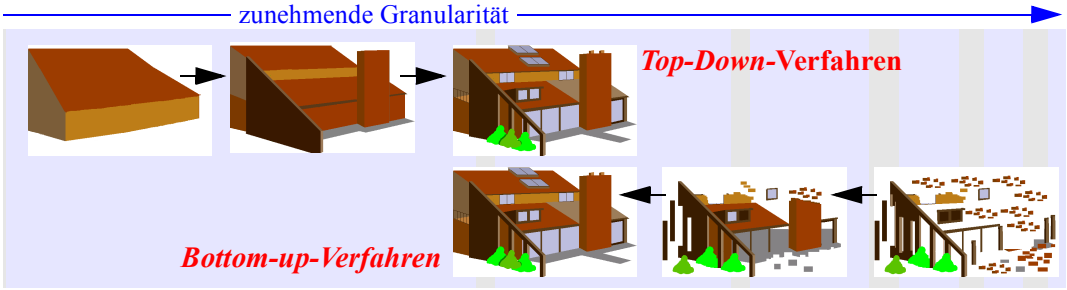


Abbildung 8: *Bottom-up*- und *Top-down-Verfahren*

2.4.1 Top-Down-Strategie

Die *Top-Down*-Strategie startet mit einem möglichst einfachen Modell. Dieses ist durch die Zerlegung in Teilsysteme zu strukturieren und zu detaillieren.

Ausgehend von nur wenigen miteinander verschalteten Komponenten geringer Komplexität (Abb. 8) wird die Granularität der Modellbeschreibung stetig erhöht. Aus der Analyse der erhaltenen Simulationsergebnisse entscheidet man sich zur Detaillierung des Modells an bestimmten Stellen und geht im Entwurfsprozess an die entsprechende Stelle zurück.

Die Vorteile der *Top-Down*-Strategie liegen in einer hohen Entwurfseffizienz. Man hat sehr früh ein nutzbares Gesamtmodell zur Verfügung. Seine Komplexität ist verfahrensbedingt so niedrig wie möglich. Man erkennt durch die iterative Detaillierung sehr schnell Einfluss und Wirkung der hinzukommenden Größen.

Die nachträglichen Detaillierungen der *Top-Down*-Strategie erfordert allerdings neben der Hinzufügung von Bausteinen des Gesamtmodells häufig die Modifikation des vorhandenen Baustein-Repertoires.

Man kann diese Problematik durch Bibliotheksstrukturen lösen, welche Komponenten gleicher Funktionalität in unterschiedlichen Detaillierungsgraden vorhalten. Aus methodischer Sicht ist dies keine gute Lösung. Aus den Mehrfachimplementierungen folgt ein hoher Erstellungsaufwand sowie eine schwierige Modellpflege. Erkennt man, dass die gemeinsame Funktionalität von mehrfach implementierten, in unterschiedlichen Detaillierungsgraden beschriebenen Modellen zu ändern ist, so kann dies nicht in einem Schritt erfolgen.

Eine bessere Lösung dieser Problematik liefert die später vorgestellte Methode des objektorientierten Entwurfs (Abschnitt 3.1). Sie verfügt über eine Vererbungshierarchie analog dem Klassenprinzip der objektorientierten Programmierung. So ist es möglich, Modellklassen der Komponentenmodelle längs einer Vererbungshierarchie sukzessive zu detaillieren.

2.4.2 Bottom-Up-Strategie

Die *Bottom-Up*-Strategie erstellt ein möglichst genaues, detailliertes Modell durch Komposition elementarer Bausteine (Abb. 8). Während des Entwicklungsprozesses steht kein lauffähiges Gesamtmodell zur Verfügung.

Die Vorgehensweise wird unterstützt durch die Nutzung von Komponentenbibliotheken mit vorkonfektionierten Komponentenbausteinen. Die Modellerstellung reduziert sich so auf das Auswählen und Verschalten der Komponentenbausteine und ist in kurzen Zeiträumen realisierbar.

Die Gefahr der bausteinorientierten Modellerstellung, welche vorwiegend vorkonfektionierte Komponentenbausteine aus einer Bausteinbibliothek einsetzt, besteht darin, sehr schnell unnötig komplexe Gesamtsysteme zu erhalten. Von großer Bedeutung für die Entwurfseffizienz ist es daher, während der Entwicklung das Modell auf die relevanten Aspekte zu beschränken und mit einer möglichst geringen Zahl von Komponentenbausteinen zu beschreiben. Da kein lauffähiges Gesamtmodell existiert, können heuristische Vorgehensweisen bei der Entscheidung, welche Komponenten notwendig sind, nicht angewendet werden. Eine Zuordnung von Ursache und Wirkung bezüglich einzelner Bausteine ist sehr schwierig.

Erst nach Erstellung des Gesamtmodells kann aufgrund der Analysen von Simulationsergebnissen das Gesamtsystem nachträglich vereinfacht werden. Es genügt hierbei häufig nicht, Komponentenbausteine zu entfernen. Vielmehr ist es erforderlich, Vereinfachungen und damit Modifikationen einzelner Komponenten vorzunehmen. Es muss daher wie im Falle der Top-Down-Strategie im Entwurfsprozess auf die Komponentenebene zurückgegangen werden. Auch hier hilft die Methode des objektorientierten Entwurfs (Abschnitt 3.1) weiter. Ähnlich der Detaillierung durch Vererbung können innerhalb einer Vererbungshierarchie (in modernen Simulationssprachen) auch Komponenten ausgetauscht werden und so auch durch vereinfachte Bausteine ersetzt werden.

Hier sollte man jedoch darauf hinweisen, dass die hohe Komplexität in der Regel nicht vom Anwender zu beherrschen ist. Denn sie wurde nur durch die für den Anwender intuitive und somit auch transparente Verschaltung vorhandener Komponenten erzeugt. In der Regel wird das hinterlegte Modell hoher Komplexität dem Anwender sogar verborgen bleiben. Es dient dem Rechner zur Erzeugung eines Simulationsmodells. In diesem Fall verliert die Notwendigkeit der Vermeidung zu großer Modelle an Bedeutung. Moderne Rechnersysteme erlauben die Behandlung auch sehr hoher Modellordnungen. Moderne Simulationswerkzeuge können durch symbolische Transformationen große Modelle redundanter Komplexität vor Beginn der eigentlichen Simulation auf die notwendige Modellgröße reduzieren [Elm-93].

2.5 Nutzung von Modellbibliotheken

Erst der Aufbau einer Bibliotheksstruktur erlaubt die systematische Einbeziehung der Wiederverwendung als Entwurfswerkzeug. Ziel wird es sein, instantiierbare Komponentenmodelle, komplette Teilprozessmodelle, aber auch fertige Gesamtmodelle zur Wiederverwendung abzulegen.

An dieser Stelle sei auf eine Besonderheit des Entwurfsprozesses in dieser speziellen Anwendungsdomäne hingewiesen. Die physikalischen Ersatzmodelle werden überwiegend vom Anwender entworfen, nicht automatisch generiert. Die Ausprägung der Schritte der Selektion, Adaption und Komposition haben dies zu berücksichtigen.

2.5.1 Selektion und Adaption

Die Dekomposition des realen technischen Prozesses sollte sukzessive erfolgen. Die Vorgehensweise folgt so zunächst der Systemanalyse des *Top-Down*-Ansatzes. Nach jeder Zerlegung ist im Schritt der Selektion zu überprüfen, ob in der Bibliothek kein geeignetes Teilprozessmodell vorhanden ist. Erst im Falle des Nichtauffindens eines passenden oder mit vertretbarem Aufwand adaptierbaren Teilprozessmodells ist die Granularität durch Dekomposition zu erhöhen. Gelangt man zu elementaren Komponentenmodellen sind diese entsprechend einer *Bottom-Up*-Vorgehensweise zu verschalten.

Hier ist bei der Modellerstellung der eingelagerten Komponentenmodelle eine möglichst intuitive Abgrenzung der Teilsysteme, die sich an der physikalischen Anschauung orientiert, unumgänglich. Bauteile, Anlagen, Anlagenkomponenten müssen durch eigene Produktfamilien mathematischer Modelle repräsentiert werden. Reale, technische Schnittstellen (Flansche, Rohrleitungen, Steckverbinder, Kabel, Kontaktflächen oder allgemeine Grenzflächen) müssen auf Schnittstellen der Modelle abgebildet werden.

Neben der Kommentierung und sinnvollen Benennung unterstützen graphische Entwicklungsumgebungen die Selektion. Mnemotechnische Icons erlauben eine intuitive und schnelle Selektion. Denn zum einen kann der Anwender geometrische Formen einfacher Graphiken schneller erfassen als Texte, zum anderen repräsentieren sie einfache Abbilder des technischen Teilprozesses. Gute mnemotechnische Icons sind ideale Repräsentanten von Produktfamilien.

Bei der Erstellung eines Modells entwirft man ein Abbild der realen Welt. Zunächst sind im Schritt der Abstraktion unwichtige Aspekte des realen Prozesses auszuklammern. Dann sind typische Eigenschaften aufzufinden, die es ermöglichen, den Prozess in eine Menge gleichartiger Fälle einzureihen. Diese gleichartigen Fälle werden auf ein gemeinsames Modell abgebildet.

Genau die gleichen Tätigkeiten sind bei der Erstellung eines guten graphischen Icons erforderlich. Auch ein graphisches Icon ist ein - wenn auch mit sehr geringem Informationsgehalt ausgestattetes - Modell eines realen Prozesses. Es zeigt die charakteristischen Gemeinsamkeiten auf und klammert die Variabilitäten aus. Dies unterstützt den Anwender bei der Selektion. Denn er wird zunächst versuchen sein aktuelles Problem anhand von Gemeinsamkeiten einer vorhandenen Produktfamilie zuzuordnen.

Falls sich die Designregeln der mnemotechnischen Icons an die in der Anwendungsdomäne üblichen graphischen Formensprachen (Schaltbilder, Flusspläne, technische Zeichnungen) anlehnen, wird der Anwender sie schnell erfassen. Er wird sogar seinen aktuellen technischen Prozess durch eine ähnliche graphische Darstellung schon abgebildet haben. Die Zuordnung

geeigneter Komponentenmodelle aufgrund ihrer graphischen Darstellung erfolgt dann nahezu intuitiv.

Nach der Selektion eines geeigneten Komponentenmodells ist dieses in das Gesamtmodell zu integrieren und individuell zu parametrieren.

Man kann jedoch nicht immer vom Auffinden eines passenden Komponentenbausteins ausgehen. Eventuell lassen sich aber existierende Komponentenmodelle durch Modifikationen anpassen oder zumindest Teile von ihnen wieder verwenden. Ansonsten bleibt nur der Neuentwurf. Was zu tun ist, kann nur anhand der Ähnlichkeit der ursprünglichen Anforderung, die der Entwicklung der vorhandenen Komponente zugrunde lag, und der momentanen Anforderung entschieden werden. Diese Abwägung erfolgt vorwiegend intuitiv aufgrund der Erfahrung des Modellentwicklers.

Die Einführung moderner Methoden des Softwareengineering können helfen solche Entscheidungsprozesse zu objektivieren und nachvollziehbar zu machen. Beim Softwareengineering geht es um Techniken, Methoden und Werkzeuge zur systematischen Konstruktion oder Fertigung von Software sowie ihrer Dokumentation nach ingenieurgemäßen Prinzipien. Dies bedeutet die Entwicklung mit abschätzbarem Personal- und Zeitaufwand bei vorgegebenen Qualitätsanforderungen. Grundlage hierbei ist die Erfassung von Messwerten während des Entwicklungsprozesses.

Übertragen auf die Modellerstellung bedeutet dies, dass zunächst bei jedem neuen Modellentwurf, aber auch bei Anpassungen vorhandener Modelle der notwendige Aufwand zu registrieren ist. Diese Daten dienen dann als Basis, den zu erwartenden Aufwand bei einem aktuellen Projekt abzuschätzen. Hierzu muss man allerdings wissen, ob die zurückliegenden Projekte und das aktuelle Projekt vergleichbar sind.

Der Begriff der Vergleichbarkeit kann auf die Gemeinsamkeiten und Unterschiede der vorliegenden und der zu erstellenden Produktfamilien zurückgeführt werden. Bei der Modellerstellung liegt jedoch nur die initiale Anforderungsbeschreibung des aktuellen Projektes vor. Nur ein Vergleich dieser mit den initialen Anforderungsbeschreibungen der zurückliegenden Projekte kann als Maßstab der Vergleichbarkeit dienen. Formal spezifizierte Anforderungsbeschreibungen erlauben hierbei eine Objektivierung der Entscheidung.

Man kann die Auswahl vorhandener Komponenten und die Entscheidung bezüglich eines Neuentwurfes auch durch geeignete Kommentierung vorhandener Komponenten bei der Ablage zur Wiederverwendung unterstützen. Hierzu gehören Beschreibungen typischer Einsatzgebiete und Anwendungsbeispiele zum Einsatz der Komponenten.

2.5.2 Komposition

Bei der Komposition werden die instantiierten, individuell parametrierbaren Komponentenmodelle zu einem bereits lauffähigen Gesamtmodell verschaltet. Als Orientierung dienen hierbei die physikalischen Schnittstellen des betrachteten Systems.

Die genutzten, vorkonfektionierten Komponentenmodelle sollten in möglichst vielen exemplarischen Konfigurationen oder Anwendungsfällen vorhergehender Projekte erprobt sein. So kann die Wahrscheinlichkeit eines Fehlers beim erneuten, ähnlichen Einsatz als gering angenommen werden.

Im Falle eines alleinigen Einsatzes bereits validierter Komponenten ist nur deren Interaktion zu überprüfen. Die erneute Verifikation auf Komponentenebene kann so entfallen. Gerade typische Fehler einer Neuentwicklung bzw. Neuimplementierung werden so vermieden und Entwicklungszeit wird eingespart. Im Idealfall wird es so möglich, ein Gesamtmodell zu erstellen, ohne in die Komponentenebene des V-Modells absteigen zu müssen. Dies befreit jedoch nicht von der Notwendigkeit einer Validierung des neuen Gesamtmodells.

2.6 Wiederverwendungsorientierter Regelungsentwurf

Eine wesentliche Motivation zur Erstellung mathematischer Modelle von technischen Prozessen ist der Entwurf von Regelungs- und Steuerungsalgorithmen. Die Aufgabe der Modelle ist es hierbei, das dynamische Verhalten des technischen Prozesses nachzubilden, um bei der Festlegung der Reglerstruktur, der Parametrierung sowie der Auswahl des geeignetesten Algorithmus zu unterstützen [Lit-01a] (siehe Abb. 9).

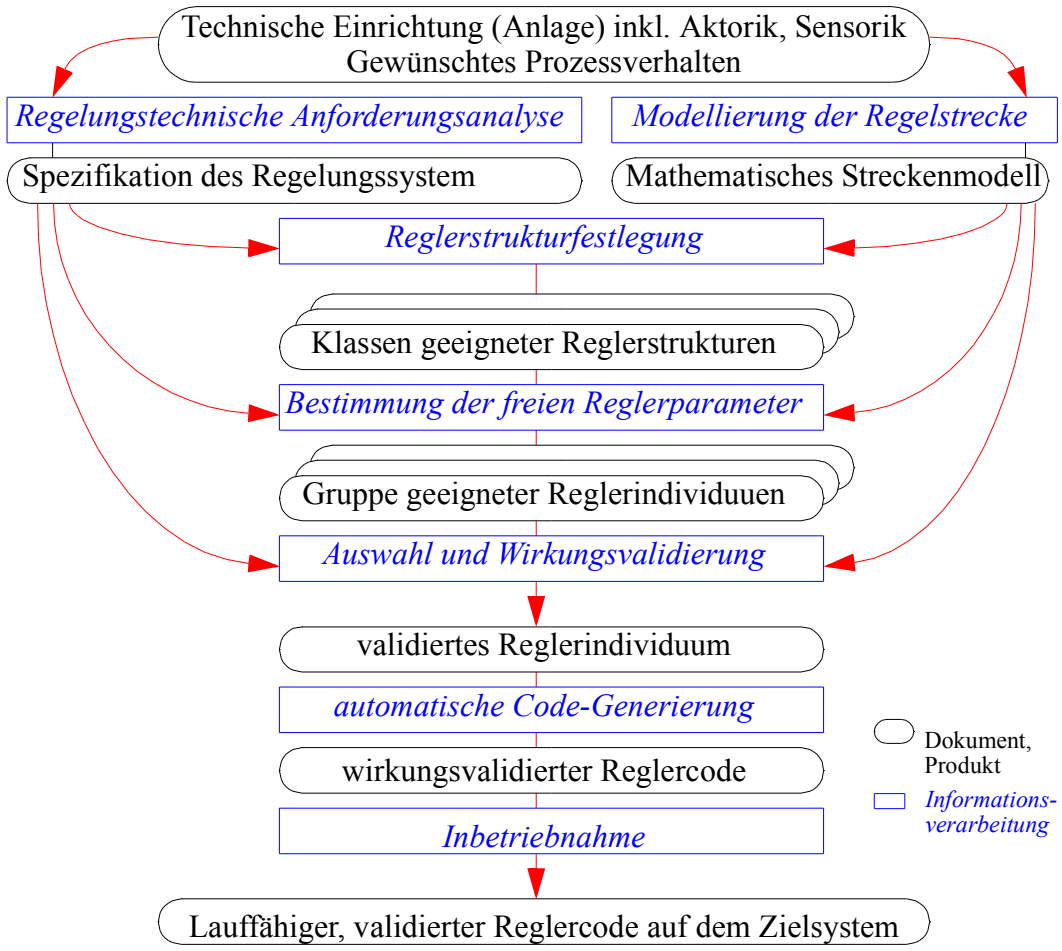


Abbildung 9: Instanzennetz eines Reglerentwurfsprozesses nach Litz [Lit-01a]

Auch werden erst durch das mathematische Modell simulative Wirkungsvalidierungen möglich. Hierzu ist es von großem Vorteil, wenn sowohl der Regelungsalgorithmus als auch das mathematische Streckenmodell unter einer gemeinsamen Simulationsumgebung implementiert werden können. Die Modellbibliotheken müssen daher in einer separaten Bibliothek auch Algorithmenbausteine enthalten, welche in ein Gesamtmodell integriert werden können. Gerade im Falle der Algorithmen bietet sich aufgrund der häufig identischen Anforderungen an eine Regelung oder Steuerung die Nutzung wiederverwendungsorientierter Entwurfsmethoden an.

Bei technischen Prozessen möchte man üblicherweise ein oder mehrere messbare Größen des physikalischen Systems auf vorgegebenen konstanten oder auch zeitlich veränderlichen Werten halten. Zudem sollen die Einflüsse äußerer Störungen auf die zu regelnde Größe kompensiert werden. Dies sind die klassischen Aufgaben einer Regelung. Die Struktur eines geschlossenen Regelkreises ist in Abb. 10 dargestellt.

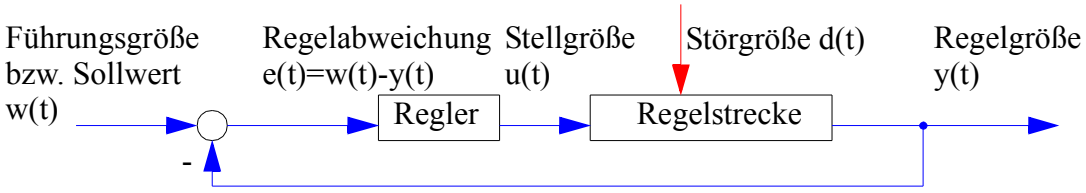


Abbildung 10: Ein geschlossener Regelkreis [Föl-94], [Lit-01a], [Lun-96]

Die zu regelnde Größe bezeichnet man als Regelgröße y(t). Diese soll der Führungsgröße w(t) nachgeführt werden. Hierzu ist vom Regler eine Stellgröße u(t) geeignet zu wählen, um die Einflüsse der Störung d(t) zu kompensieren. Als Information zur Erfüllung dieser Aufgabe steht dem Regler die Regelabweichung e(t) = w(t)-y(t) zur Verfügung.

Komplementär hierzu findet man den Begriff der Steuerung (Abb. 11). Die Definition folgt Litz [Lit-01a]. Ähnlich wie der Begriff Regelung sowohl den Regelungsalgorithmus als auch die zu beeinflussende Regelstrecke umfasst, bezeichnet der Begriff Steuerung sowohl das steuernde als auch das gesteuerte System [LiFr-99]. Eine industrielle Steuerung weist meist die Form eines sogenannten Steuerkreises auf. Ein- und Ausgangsgrößen des steuernden Systems bestehen überwiegend aus binären Größen. Analoge Eingangsgrößen werden vor der Weiterverarbeitung auf binäre Größen abgebildet. Es existiert häufig eine Vielzahl von Ein- und Ausgangsgrößen. Die binäre Signalverarbeitung steht im Vordergrund. Es treten vielfach Rückführungen auf. Jedoch wirkt eine bestimmte Größe nicht zwangsläufig auf sich selbst zurück. Der Begriff der Kreisverstärkung und damit das klassische Stabilitätsproblem existiert aufgrund der binären Signale nicht. Die Störungsbehandlung ist von wesentlicher Bedeutung beim Entwurf eines Steuerungssystems. Im Gegensatz zu einer Regelung führt das Auftreten unvorhergesehener Störungen zu gravierenden Problemen. Man versucht daher eine möglichst umfassende Behandlung aller vorhersehbaren Störungen vorzunehmen.

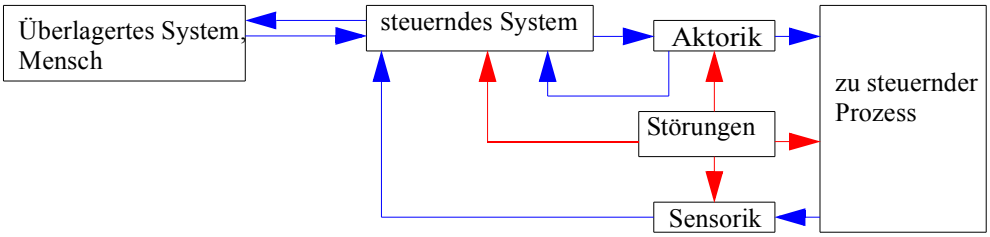


Abbildung 11: Struktur eines Steuerkreises nach Litz [Lit 01a]

Es war nicht Ziel dieser Arbeit neue, höhere Regelungskonzepte zu entwickeln, sondern es wurden ausschließlich Regelungsbausteine entworfen um auch geregelte oder gesteuerte komplexe Anlagen und Prozesse beschreiben zu können.

Im Rahmen einer wiederverwendungsorientierten Entwurfsmethode wurde in dieser Arbeit das bausteinbasierte Konzept der parametrierbaren, komponierbaren Grundalgorithmen (PKG) genutzt.

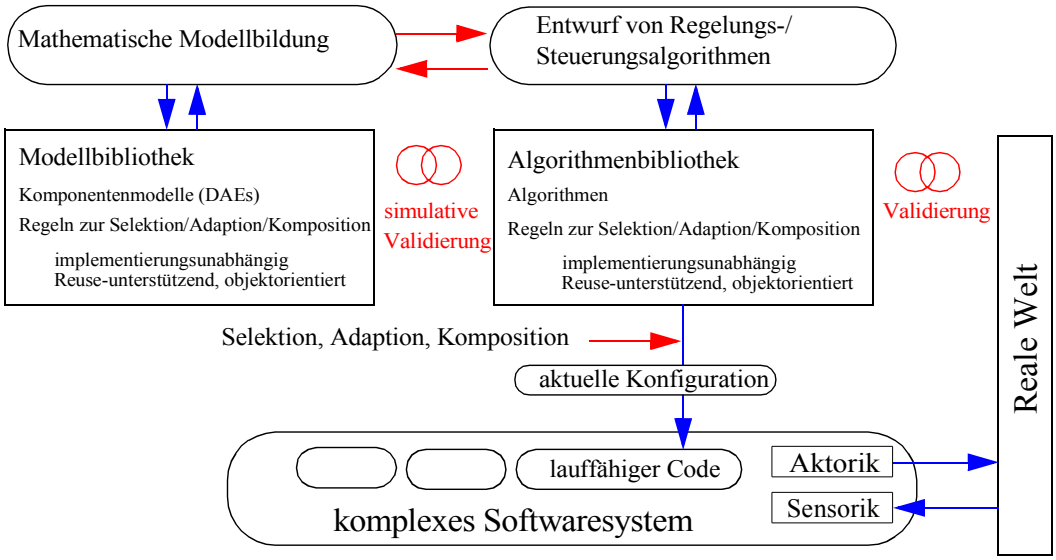


Abbildung 12: Modellbasierte Algorithmenentwicklung unter Nutzung von Bibliotheken [MeLi-00b]

Die in Bibliotheken gespeicherten Algorithmenkomponenten sind zu selektieren, an die konkreten Anforderungen zu adaptieren und zu einem Gesamtalgorithmus zu komponieren (Abb. 12). Zur Selektion des geeigneten Algorithmus dient das mathematische Modell des zu beeinflussenden Prozesses. Es ermöglicht eine detaillierte mathematische Analyse der Wirkungszusammenhänge sowie simulative Tests. Hierauf basiert auch die Adaption der Algorithmen sowie ihre Komposition zu einem strukturierten Gesamtalgorithmus.

3 Konzepte zur Strukturierung

3.1 Objektorientierte mathematische Modellierung

Die Methode der objektorientierten mathematischen Modellierung etabliert sich zunehmend bei der Behandlung großer Systeme (z.B. [Bea-00], [BaWi-00], [Cel-91], [CSS-00a,b], [KFS-95], [Ott-99a,b,c,d], [OEM-99a,b], [OtBa-99a,b], [OtSc-99], [Ott-00], [Tum00a,b,c], [TuTi-00] [TKE-97]). Sie stellt den Schlüssel zur hierarchischen Strukturierung komplexer Systeme dar.

Die Vorgehensweise orientiert sich stark an der objektorientierten Softwareentwicklung. Viele Begriffe und Definitionen erinnern an Komponenten des *UML*-Klassendiagramms [SeGu-00]. Die wesentlichen Charakteristika und ihre besondere Ausprägung in der objektorientierten mathematischen Modellbildung werden im folgenden zusammengestellt.

3.1.1 Modularität

Die sukzessive hierarchische Dekomposition komplexer Systeme in Komponenten orientiert sich in der mathematischen Modellbildung an der physikalischen Anschauung. Sie benötigt jedoch meist auch viel Erfahrung und teilweise Intuition. Einen guten Anhaltspunkt zur Dekomposition bietet hierbei die Strukturierung des realen Prozesses.

Eine sinnvolle Aufspaltung eines Systems sollte gewisse Randwerte berücksichtigen. So sollten die Teilsysteme unabhängig voneinander bearbeitbar sein. Die Definition von klaren Schnittstellen oder Übergabepunkten muss möglich sein. Eine zu hohe Anzahl von Subsystemen reduziert die Übersichtlichkeit. Eventuell hilft sogar eine weitere Untergliederung der Subsysteme, bis man zu elementaren Komponenten gelangt, die mehrfach im Gesamtsystem auftauchen.

Dies stellt einen ersten Schritt zum Einsatz der Wiederverwendung als Entwurfsmethode dar. Anhand der Dekomposition des Systems können so bereits entwickelte Komponentenbausteine aus zurückliegenden Projekten oder bereits gelösten Problemen des aktuellen Projektes ausgewählt, angepasst und zu einem Gesamtsystem verschaltet werden. Die Kunst hierbei ist die Art und Weise, wie die Komponenten innerhalb eines Projektes zu entwerfen sind, um eine möglichst häufige Wiederverwendbarkeit zu garantieren.

Diese Vorgehensweise ist auch Voraussetzung für die Automatisierung des Weges vom vollständigen physikalischen Ersatzmodell zum lauffähigen Simulationsmodell. Hierzu ist innerhalb der Komponenten das physikalische Wissen in einer computerlesbaren, formalen Spezifikation zu hinterlegen. Nach der Komposition des Gesamtmodells kann so das implementierte Rechenmodell automatisch generiert werden.

Eine auf die Wiederverwendung von Komponentenbausteinen gestützte Modellerstellung wird eine Vorgehensweise analog dem *Bottom-up*-Verfahren favorisieren. Von Bedeutung für die Effizienz des Verfahrens wird hierbei die Vermeidung zu hoher Modellordnungen sowie die Durchführung einer nachträglichen Modellreduktion sein.

3.1.2 Klassenprinzip

Die Terminologie orientiert sich an der objektorientierten Softwareentwicklung, wie sie beispielsweise im *UML*-Klassendiagramm genutzt wird [SeGu-00]. Dort beschreibt eine *Klasse* eine Menge von *Objekten* mit gemeinsamem Verhalten und gemeinsamen Eigenschaften. Ein

einzelnes *Objekt* einer *Klasse* wird als *Instanz* bezeichnet. Im Rahmen der objektorientierten mathematischen Modellbildung spricht man stattdessen von *Modellklassen*. Deren *Instanzen*, also die *Objekte,* werden als *Modelle* bezeichnet.

Es besteht ein wesentlicher Unterschied zwischen *Modellklassen* und *Modellen*. *Modellklassen* dienen als formale Spezifikation der physikalischen Eigenschaften und des Verhaltens. Sie stellen computerlesbare Wissensspeicher dar, die bei der *Modell*erstellung genutzt werden. Erst die *Modelle* als *Instanzen* der *Modellklassen* erlauben nach automatischer Codegenerierung die Simulation.

Neben ihrem Namen oder *Bezeichner* verfügt eine *Klasse* über *Attribute* und *Operationen*. Die *Attribute* in der mathematischen Modellbildung sind letztendlich *Konstanten-*, *Parameter-* und *Variablendeklarationen*. Die Rolle der *Operationen* übernehmen *mathematische Gleichungen*.

Objekte, also auch *Modelle,* unterscheiden sich durch individuelle *Attribute*. Während in der *Modellklasse* allenfalls der Typ, der Wertebereich und eventuell *Default*-Werte definiert werden, besitzen die instantiierten *Modelle* individuell unterscheidbare Werte. Ihre *Parametrierung* ist somit ein wesentlicher Bestandteil der *Instantiierung* zur Erzeugung unterscheidbarer *Modelle*.

Darüber hinaus gibt es innerhalb der *Modellklasse* vollständig definierte und parametrierte *Attribute,* die allen *Objekten* der *Klasse* gemein sind. So können beispielsweise allgemeingültige physikalische Konstanten oder gemeinsam genutzte Stoffdaten allen betroffenen *Objekten* zur Verfügung gestellt werden. Eine Änderung eines solchen Parameters wirkt sich unmittelbar auf alle *Instanzen* der *Klasse* aus.

Tabelle 1: Analogie der Begriffsbildung

Softwareentwicklung	***Mathematische Modellbildung***
Klasse	*Modellklasse*
Objekt, Instanz	*Modell*
Attribute	*Konstanten-, Parameter- und Variablendeklarationen*
Operationen	*mathematische Gleichungen*

Die *Klassen* können in einem *Klassendiagramm* graphisch dargestellt werden. Innerhalb eines jeden *Klassendiagrammes* können auch *Objekte* eingezeichnet werden. Ein *Klassendiagramm*, das ausschließlich *Objekte* enthält, wird als *Objektdiagramm* bezeichnet. Die Zugehörigkeit des *Objektes* zu einer *Klasse* wird durch Angabe des *Klassennamens* gekennzeichnet. Die *Klassen* oder *Objekte* sind durch Verbindungslinien miteinander verknüpft. Längs dieser Verbindungslinien werden Informationen ausgetauscht. Als Spezialfälle von Verbindungslinien können gerichtete Linien existieren, bei denen die Informationsflussrichtung festgelegt wird. Das *Klassen-* oder *Objektdiagramm* beschreibt so neben der Untergliederung des Gesamtsystems vorwiegend die Topologie. Dies beschränkt sich jedoch auf die Verbindungsstruktur der Komponenten innerhalb einer Hierarchieebene.

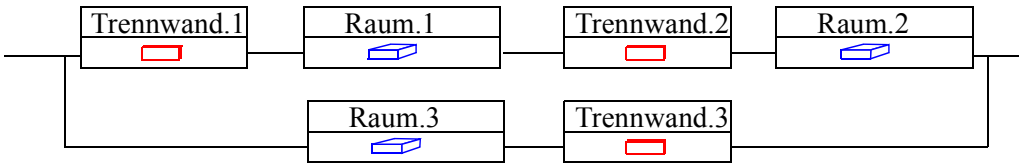


Abbildung 13: Objektdiagramm zur Beschreibung eines Gebäudes mit 3 Räumen

3.1.3 Aggregationshierarchie

Erst die durch die Aggregation erzeugbare hierarchische Strukturierung der *Klassen-* bzw. *Objektdiagramme* erlaubt die sukzessive Detaillierung der Modellbeschreibung bis zu atomaren, d.h. nicht mehr weiter zerlegbaren Subsytemen, denen eine physikalische Verhaltensbeschreibung in Form mathematischer Gleichungen zugewiesen werden kann.

Ein objektorientiertes Modellierungssystem soll die Entwicklung und Definition der Komponenten, ihrer Verbindungen und Schnittstellen durch formale graphische Spezifikationen des mathematischen Modells in Form editierbarer, hierarchisch strukturierter *Objektdiagramme* unterstützen. Hierarchisch bedeutet, dass jede der Komponenten des *Objektdiagrammes* selbst wieder durch *Objektdiagramme* definiert werden kann. Man gelangt so beim Abstieg längs der hierarchischen Struktur durch immer feinere Untergliederung zu den atomaren Komponenten. Erst diese sind rein textuell implementiert. Sie enthalten beispielsweise Variablendeklarationen, algebraische Gleichungen bzw. Differentialgleichungen in einer geeigneten Implementierung. Die Gleichungen der atomaren Komponenten werden im folgenden als Komponentengleichungen bezeichnet.

Die Einordnung der Objekte in eine Aggregationshierarchie wird in Abb. 14 veranschaulicht. Es wird die *UML*-Notation genutzt. Vernachlässigt werden innerhalb der Darstellung dieses Hierarchiebaumes die topologischen Beziehungen der Komponenten auf gleicher Hierarchieebene. So besteht ein Gebäude aus 1 bis n Stockwerken und einem Dach. Jedes Stockwerk wiederum besteht aus 1 bis m Räumen, 1 bis j Innenwänden und 1 bis k Außenwänden. Alle Komponenten außer den *Blättern* der Struktur können allerdings graphisch durch eigene *Objektdiagramme* beschrieben werden. Diese enthalten dann die topologische Information. Nur in den elementaren Wärmewiderständen und Wärmespeichern sind ausschließlich physikalische Gleichungen zu hinterlegen.

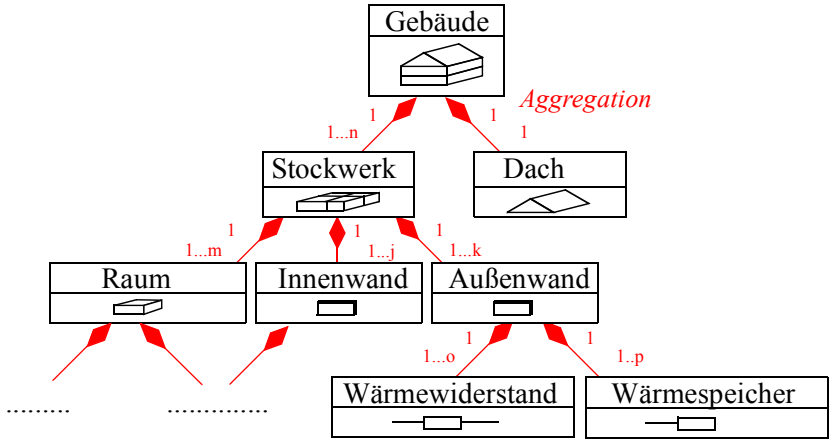


Abbildung 14: Beispiel einer Aggregationshierarchie

3.1.4 Vererbungshierarchie

Von entscheidender Bedeutung für die Strukturierung des Entwurfes der Komponentenmodelle ist die *Vererbungshierarchie*. Die Relation wird häufig auch *Verallgemeinerung* bzw. in umgekehrter Richtung auch *Spezialisierung* genannt. Hierbei geht es nicht um die Strukturierung des Gesamtmodells, sondern um die abstraktere Strukturierung von Eigenschafts- bzw. Verhaltensbeschreibungen der Komponentenmodelle. Im Besondern können mehrfach genutzte Eigenschafts- bzw. Verhaltensbeschreibungen geeignet strukturiert spezifiziert werden.

Durch *Vererbung* stehen den abgeleiteten Modellklassen neben ihren eigenen Eigenschaften auch diejenigen des Basistyps zur Verfügung. Dies können neben allgemeinen Konstanten-, Parameter-, Variablen- und Schnittstellen-Deklarationen auch gemeinsam benötigte, lokale Komponentengleichungen sein. So kann es vorkommen, dass eine übergeordnete Modellklasse, die beispielsweise nur die Aufgabe hat, gemeinsame Stoffdaten an ihre abgeleiteten Modellklassen weiterzugeben, unvollständig ist. Dies bedeutet: es können aus ihr keine lauffähigen Modelle instantiiert werden. Man spricht dann von einer *abstrakten Modellklasse*.

Im Beispiel (Abb. 15) vererbt die generalisierte *Modellklasse Wand* an die Spezialisierungen *Außenwand* sowie *Innenwand* die von beiden benötigte Wärmeleitfähigkeit λ. Für den Wärmeübergangskoeffizienten α nutzen beide eigene Definitionen.

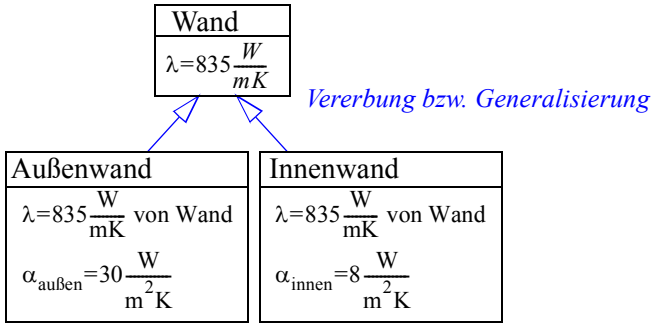


Abbildung 15: Beispiel einer Vererbungshierarchie

Das Konzept der Vererbung hat, wie die Bezeichnungen *Spezialisierung* bzw. *Generalisierung* schon implizieren, Auswirkungen auf die methodische Vorgehensweise bei der Modellerstellung.

Es ist so beispielsweise möglich, eine bereits einfache Modellbeschreibung nachträglich zu detaillieren und trotzdem die volle Funktionalität der vererbenden Komponente weiter zu nutzen. Dies ermöglicht die Realisierung einer *Top-Down*-Vorgehensweise.

Weitaus schwieriger zu realisieren ist die nachträgliche Modellvereinfachung der *Bottom-Up-Methode*. Hierbei ist letztendlich die Komplexität einzelner Komponenten nach Modellerstellung zu reduzieren. In der Regel wird dies nicht durch einfaches Entfernen von Bausteinen zu bewerkstelligen sein. Aber auch hier bietet die Vererbungshierarchie eine Lösung. Moderne Simulationssprachen zur objektorientierten, mathematischen Modellbildung besitzen die Möglichkeit eines Austausches einzelner Module einer Modellklasse innerhalb einer Vererbungshierarchie. In *Modelica* ist dies beispielsweise über die sogenannte *redeclare*-Anweisung möglich. Mit ihr können Komponenten einer Beschreibung innerhalb einer Vererbungshierarchie einfach ausgetauscht werden. So kann eine sich als zu detailliert erweisende Struktur auch noch in späteren Entwicklungsschritten durch eine einfachere ersetzt werden.

3.1.5 Kapselung und Schnittstellen

Die Wiederverwendbarkeit der Komponenten (unabhängig von der aufrufenden Umgebung) erfordert ihre konsequente Kapselung. Eine strenge Abtrennung der *Modellklasssen* erfordert die Definition von Schnittstellen. Diese sind meist als eigene Modellklasse definiert und werden von übergeordneten Modellklassen aggregiert. Letztere Modellklassen interagieren miteinander längs der Verbindungslinien der von ihnen aggregierten Schnittstellen.

Die Forderung der Kapselung bezieht sich in der mathematischen Modellbildung vorwiegend auf Variablen. Längs der Verbindungslinien werden die sogenannten Schnittstellenvariablen in Beziehung gesetzt. Diese sind innerhalb der Schnittstellen deklariert und können aufgrund der Aggregationshierarchie durch die jeweils übergeordneten Modellklassen angesprochen werden.

Die konsequente Kapselung einer Modellkomponente erfordert die alleinige Verwendung lokaler Variablen und Schnittstellenvariablen innerhalb ihrer Komponentengleichungen.

3.2 Topologische Systembeschreibungen

3.2.1 Ein Illustrationsbeispiel

In Abschnitt 3.1 stand vorwiegend die Dekomposition eines Systems in Teilsysteme sowie deren hierarchische, vertikale Strukturierung auch unter dem Aspekt der Wiederverwendbarkeit im Vordergrund der Betrachtungen. Hierbei gelangte man zu atomaren Subsystemen, denen eine Verhaltensbeschreibung in Form mathematischer Gleichungen zugewiesen wurde. Im folgenden werden topologische Aspekte beschrieben. Hierbei geht es um die Art und Weise des Informationsaustausches der Subsysteme und ihre Verbindungen. Die verschiedenen Ansätze zur Lösung dieser Problematik führen zu unterschiedlichen Beschreibungskonzepten, die sich in den Anforderungen an anzuwendende Modellierungswerkzeuge widerspiegeln. Dies ist bei der Auswahl einer Simulationsumgebung zu berücksichtigen (siehe Kapitel 4).

Die topologische Analyse führt zur Unterscheidung verschiedener Arten von Systembeschreibungen. Panreck et al. [PJD-94, Pan-99] untergliedern die möglichen Systembeschreibungen in drei Gruppen. Die vorliegende Arbeit folgt dieser Darstellung [Pan-99]. Die Unterschiede werden nach der Vorstellung der drei Gruppen an einem Beispiel (Abb. 16) veranschaulicht, dessen mathematisches Modell nach den drei Arten spezifiziert werden soll (Abschnitte 3.2.2, 3.2.3 und 3.2.4). In Abschnitt 6.1 wird das Beispiel in geeigneten Simulationsumgebungen implementiert und simuliert.

Beispiel

Es soll die Änderung der Raumlufttemperatur als Folge des vorgegebenen zeitlichen Verhaltens der Außentemperaturen auf der Nord- und Südseite eines Gebäudes (T_N bzw. T_S) berechnet werden. Die Aufgabenstellung ist streng kausal.

Geg.: $T_N(t)$, $T_S(t)$
Ges.: $T_L(t)$

Abkürzungen

N (Nordseite, außen)
S (Südseite, außen)
NW (Nordwand)
SW (Südwand)
L (Raumluft)

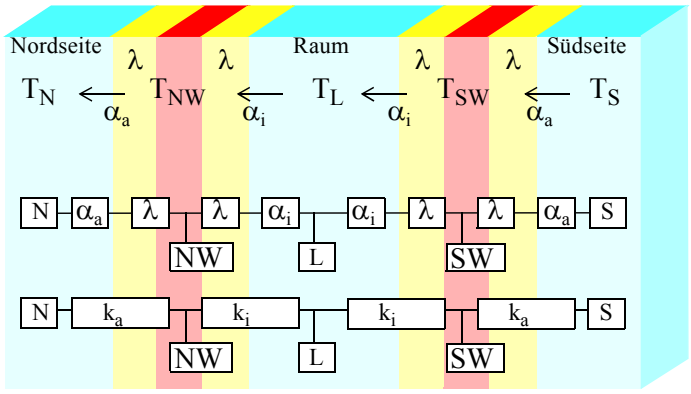


Abbildung 16: Beispiel zur Beschreibung des thermischen Verhaltens eines Einraumhauses mit unterschiedlichen Außenlufttemperaturen vor der Nord- und Südfassade.

Der durch eine dünne Wandschicht der Wärmeleitfähigkeit λ, der Dicke s und der Fläche A fließende Wärmestrom j ergibt im stationären Fall $j = \lambda A/s \cdot (T_1 - T_2)$, wenn an der Wandschicht längs der Flussrichtung eine Temperaturdifferenz $T_1 - T_2$ anliegt.

Der Wärmeübergang zwischen zwei miteinander über eine Fläche der Größe A in direktem Kontakt stehenden Körpern der Temperaturen T_1 bzw. T_2 wird durch den Wärmeübergangs-

koeffizient α beschrieben. Der Wärmestrom berechnet sich nach der Gleichung $j = \alpha A \cdot (T_1 - T_2)$.

Die Änderung der inneren Energie Q eines quellen- und senkenfreien Körpers erhält man aus der Bilanzierung aller ein- und ausfließenden Wärmeströme. Besitzt der Körper die Masse m und die spezifische Wärmekapazität c, korrespondiert dies mit einer Änderung seiner Temperatur T. Es gilt die Beziehung $\sum(j_{ein} - j_{aus}) = mc \cdot dT/dt$.

Die Systembeschreibung soll in Form eines konzentrierten Modells erfolgen. Dies führt zu dem ebenfalls in Abb. 16 gezeigten Ersatzschaltbild. Auf die Darstellung eines Referenzpotentials wurde verzichtet.

Wie das Ersatzschaltbild nahelegt, können Wärmeleitfähigkeit und Wärmeübergang als Reihenschaltung von Widerständen aufgefasst werden und in einem Wärmedurchgangskoeffizienten $k = 1/(1/\alpha + s/\lambda)$ zusammengefasst werden. Der Wärmestrom berechnet sich dann nach der Gleichung $j = kA \cdot (T_1 - T_2)$.

Das mathematische Simulationsmodell wird in den folgenden Abschnitten (3.2.2, 3.2.3, 3.2.4) nach den drei unterschiedlichen Methoden der topologischen Systemstrukturierung spezifiziert.

3.2.2 Signalflussverknüpfung

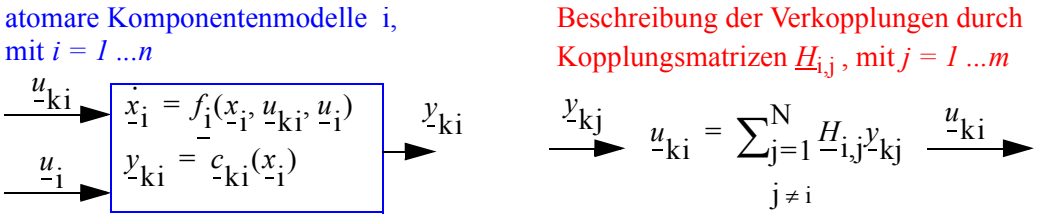


Abbildung 17: Komponentenmodell und Koppelmatrix bei der Signalflussverknüpfung. Die Verkopplungen sind gerichtet und rückwirkungsfrei (nach [Pan-99]).

Die in Abb. 17 gezeigte Zustandsdifferentialgleichung [Lit-83, Pan-99] beschreibt das Verhalten der Komponente i. Der Zustandsvektor ist durch x_i bezeichnet. Die Komponente i besitzt neben äußeren Eingangsgrößen u_i die Koppeleingangsgrößen u_{ki}.

Die Kopplungsmatrizen $H_{i,j}$ beschreiben, wie sich der Koppelausgang y_{kj} der j-ten Komponente auf den Koppeleingang der i-ten Komponente auswirkt. Dies entspricht einer gerichteten, signalflussartigen, rückwirkungsfreien Verkopplung der Komponenten. Die Kopplungsmatrizen werden in der Regel nur mit 0 oder 1 besetzt.

Im allgemeinen können die algebraischen Gleichungen sukzessive aufgelöst werden, und die Simulation kann auf die numerische Integration der verbleibenden Vektordifferentialgleichung reduziert werden [CeEl-93].

Bei der Beschreibung eines physikalischen Systems nach dieser Methode geht man von den bekannten Eingangsgrößen aus, aus denen die unbekannten Ausgangsgrößen berechnet werden. Die Berechnungsvorschrift wird durch ein Blockschaltbild graphisch spezifiziert.

Ein Blockschaltbild besteht aus graphischen Komponenten, d.h. Blöcken, die längs einer Wirkungsrichtung miteinander verschaltet sind. Jeder Block enthält die Vorschrift, wie aus seinen Eingangsgrößen seine Ausgangsgrößen zu berechnen sind. Die Ausgangsgrößen werden addiert oder subtrahiert und entsprechend der Verschaltung als Eingangsgröße in andere Blöcke geleitet.

Abb. 17 zeigt das Blockschaltbild des Beispiels. Es aggregiert N=3 Komponentenbausteine. Jeder der drei Komponentenbausteine enthält eine der drei Differentialgleichungen zur Spezifikation des Zustandsvektors.

Es zeigt sich jedoch, dass es nicht möglich ist, die Komponentenbausteine so zu modellieren, dass sie ausschließlich das Verhalten der abgebildeten Komponente beschreiben. Denn durch die phänomenologischen Gesetze zur Berechnung der Wärmeströme kommt es zu Rückkopplungen. Um trotzdem eine Signalflussverknüpfung anwenden zu können, wurden die phänomenologischen Gesetze direkt in die Differentialgleichungen der Komponentenbausteine eingesetzt.

Die topologische Information der Koppelmatrizen findet sich in Abb. 18 in Form der gerichteten Verbindungslinien zwischen den Komponentenbausteinen wieder.

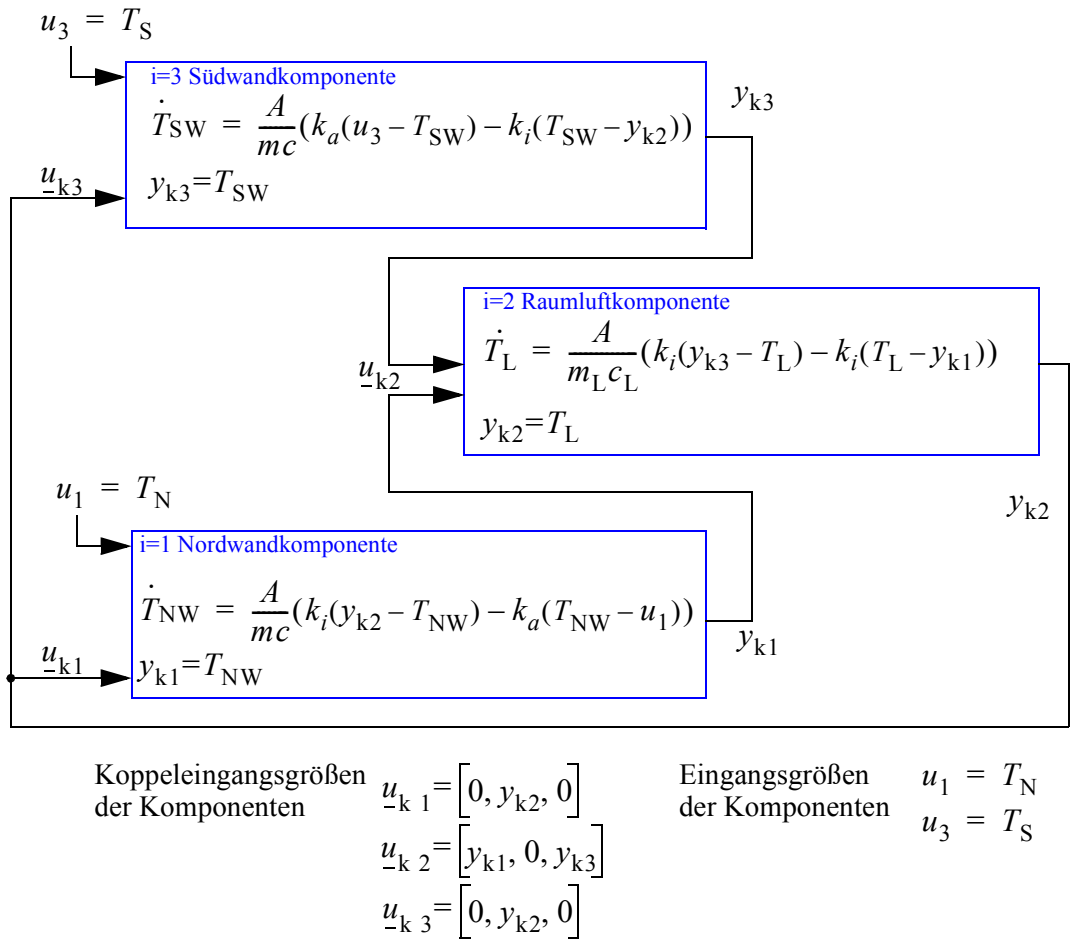


Abbildung 18: Beschreibung des Beispiels mittels der Signalflussverknüpfung

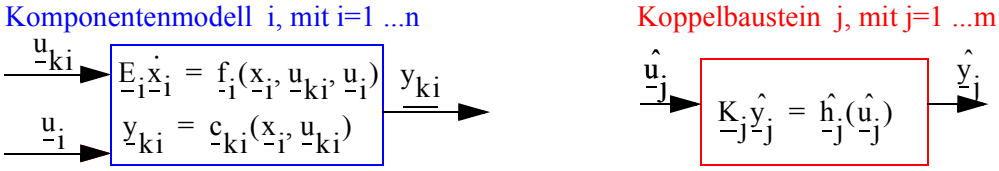
Gut geeignet ist diese Beschreibungsform für alle typischen Signalflusssysteme. Dies sind Systeme, deren Komponenten mit gerichteten, rückwirkungsfreien Verbindungen Informationen austauschen.

Die Berücksichtigung anderer, nicht rückwirkungsfreier Verkopplungen, wie sie häufig aus phänomenologischen Gesetzen folgen, führt zu einer Aufweichung der scharfen Trennung zwischen Koppel- und Komponentengleichungen. Die phänomenologischen Gesetze, müssen, wie im Beispiel (Abb. 18) gezeigt wurde, von den Komponentenmodellen absorbiert werden. Die Transparenz des Ansatzes geht verloren.

Dieser Verbindungstyp bildet die Grundlage blockschaltbildorientierter Simulationssysteme. Haupteinsatzgebiet ist die Regelungstechnik. Graphisch dargestellte Komponentenmodelle mit starrer Festlegung von Ein- und Ausgangsvariablen und klar definierter Wirkungsrichtung werden in Blockschaltbildern verschaltet.

Die Modellerstellung ist recht aufwändig. Zunächst muss eine geeignete mathematische Struktur manuell aus den physikalischen Beziehungen abgeleitet werden. Insbesondere bei großen komplexen Systemen ist dies zeitaufwändig und fehleranfällig. Zudem sind Modelle nur in beschränktem Maße wiederverwendbar; denn schon bei der Modellerstellung muss eine Festlegung aller Eingangs- und Ausgangsgrößen erfolgen. Bei nur leicht veränderter Aufgabenstellung ist ein Neuentwurf notwendig.

3.2.3 Energieflussverknüpfung



Topologiebeziehungen

$$\underline{u}_{ki}^{(1)} = \hat{\underline{B}}_i\left[\hat{\underline{y}}_1^T, \ldots, \hat{\underline{y}}_M^T\right]^T \quad (1)$$

$$\hat{\underline{u}}_j = \hat{\underline{C}}_j\left[\underline{y}_{k1}^T, \ldots, \underline{y}_{kN}^T\right]^T \quad (2)$$

$$\underline{u}_{ki}^{(2)} = \hat{\underline{D}}_i\left[\underline{y}_{k1}^T, \ldots, \underline{y}_{k\,i-1}^T, \underline{y}_{k\,i+1}^T, \ldots, \underline{y}_{kN}^T\right]^T \quad (3)$$

$$\underline{u}_{ki} = \underline{u}_{ki}^{(1)} + \underline{u}_{ki}^{(2)} \quad (4)$$

Abbildung 19: Komponentenmodell und Koppelbaustein bei der Energieflussverknüpfung. Die Verkopplungen sind gerichtet, aber durch das Auftreten algebraischer Schleifen nicht rückwirkungsfrei.
Neu eingeführt werden die Eingangs- und Ausgangsgrößen der Koppelbausteine $(\hat{\underline{u}}_j)$ bzw. $(\hat{\underline{y}}_j)$ (nach [Pan-99]).

Bei der Energieflussverknüpfung werden die überwiegend aus phänomenologischen Gesetzen stammenden physikalischen Verknüpfungen in Form eigenständiger, verhaltensbehafteter Modellbausteine beschrieben [PJD-94].

Die Komponente besitzt wie im Falle der signalflussorientierten Verknüpfung die äußeren Eingangsgrößen $\underline{u}_{\mathrm{i}}$ und die Koppeleingangsgrößen $\underline{u}_{\mathrm{ki}}$. Das atomare Komponentenmodell wird durch die Deskriptormatrix $\underline{E}_{\mathrm{i}}$ ergänzt (Abb. 19). Der Vorteil ist, dass das System weitgehend in seiner ursprünglichen Form belassen werden kann und die Zahl notwendiger manueller Umformungen beim Modellentwurf reduziert wird. Die Deskriptormatrix kann auch singulär sein.

Neu hinzu kommen die Koppelbausteine in Form linearer, impliziter Gleichungssysteme. Als Eingangsgrößen $(\hat{\underline{u}}_{\mathrm{j}})$ dienen meist Potentialgrößen, als Ausgangsgrößen $(\hat{\underline{y}}_{\mathrm{j}})$ Flussgrößen.

Die Verbindungen zwischen den Modellbausteinen werden durch einen Satz Topologiebeziehungen beschrieben. In Abb. 19 spezifizieren die Topologiebeziehungen (1) und (2) Verschaltungen von Komponentenmodellen mit Koppelbausteinen und die Gleichung (3) direkte Signalverschaltungen von Komponentenmodellen untereinander. Üblicherweise sind die Matrizen $(\underline{B}_i, \underline{C}_i, \underline{D}_i)$ nur mit Nullen oder Einsen besetzt und in (4) die einander entsprechenden Elemente von $\underline{u}_{\mathrm{ki}}^{(1)}$ und $\underline{u}_{\mathrm{ki}}^{(2)}$ nicht gleichzeitig ungleich null.

Mit diesem Ansatz lassen sich eine Vielzahl physikalischer Systeme beschreiben. Die Systemtopologie findet sich im Modell wieder und nicht wie bei einem signalflussorientierten Ansatz allein in den Koppelmatrizen.

Die Darstellung des Beispiels unter Anwendung der Energieflussverknüpfung hat die in Abb. 21 gezeigte Form. Die topologischen Gleichungen werden im Blockschaltbild durch gerichtete Verbindungslinien dargestellt. Die Ausgangsgrößen der Koppelbausteine repräsentieren Wärmeströme, die an den Eingängen der Komponentenmodelle addiert bzw. subtrahiert werden müssen.

Da die Eingangsgrößen des Modells nur in Komponentenbausteinen eingekoppelt werden können, müssen zwei zusätzliche Bausteine für die Außenluft auf der Nordseite *(i=4)* bzw. auf der Südseite *(i=5)* hinzugenommen werden.

Die topologischen Beziehungen (Abb. 20) beschreiben die Verknüpfung der Komponentenausgangsgrößen zur Erzeugung der vektorwertigen Koppelgliedeingangsgrößen, die Addition bzw. Subtraktion der Koppelgliedausgangsgrößen zur Berechnung der Komponenteneingangsgrößen.

In der Abb. 20 werden die Topologiebeziehungen analog der Definition (Abb. 19) als vektorwertige Gleichungen zusammengefasst. Eine direkte Signalverschaltung der Komponentenmodelle liegt nicht vor, daher gilt $\underline{u}_{\mathrm{ki}}^{(2)} = \underline{0}$ und $\underline{u}_{\mathrm{ki}} = \underline{u}_{\mathrm{ki}}^{(1)}$.

Eingangsgrößen der Koppelbausteine $\hat{\underline{u}}_j \quad j = 1..4$	Koppeleingangsgrößen der Komponenten $u_{ki}^{(1)} \quad i = 1..5$	Eingangsgrößen der Komponenten $u_i \quad i = 1..5$
$\hat{\underline{u}}_{j=1} = [y_{k1}, 0, 0, 0, y_{k4}]$	$u_{k1}^{(1)} = \hat{y}_2 - \hat{y}_1$	$u_4 = T_N$
$\hat{\underline{u}}_{j=2} = [y_{k1}, y_{k2}, 0, 0, 0]$	$u_{k2}^{(1)} = \hat{y}_3 - \hat{y}_2$	$u_5 = T_S$
$\hat{\underline{u}}_{j=3} = [0, y_{k2}, y_{k3}, 0, 0]$	$u_{k3}^{(1)} = \hat{y}_4 - \hat{y}_3$	
$\hat{\underline{u}}_{j=4} = [0, 0, y_{k3}, 0, y_{k5}]$		

Abbildung 20: Topologiebeziehungen des Beispiels in Energieflussverknüpfung

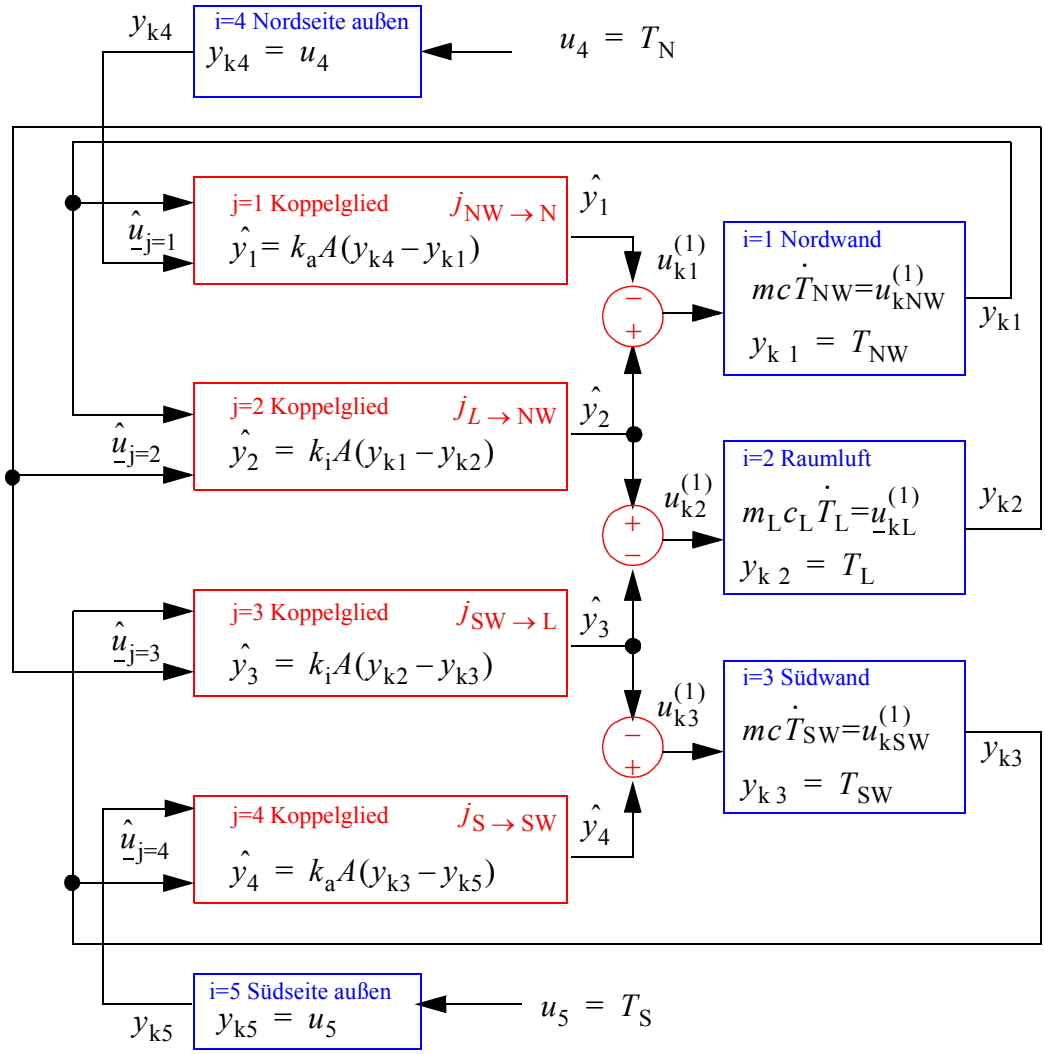


Abbildung 21: Beschreibung des Beispiels in Form der Energieflussverknüpfung

3.2.4 Phänomenologische Verknüpfung

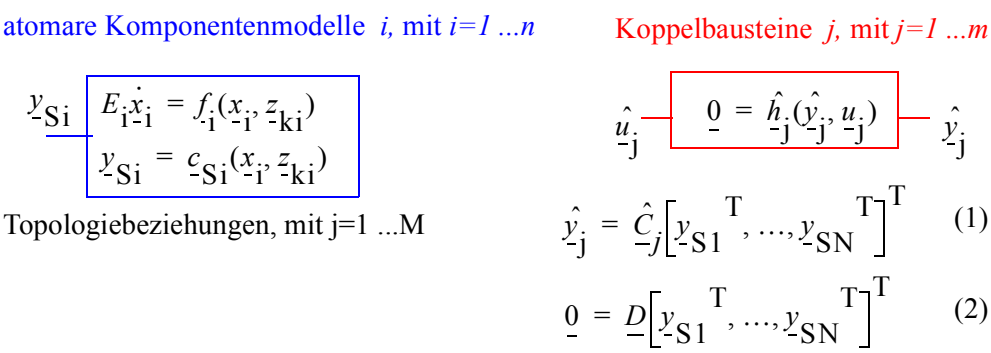


Abbildung 22: Komponentenmodell und Koppelmatrix bei der Signalflussverknüpfung. Die Verkopplungen sind ungerichtet und nicht rückwirkungsfrei (nach [Pan-99]).

Der Nachteil der bisherigen Konzepte besteht in der Notwendigkeit der festen Unterteilung der Variablen in Eingangs- und Ausgangsgrößen, was die universelle Einsetzbarkeit eines Bausteins drastisch einschränkt. Dies wird bei der phänomenologischen Verknüpfungsweise aufgegeben.

Das Komponentenmodell in Abb. 22 enthält als lokale Größe den algebraischen Koppelgrößenvektor z_{ki}. Im Komponentenmodell werden keine Eingangsgrößen definiert, sondern ausschließlich die Schnittstellengrößen y_{Si}.

Die Koppelbausteine enthalten meist ein algebraisches Gleichungssystem, das sich in linear-impliziter Form angeben lässt. Der in den Koppelbausteinen genutzte Vektor $\hat{y}_j$ fasst die Schnittstellengrößen y_{Si} zusammen, die Teil des Gleichungssystems sind.

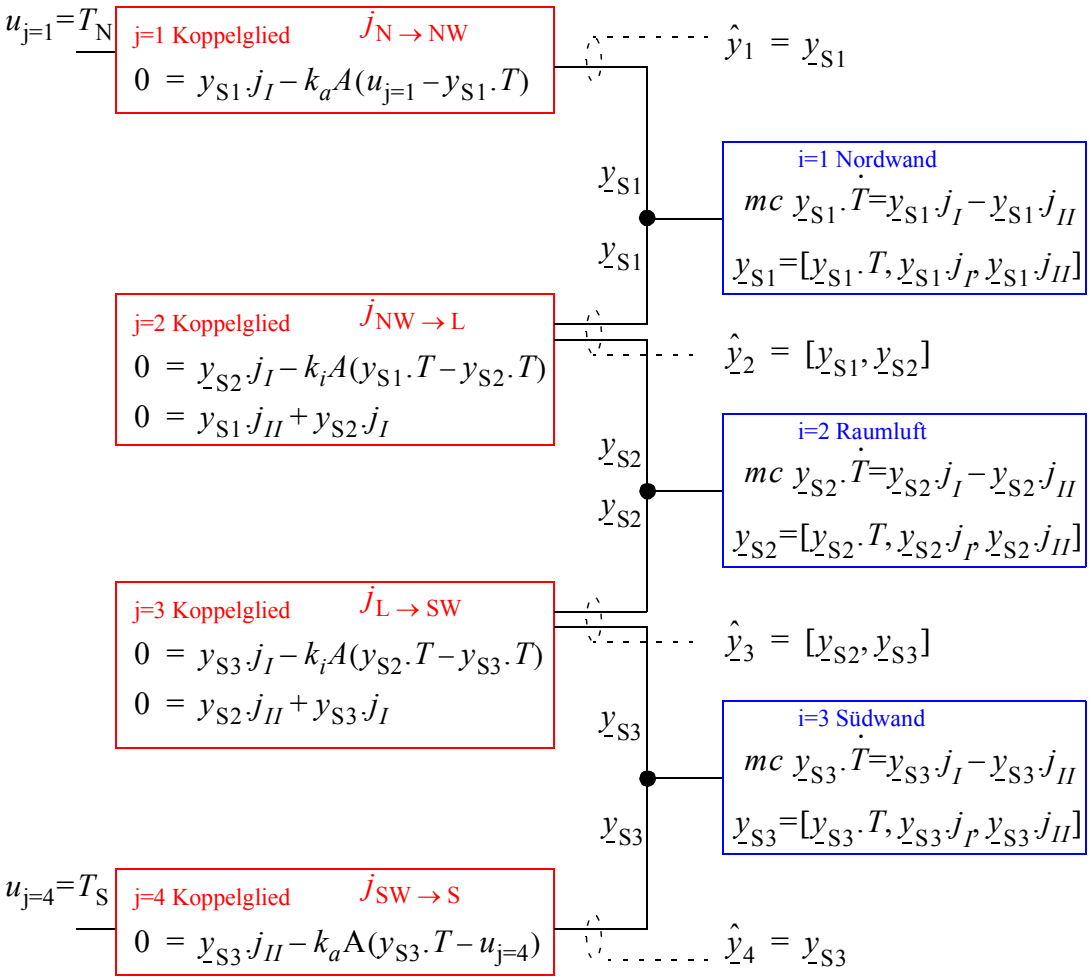


Abbildung 23: Beschreibung des Beispiels in Form der phänomenologischen Verknüpfung. (Die Notation $\underline{y}.T$ bedeutet: Komponente T des Vektors $\underline{y}$.)

Die Topologiegleichung (1) beschreibt die Verschaltungen von Komponenten mit Koppelbausteinen. Die Gleichung (2) beschreibt die direkte Komponentenverschaltung.

Dem Koppelbaustein fällt eine besondere Rolle zu. Neben der physiknahen Beschreibung der Komponentenverknüpfung bietet er die Möglichkeit, Eingangsgrößen $\underline{u}_j$ in das System von außen einzukoppeln.

Die Implementierung der Komponentenmodelle stellt eine unabhängige Verhaltensbeschreibung ohne Festlegung fester Eingangs- und Ausgangsgrößen dar.

Entgegen der zuweisungsorientierten Implementierung der beiden vorangegangenen Methoden erfordert die nicht berechnungskausale Modellierung vom Modellierungswerkzeug die Möglichkeit, symbolische Operationen durchführen zu können. Dies entlastet den Systementwickler und reduziert mögliche Fehlerquellen, da die manuellen Umformungen zur Erstellung eines Blockschaltbildes entfallen.

Die Darstellung des Beispiels in phänomenologischer Verknüpfung (Abb. 23) zeigt eine klare, intuitive Strukturierung. Die Topologiebeziehungen sind in Form bidirektionaler Verbindungslinien spezifiziert.

Die phänomenologische Systembeschreibung bildet die Grundlage nicht berechnungskausaler mathematischer Modellierungssprachen und -werkzeuge wie *Modelica/Dymola* [Mod-02, Dym-02]. Im folgenden Abschnitt 3.2.5 wird dies näher erläutert, und die notwendigen Ergänzungen zum Konzept der nicht berechnungskausalen mathematischen Modellbildung werden vorgestellt.

3.2.5 Nicht berechnungskausale Modellierung

Die nicht berechnungskausale Modellierung basiert auf der Modellerstellung durch Bilanzierung der Erhaltungsgrößen. Sie stellt eine Erweiterung der phänomenologischen Verknüpfung (Abschnitt 3.2.4) dar. Die Unterscheidung von Eingangs- und Ausgangsgrößen entfällt. Das physikalische System wird mittels eines Objektdiagrammes beschrieben. Dieses enthält Komponenten, die längs Verbindungslinien miteinander kommunizieren [Cel-91].

Innerhalb einer Komponente werden die Komponentengleichungen zunächst in Form mathematischer Gleichungen implementiert, wie sie dem Lehrbuch entnommen wurden. Die Gleichungen stellten Relationen zwischen physikalischen Größen dar, keine Funktionsausdrücke mit starr definierten Eingangs- und Ausgangsgrößen. Erst beim aktuellen Einsatz der Komponente, bei der Instantiierung, im Kontext eines Objektdiagrammes eines gesamten mathematischen Modells kann das Simulationssystem entscheiden, nach welchen Größen jede einzelne Gleichung aufzulösen ist.

Die wesentliche Ergänzung gegenüber der Methode der phänomenologischen Verknüpfung stellt die Behandlung der Schnittstellen dar. Man erkennt, dass an den Schnittstellen nur zwei grundlegend verschieden zu behandelnde Variablentypen auftreten können. Dies sind Potentialvariablen *(across variables),* welche nicht mengenartige Größen beschreiben, und Flussvariablen *(through variables),* welche mengenartige Größen beschreiben. An der Verbindungslinie müssen die Potentialvariablen gleich gesetzt werden. Die Flussvariablen hingegen addieren sich zu Null, wobei die Wahl einer identischen Flussrichtung an allen Komponentenschnittstellen vorauszusetzen ist. Man wertet daher alle in eine Komponente einfließenden Ströme positiv, alle ausfließenden Ströme negativ.

In Analogie zum elektrischen Fall entspricht dies dem Kirchoff'schen Gesetz für Spannung und Strom in einem Knoten. Es zeigt sich, dass die strikte Einhaltung dieser Konvention ausreicht, absolut modulare, universell einsetzbare Komponenten zu schaffen.

Gerade bei der Beschreibung komplexer physikalischer Systeme erlaubt diese Modellierungsmethode eine weitaus intuitivere Modellerstellung als die streng kausale blockschaltbildorientierte Modellierung. Das physikalische System besteht aus realen Komponenten. Sie stehen miteinander in Verbindung und tauschen Materie, Energie und Information aus. In gleicher Weise besteht das mathematische Modell aus modularen Komponentenmodellen, in denen unabhängig von der konkreten Fragestellung das physikalische Verhalten einer realen technischen Komponente in formaler Spezifikation hinterlegt ist. Sie sind daher weitaus häufiger wieder verwendbar als Blöcke, die für eine spezifische Fragestellung entwickelt wurden.

Das Modell ist durch Hinzunahme weiterer Komponenten erweiterbar und auch auf andere Fragestellungen anwendbar. Wiederverwendbarkeit ist somit nicht nur auf der Komponentenebene, sondern auch auf höheren Ebenen möglich und wird optimal durch die nicht berechnungskausale Modellierung unterstützt.

Die Aufgabe des objektorientierten Modellierungssystems ist es, zunächst das vorliegende Objektdiagramm zu interpretieren, die lokalen Komponentengleichungen und Verbindungsgleichungen zu sammeln und daraus ein Gesamtgleichungssystem des mathematischen

Modells zu erzeugen. Ein automatischer Transformationsalgorithmus ([CeEl-93], [Elm-93], [ElMa-97]) führt eigenständig die geeigneten symbolischen Umformungen durch.

Nach Sammeln aller Komponentengleichungen und Erzeugung der Verbindungsgleichungen ergibt sich im allgemeinen ein umfangreiches Gleichungssystem. Es hat die Form eines großen, dünn besetzten differential-algebraischen Gleichungssystems *(DAE)*. Ein effizienter Weg ist es, mittels symbolischer Transformationsalgorithmen dieses System analytisch in eine Zustandsform ($d\underline{x}/dt = f(\underline{x}, t)$) zu überführen und mittels klassischer numerischer Integrationsverfahren zu lösen.

Ausgehend von der als bekannt angenommenen Zustandsgröße ist das System nach den Unbekannten aufzulösen. Hierzu zählt insbesondere die Ableitung der Zustandsgröße. In der Regel besitzen komplexe Systeme eine höhere Anzahl von Bestimmungsgleichungen und auch mehrere Zustandsgrößen. Aufgabe des Systems ist es, auch überflüssige Variablen wie beispielsweise die Schnittstellenvariablen zu eliminieren. Nach Sortieren und rekursivem Einsetzen erhält man die numerisch zu integrierende Zustandsgleichung.

4 Charakterisierung und Auswahl einer Entwicklungsumgebung

Die Anwendung einer universellen, objektorientierten Programmiersprache zur Implementierung mathematischer Modelle *(Java, C++, ...)* lässt viele Freiheiten und hält nahezu alle Möglichkeiten offen. Allerdings sind praktisch alle Komponenten bis zum lauffähigen, implementierten Rechenmodell als Eigenlösung in Pionierarbeit zu erstellen. Typisch für die Modellbildung ist jedoch eine Vielzahl häufig wiederkehrender gleichartiger Aufgaben.

In den Ingenieurwissenschaften existiert eine Vielzahl etablierter Simulationssysteme (z.B. *Matlab/Simulink* [Mat-02], *Dymola/Modelica* [Dym-02], [Mod-02] u.v.a. (*)). Sie unterstützen methodisch neue Entwicklungsprozesse. Häufig wiederkehrende Aufgaben wie die numerische Integration, die Nullstellensuche, graphische Darstellungen oder Aufbau und Verwaltung einer Modellbibliothek werden von der Entwicklungsumgebung übernommen. Allein durch Anwendung des Simulationssystems nutzt man implizit Expertenwissen, Erfahrungen bzw. in der Domäne etablierte Methoden.

Darüber hinaus existieren anwendungsdomänenspezifische, fertig konfektionierte Simulatoren für spezielle Anwendungsbereiche. Diese lassen im Extremfall kaum eigene Entwicklungsarbeit zu, sondern das häufig verborgene mathematische Modell ist fest implementiert. Die Anpassung an die aktuelle Konfiguration kann allenfalls durch menügeführte Parametrierung erfolgen. Gerade im Bereich der Gebäudesimulation existieren eine Reihe solcher Tools für meist private Anwender ohne spezielle Vorkenntnisse. Solche Systeme werden hier nicht betrachtet.

> (*) *TRNSYS [Abschnitt 4.2] nimmt eine Sonderstellung ein. Es ist konzeptionell kein universelles mathematisches Werkzeug wie Matlab bzw. Dymola. Es erlaubt aber auch die freie Entwicklung und Einbindung eigener Module.*

4.1 Matlab/Simulink

4.1.1 Die Entwicklungsumgebung

Im Bereich der mathematischen Systemmodellierung und Simulation ist die Entwicklungsplattform *Matlab* [Mat-02, BiBr-97, Hof-98] stark verbreitet. *Matlab* stellt eine leistungsfähige mathematische Entwicklungsumgebung dar, die auf der Matrizenkalkulation basiert. Sie verfügt über eine eigene Script-Sprache und umfangreiche mathematische Bibliotheken.

Verschiedenste Forschergruppen erweiterten *Matlab* in den 20 Jahren seines Bestehen für ausgewählte Anwendungen. Diese Komponenten wurden dem *Matlab*-Programmpaket hinzugefügt und stellen die sogenannten *Matlab-toolboxes* dar. Zur Zeit existieren etwa 150 *Toolboxes* zu den verschiedensten Anwendungsgebieten wie Betriebswirtschaft, Statistik, Steuerungs- und Regelungstechnik und viele andere. Alle Berechnung werden im sogenannten *Matlab-workspace* in Form von Befehlszeilen abgearbeitet.

Eine dieser *Toolboxes* stellt die graphische Entwicklungs- und Simulationsumgebung *Simulink* dar. *Simulink* wird zur Systemmodellierung und Simulation unter *Matlab* genutzt. Die Modellspezifikation erfolgt hierbei unter einer graphischen Entwicklungsoberfläche. Über

drag-and-drop werden aus der graphischen Bibliothek Komponenten entnommen, in *Simulink* angeordnet und mit gerichteten Verbindungen verschaltet. Diese implizieren eine feste Informationsflussrichtung. Dies macht *Simulink* zu einem typischen Vertreter blockschaltbildorientierter Simulationsumgebungen. Während der Simulation muss stets *Matlab* im Hintergrund arbeiten. *Simulink* stellt nur ein graphisches Interface zum *Matlab*-Simulator dar.

Die Verfügbarkeit einer enormen Anzahl implementierter Modellbibliotheken, *Toolboxes* und Analysewerkzeuge auch zu *Simulink* folgt u.a. aufgrund der hohen Verbreitung. Die Standardbibliothek enthält mehr als 100 vordefinierte Funktionen zur Erstellung von Blockdiagrammen für lineare, nichtlineare, diskrete, zeitkontinuierliche, hybride *SISO*- und *MIMO*-Systeme *(*Abb. 24*)*. Eigene Blöcke können entweder durch Modifikation bestehender Blöcke oder durch Integration von *Matlab, C* oder *Fortran*-Code erzeugt werden. Darüberhinaus existiert ein breites Spektrum von sehr anwendungsspezifischen *Toolboxes*. Dies gilt besonders im Bereich der Regelungstechnik. So existieren neben allgemeinen *Toolboxes* zum Reglerentwurf spezielle Analysewerkzeuge für ausgewählte Themenbereiche wie robuste Regler, modellprädiktive Regler, Fuzzy Control oder auch zur Systemidentifikation.

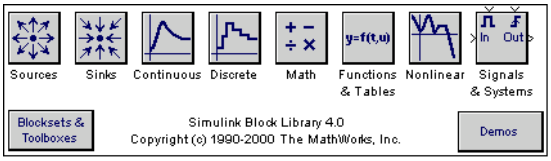


Abbildung 24: Standardbibliotheken unter *Matlab/ Simulink [Mat-02]*

Auch existiert eine Erweiterung *(Stateflow)* zur graphischen Implementierung ereignisgesteuerter Simulationen. Die Darstellung basiert auf Statecharts und Flussdiagrammen. Sie erlaubt die schnelle Implementierung diskreter Steuerungsvorgänge in zeitkontinuierlichen Systemen. Der *Realtime-Workshop* stellt einen Code Generator zur Erzeugung von effizientem *C*-Code aus *Simulink* dar.

4.1.2 Das Carnot-Gebäudemodell

4.1.2.1 Charakteristika

Zu *Simulink* existiert ein breites Spektrum anwendungsspezifischer *Toolboxes*. Viele davon gehen auf Anwender zurück, die für ihre Domäne spezifische, maßgeschneiderte Lösungen ausgearbeitet haben. So wurde vom Solar Institut Jülich für den Bereich Heizungs- und Klimatechnik das *Carnot-Blockset* ([Car-99], [Car-02], [WHS-00a], [WHS-00b]) entwickelt. Es enthält graphische *Simulink* Komponentenmodelle zur Simulation von konventionellen und solaren Heizsystemen, wobei der Schwerpunkt bei der hydraulischen Heizungsanlage nicht bei der Wärmeverteilung komplexer Gebäudearchitekturen liegt. Die Aufgabe des *Carnot*-Gebäudemodells ist es, als Verbraucher für die *Carnot*-Modelle von Heizungsanlagen zu dienen. Stoffdatenbanken existieren in beschränktem Umfang in Form einer Liste vorgeschlagener und auszuwählender Parameterwerte. Diese Listen sind allerdings im Umfang nicht vergleichbar mit ausführlichen Stoffdatenbanken spezieller Gebäudesimulatoren wie z.B. des im folgenden beschriebenen TRNSYS (Abschnitt 4.2).

4.1.2.2 Komponentenmodelle

Im Bereich Gebäudearchitektur existieren einfache vorgefertigte Zweiraumhausmodelle (Abb. 25). Komplexere Modelle lassen sich durch Verknüpfung, aber auch aus einzelnen Raumknoten erstellen ([Car-99], [Car-02]). Die Modelldarstellung ist aufgrund des Blockschaltbildcharakters wenig intuitiv und bei größeren Gebäudestrukturen sehr aufwändig.

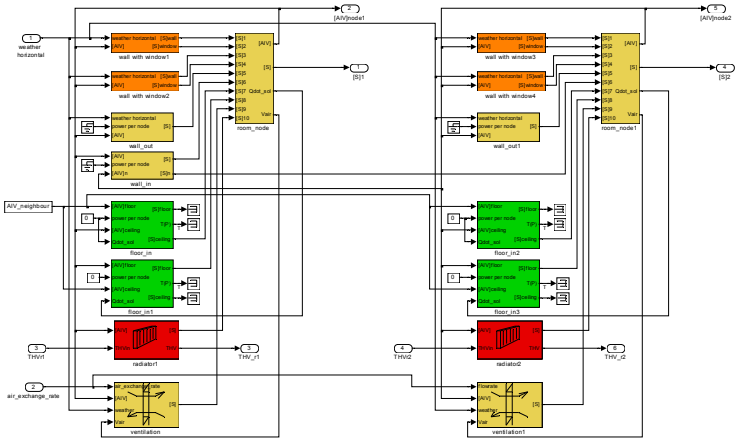


Abbildung 25: Carnot-Blockset Modell eines Zweiraumhauses [Car-02]

Die Wände werden als eindimensionale, speichernde Wärmeleiter ähnlich dem Beuken-Modell (siehe Abschnitt 9.1) beschrieben. Interner Strahlungsaustauch im Raum wird über das Zweisternmodell berücksichtigt. In Abb. 26 wird der sogenannte Raumknoten gezeigt. Hier werden neben der Bilanzierung der ein- und ausfließenden Wärmeströme auch der Luftaustausch und damit Feuchte- sowie CO_2-Gehalt der Raumluft berechnet. Man erkennt: als Eingangsgrößen dienen die Wärmeströme durch die Raumbeheizung, durch solare Einstrahlung, durch technische Geräte oder Personen sowie die Wärmeströme von den Wänden. Hinzu kommen Massenflüsse von Luft, Feuchte oder CO_2. Ausgangsgrößen des Raumknotens sind neben der Raumlufttemperatur und der Empfindungstemperatur die Temperatur des Strahlungsknotens sowie die Feuchte, der CO_2-Gehalt und abgeleitete Größen.

Die Ausgangsgrößen werden in einem Gebäudevektor zusammengefasst, um sie anderen Komponenten verfügbar zu machen. So benötigen beispielsweise die Wände Informationen über die Raumtemperaturen um die Wärmeströme berechnen zu können. Die Darstellung entspricht der Energieflussverknüpfung (siehe Abschnitt 3.2.3). Die Wände entsprechen den Koppelbausteinen, welche die Raumknoten verbinden.

Einem Zweisternmodell [Fei-94] entsprechend finden sich innerhalb des Raumknotens die in Abb. 27 detaillierter dargestellten Blöcke zur Beschreibung des konvektiven Knotens und des Strahlungsknotens. Letztendlich wird im konvektiven Knoten die Änderung der Knotentemperatur T_{Knoten} als Ergebnis der Summe aller ein- und abfließenden Wärmeströme Q_{in} nach der zeitlichen Differentialgleichung

$$\dot{T}_{\text{Knoten}} = Q_{in} / (m_{\text{Knoten}} c_{\text{Knoten}})$$ berechnet.

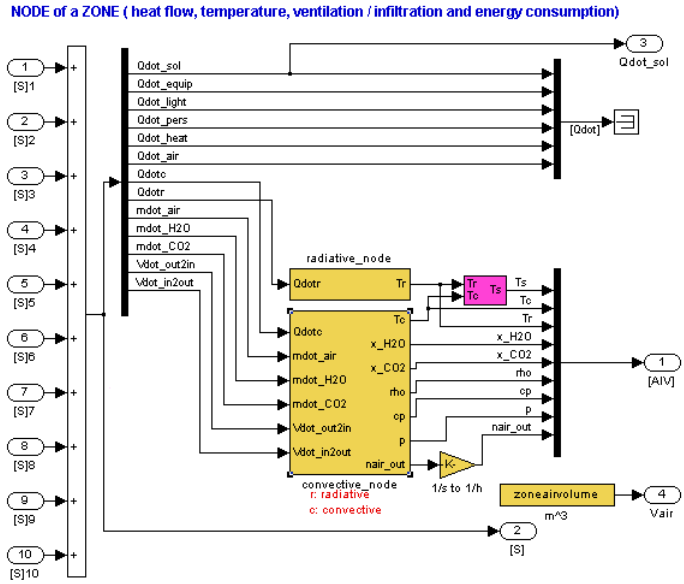


Abbildung 26: Carnot-Blockset Modell einer thermischen Zone [Car-02]

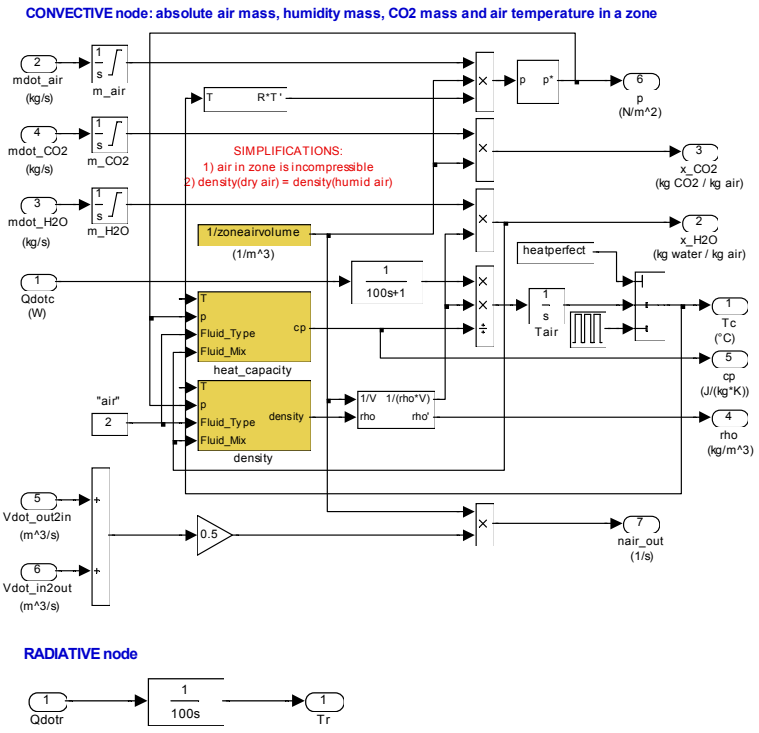


Abbildung 27: Der konvektive Knoten und der Strahlungsknoten des Raummodells [Car-02]

4.1.2.3 Hydraulische Anlage

Die hydraulische Anlage kann mit Rohrelementen, Verzweigungen, Zusammenführungen und Pumpen sehr detailliert abgebildet werden. Die Differentialgleichung der thermohydraulischen Effekte wird über eine finite Volumenmethode diskretisiert und in Knotenpunkten berechnet. Eine von den thermischen Effekten unabhängige Einstellung des Strömungsmodells ist möglich. Modellkomponenten für verschiedene Solarkollektoren existieren.

Die hydraulische Anlage nutzt auf höheren Ebenen die Hilfskonstruktion des sogenannten thermohydraulischen Vektors *(Thermo-hydraulic Vector THV)* zur Verbindung zweier hydraulischer Modellkomponenten. Der *THV* stellt das hydraulische Pendant zum Gebäudevektor dar. Der *THV* stellt Informationen über die Art des Mediums (Luft, Wasser, Wasser-Glykol-Gemisch), die Temperatur, den Massenfluss und den anliegenden Druck zur Verfügung. Die Berechnungen werden in den durch ihn verbundenen Blöcken ausgeführt. Die Modellierung einer hydraulischen Schaltung wird im folgenden Beispiel (siehe Abschnitt 4.1.2.4) veranschaulicht.

4.1.2.4 Ein Beispiel zur Verschaltung von Rohren

Das Beispiel umfasst die Verzweigung und die anschließende Zusammenführung einer Rohrleitung (Abb. 28). Das Beispiel Abb. 28 soll demonstrieren, wie stark die intuitive Vorstellung und das Blockschaltbild voneinander abweichen.

Der ursprünglich einfließende Massenstrom $\dot{m}_{Ges}$ soll sich in der Verzweigung entsprechend den Rohrwiderständen der beiden Zweige aufspalten und sich bei der Zusammenführung vereinigen. Es wird eine laminare Strömung angenommen.

Ausgehend von der Verzweigung müssen innerhalb jedes Zweiges zunächst alle in Serie geschalteten Einzelrohrwiderstände bis zum Erreichen der Zusammenführung addiert werden. Die resultierenden Gesamtwiderstände beider Zweige sind dann zum ursprünglichen Verzweigungselement zurückzuführen, so dass dieses das Verhältnis der beiden Massenströme $(\dot{m}_1, \dot{m}_2)$ der Zweige bestimmt. Da die Verzweigung auch über den ursprünglichen Massenstrom verfügt, können die Absolutwerte der Massenströme der Zweige bestimmt werden.

Das Element der Zusammenführung berechnet den am Gesamtsystem abfallenden Druckverlust vom Rohr *0* bis zur Parallelschaltung der Rohre *1* und *2*. Hierzu benötigt es neben dem effektiven Rohrwiderstand der beiden parallelen Zweige den Rohrwiderstand des vorangegangenen Elementes (Rohr *0)*. Dieser muss ihm separat zugeleitet werden.

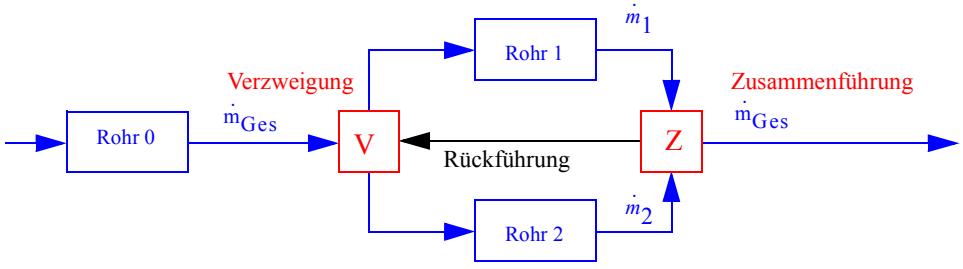


Abbildung 28: Berechnungen zur Verhaltensbeschreibung der Verzweigung einer Rohrleitung nach dem Konzept des thermohydraulischen Vektors (in Anlehnung an [Car-99])

Startet man die Berechnung des Kreises an einer Pumpe, so lassen sich durch sukzessive Abarbeitung aller Rohrstückmodelle alle relevanten Größen des hydraulischen Kreises berechnen.

Dem hydraulischen Kreis wird so entsprechend der Wasserströmung eine Informationsflussrichtung aufgeprägt, die an den Verzweigungen über Rückkopplungen verfügt.

4.1.3 Das HeatCon-Gebäudemodell

Die Entwicklungsumgebung *HeatCon* wurde am Lehrstuhl für Automatisierungstechnik der Universität Kaiserslautern entwickelt [LiKö-95]. Sie erlaubt die graphische Konfiguration eines flexiblen Simulators für fußboden- und konventionell beheizte Gebäude. Hierbei werden die geometrischen und bauphysikalischen Daten mit einer eigenentwickelten Eingabeoberfläche spezifiziert. Der Grundriss des Gebäudes kann graphisch erstellt werden. Anschließend werden die notwendigen Parameter mit Hilfe einer integrierten Datenbank [RSS-95] festgelegt (Abb. 30).

HeatCon gibt einen Simulationscode aus, der unmittelbar für die Simulationen unter *Matlab/Simulink* genutzt werden kann. Das *HeatCon*-Modell wird als *Simulink*-Block in die *Simulink*-Oberfläche eingebunden. Für jeden Raum können die Temperaturen des Estrichs, des Fußbodens, des Heizkörpers, der Wände und der Decke sowie die Raumlufttemperatur bestimmt werden. Dies sind die Ausgangsgrößen des Modells.

Das durch *HeatCon* erstellte Gebäudemodell (Simulator in Abb. 29) verfügt über eine streng kausale Aufteilung aller Schnittstellen in Ein- und Ausgangsgrößen. Es kann unter *Simulink* mit einem Kessel/Verteilermodell zur Darstellung der Heizungsanlage, einem Umgebungsmodell zur Berücksichtigung der Außenlufttemperatur und solaren Einstrahlung verschaltet werden. Die internen Schnittstellen (e) und (f) repräsentieren den gerichteten Austausch von Energieströmen. Die Eingänge (a) und Ausgänge (b), (c) und (d) erlauben die Anbindung eines Modells einer Steuerungsanlage (Abb. 29).

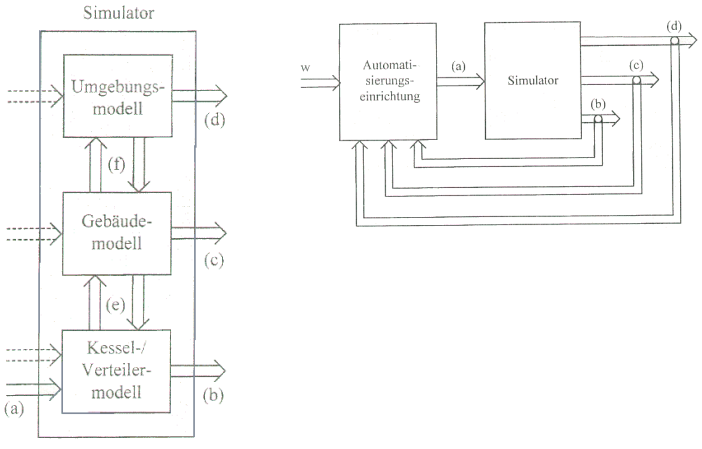


Abbildung 29: Gesamtstruktur des Simulators mit externen und internen gerichteten Schnittstellen sowie die Kopplung des Simulators mit einem Modell einer Steuerung [LiKö-95].

HeatCon konfiguriert nur das Gebäudemodell. Das Kessel/Verteilermodell, das Umgebungsmodell sowie das Steuerungsmodell sind eigenständig in *Simulink* zu implementieren.

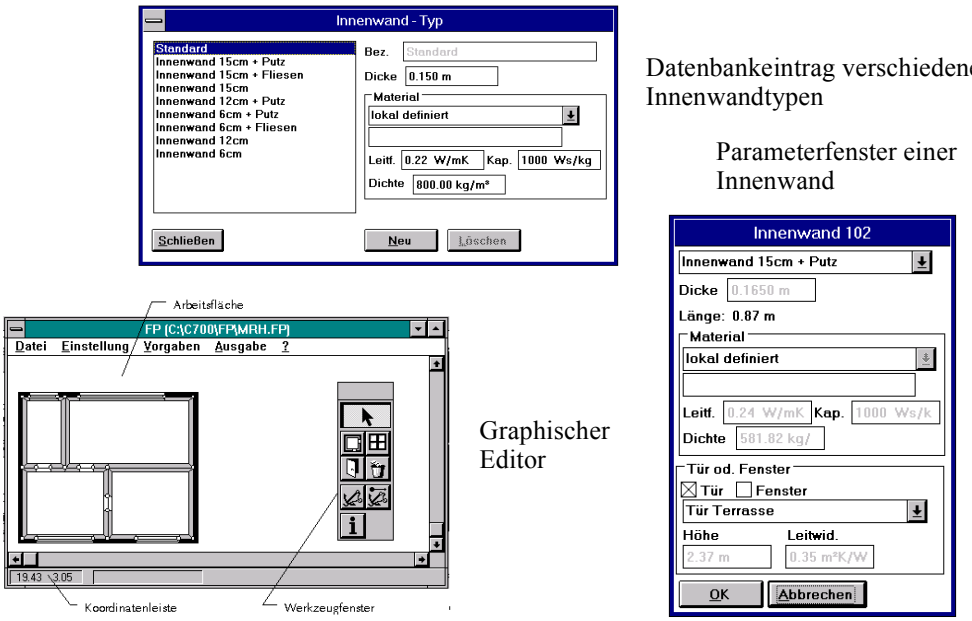
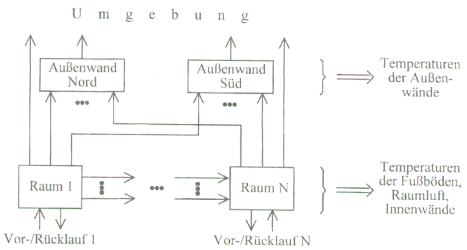


Abbildung 30: Die *HeatCon* Arbeitsfenster [LiKö-95].

Das mathematische Gebäudemodell ist signalflussorientiert strukturiert (siehe Abschnitt 3.2.2). Alle Wände werden aus zwei fiktiven Schichten aufgebaut. Eine Schicht wird dem Innenraum, die andere dem Nachbarraum bzw. der Außenwand zugeteilt. Jeder Raum verfügt so über 7 konzentrierte Speicher (*Raumluft, Heizkörper, Innenwandhälfte zu Nachbarräumen, Innenwandhälfte zur Außenwand, Decke, Fußboden* und *unterer Estrich*). Die Außenwand als äußere Hülle des Gebäudes wird unterteilt in *Nord*- und *Südseite* und ergibt zwei zusätzliche Speicher. Die Modellordnung des Gebäudes ist daher *7n+2*, wenn *n* die Zahl der Räume beschreibt (Abb. 31).



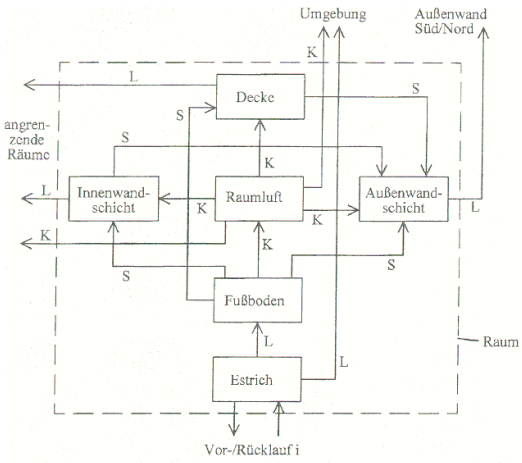


Abbildung 31: Darstellung des Gebäudemodells und der Raumzelle [LiKö-95].

Zur Gebäudebeheizung kann ein Energieeintrag in die Estrich- oder Heizkörperkomponente erfolgen. Beide enthalten zur Darstellung der Vor- und Rücklaufanschlüsse Schnittstellen zum Kessel-/Verteilermodell. An diesen Stellen fließen gerichtete Energieströme.

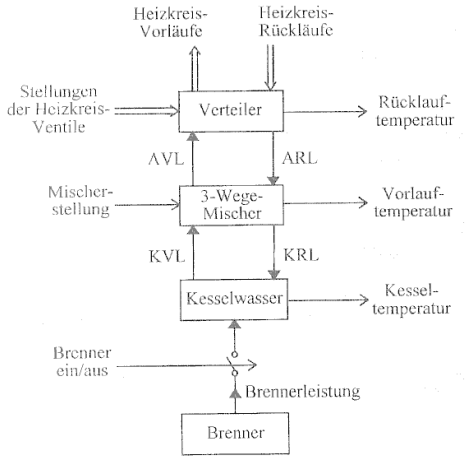


Abbildung 32: Struktur des Kessel-/Verteilermodells [LiKö-95]

Das Kessel-/Verteilermodell [Abb. 32] besteht aus den Komponenten *Brenner, Kessel, Mischer, Umwälzpumpe* und *Heizkreisverteiler*. Die Schnittstellen zum Gebäudemodell bilden die Vor- und Rückläufe der einzelnen Heizkreise. An diesen Stellen werden gerichtete Energieströme streng unterteilt nach ihrer Wirkungsrichtung übergeben. Die *Energieströme in die Heizkreisvorläufe*, die *Kesseltemperatur*, die *Vorlauftemperatur* und die *Rücklauftemperatur des Heizwassers* sind Ausgangsgrößen des Kessel-/Verteilermodells. Die Energieströme der Heizkreisrückläufe sowie die Stellgrößen des Steuerungsalgorithmus (*Stellung der Heizkreisventile, Mischerstellung, Brennerfeuerung*) sind Eingangsgrößen.

Man kann *HeatCon* als generisches Werkzeug zur Erstellung eines mathematischen Modells bezeichnen. Es erstellt anhand des graphischen Gebäudegrundrisses ein starr strukturiertes modulares mathematisches Modell der Ordnung *7n+2*. Die Nachbarschaftsverhältnisse und Verkopplungen der einzelnen Räume werden aus dem Grundriss gelesen.

Auf Raumebene sind keinerlei strukturelle Variationen des Modells mehr möglich. Die Adaption des ansonsten starren Raummodells an die baulichen Gegebenheiten erfolgt allein durch die individuelle Parametrierung. Ohne Eingriff in den *HeatCon*-Quellcode können keinerlei Änderung des mathematischen Modells erfolgen.

4.2 TRNSYS

4.2.1 Die Entwicklungsumgebung

Die Abkürzung TRNSYS steht für *transient systems simulation*. Es handelt sich um ein Programmpaket zur Simulation zeitabhängiger Prozesse in modular strukturierbaren Systemen [Trn-02]. Es wurde 1974 am Solar Energy Laboratory der Universität von Wisconsin in Madison entwickelt und seitdem ständig optimiert [Kle et al.-96]. Zunächst soll TRNSYS kurz charakterisiert werden.

Das TRNSYS Programmpaket besteht aus dem Umgebungsprogramm (TRNSHELL), aus Zusatzprogrammen zur Aufbereitung und Editierung der Eingabefiles (PREBID, PRESIM, TRNSED, IISiBat), dem eigentlichen Simulationsprogramm (TRNSYS) sowie aus Programmteilen für die Aufbereitung und Darstellung der Resultate (PLOT, SPREAD, ONLINE). IISiBat kann auch an Stelle von TRNSHELL als integrierte Simulationsumgebung verwendet werden. Das eigentliche Simulationsprogramm arbeitet im Batch-Betrieb, d.h. alle Eingaben und Ausgaben erfolgen über entsprechende Files.

TRNSYS untergliedert sich in zwei Hauptteile.

Die eigentliche Simulationsmaschine übernimmt das Einlesen der Daten, die Strukturierung des zu simulierenden Systems sowie die Steuerung und Lösung der Simulationsaufgabe. Sie ist monolithisch implementiert und bleibt dem Anwender weitgehend verborgen.

Der zweite, offene, modular gestaltete Teil besteht aus einzelnen Komponenten-Subroutinen. Dieser Teil kann vom Benutzer verändert werden, d.h. durch neue oder auch verbesserte Module ergänzt werden. So werden alle technischen Einzelkomponenten des zu simulierenden Systems durch solche sogenannte *type*-Subroutinen beschrieben. Sie verfügen an den Schnittstellen über eine Liste möglicher Ein- und Ausgabegrößen, Systemparameter und auch ihre Zustandsgrößen. Innerhalb der Subroutinen findet sich die Implementierung des mathematischen Komponentenmodells. Letztendlich reduziert sich die Modellerstellung im konkreten Anwendungsfall auf die Selektion geeigneter Komponenten, ihre Parametrierung und Verschaltung zu einem Gesamtsimulationsmodell. Es existieren unterschiedliche Versionen der einzelnen Komponenten zur Abbildung unterschiedlicher Detaillierungsgrade. In beschränktem Umfang kann die Art der Modellierung auch durch die gewählten Systemparameter eingestellt werden. Eine besondere Stellung bei der Simulation thermischen Gebäudeverhaltens nimmt der *type 56* als Repräsentant eines allgemeinen Mehrzonengebäudes ein.

Darüberhinaus existieren rein informationsverarbeitende *types* zur Darstellung oder auch Summenbildung einzelner Größen, die kein reales technisches Pedant besitzen. Viele der TRNSYS-*types* beinhalten quasistationäre Systembeschreibungen. Eine explizite Bestimmung der Zustandsgrößen erfolgt daher nicht in allen Fällen.

Die TRNSYS-*types* repräsentieren letztendlich in einer Bibliothek abgelegte Modellklassen, deren Instanzen (hier *units* genannt) innerhalb eines Objektdiagramms miteinander verschaltet werden können. Bei der Verschaltung muss explizit definiert werden, welche *In*- und *Output*-Variablen einer Komponente mit den entsprechenden *In*- und *Output*-Variablen der anderen verbunden werden. Diese Verbindung entspricht je nach Variable einem eindeutig gerichteten Massen-, Energie- oder auch Informations-Fluss, wobei auch Schleifen auftreten können. Es handelt sich somit um eine blockschaltbildorientierte Entwurfsmethode.

Es existieren sehr umfangreiche Bibliotheken einsetzbarer TRNSYS-*types***.** Man findet neben einer Vielzahl von Gebäudemodellen (*Ein- und Multizonenmodelle unterschiedlicher Detail-*

lierung) Modelle einzelner Bauteile (*Fenster, Wänden, Wintergärten, Schrägdächern und Dachgeschosse*), technische Komponenten (*geschichtete Flüssigkeitsspeicher, Geröllspeicher, algebraischer Speicher*), Regler (*Mehrpunktregler mit bzw. ohne Hysterese und sog. Mikroprozessor-Regler*), hydraulische Komponenten (*Pumpen, Rohren, Ventilen, T-Stücken zur Flussteilung oder -vereinigung*) und Sonnenkollektoren unterschiedlicher Typen.

Auch existieren *types* zur Beschreibung kompletter Subsysteme wie eines *Wasser-* oder *Luft-Kollektor-Speicher-Systems* oder einer *kompletten Brauchwarmwasseranlage*. Von Anwendern wurden *types* zur Beschreibung von *Fußbodenheizungen, Kühldecken, Elementen zur transparenten Wärmedämmung, Fensterkollektoren* oder *Latentwärmespeichern* entwickelt. Zudem existieren Komponenten zur Simulation *photovoltaischer Anwendungen* und Erweiterungen zu *beleuchtungstechnischen Fragestellungen*.

In der folgenden Abbildung (Abb. 33) soll ein Überblick des Zusammenspiels der verschiedenen TRNSYS-Programmpakete gegeben werden. Unter der Benutzerumgebung TRNSHELL können die einzelnen Teilprogramme über Menüs aufgerufen werden. Da TRNSHELL nur die textuelle Erstellung und Bearbeitung der Eingabefiles erlaubt, folgte die Entwicklung grafischer Benutzeroberflächen wie IISIBat oder PRESIM. Sie erzeugen das sogenannte *.deck-file*, das die Information über alle genutzten Komponenten sowie ihre Verschaltung enthält. TRNSED stellt eine später hinzugekommene Eingabemaske dar, die es erlaubt einfach Parametervariationen durchzuführen.

Von besonderer Bedeutung ist die Parametrierung des Mehrzonengebäudemodells (*type* 56). Um die Eingabe der notwendigen Gebäudeparameter transparenter zu gestalten, wird ein spezielles Unterprogramm (PREBID, bzw. BID) vorgeschaltet. Zunächst werden die relevanten Gebäudeparameter mittels des Eingabeeditors PREBID spezifiziert und in einem *.bui-file* hinterlegt. Alternativ kann ein graphischer Editor (SIMCAD) genutzt werden. Die Daten umfassen die Festlegung der thermischen Zonen, die Abmessungen, die Materialdaten sowie weitere gebäudetypische Daten (*Luftwechselraten, Sollwerttemperaturen, ...*). Unterstützende Stoffdatenbanken und exemplarische Wandkonfigurationen stehen in Bibliotheken zur Verfügung. Erstes Zwischenergebnis des BID Programmes sind dann eine Datei zur Gebäudebeschreibung (*.bid*), eine Datei zur Spezifikation der Transferfunktionen der Wände *(.trn)* sowie eine Datei mit zusätzlichen Informationen *(.inf)*. Hieraus werden die während der Simulation vom *type* 56 benötigten Übergabedateien (*.bld* und *.trn*) erzeugt. Wobei PREP die Bestimmung der Koeffizienten der Transferfunktionen der Wände übernimmt.

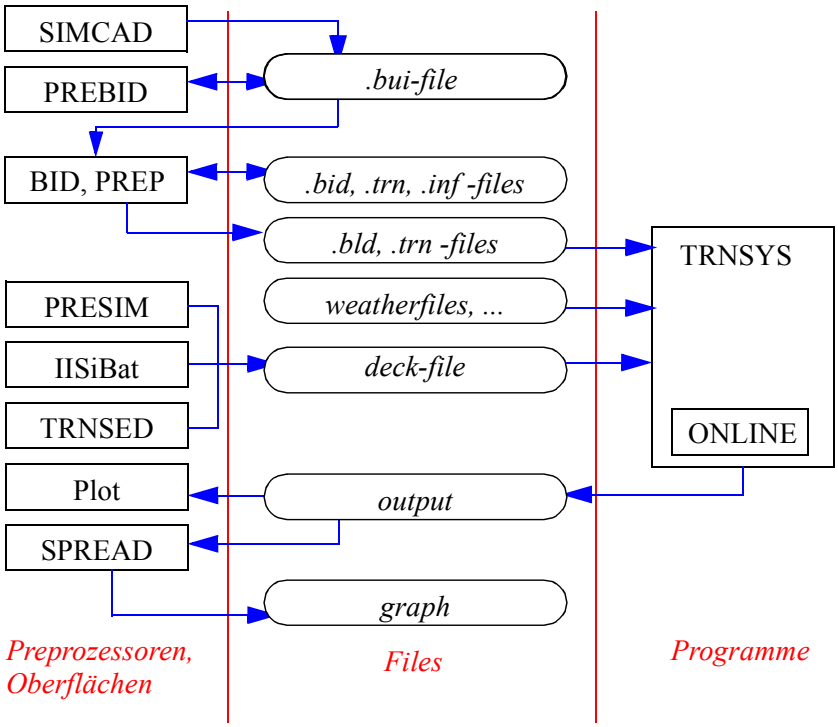


Abbildung 33: Die Module des TRNSYS-Softwarepaket [Trn-02], [Kle et al. 1996]

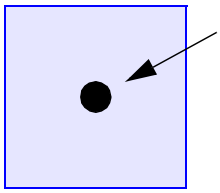
4.2.2 Das Gebäudemodell

Da TRNSYS in der vorliegenden Arbeit zur Validierung genutzt wird, wurde vorab eine exemplarische Studie zur TRNSYS Anwendung bei der Simulation des thermischen Verhaltens eines Niedrigenergiehauses durchgeführt [SpMe-01]. Im Folgenden sollen unter Anwendung der hierbei erhaltenen Erfahrungen die in TRNSYS realisierten Konzepte ausführlich vorgestellt werden.

4.2.2.1 Bilanzierung der Wärmeströme

Das Grundprinzip der TRNSYS-Gebäudemodells beruht auf der Bilanzierung der möglichen Energieströme eines Gebäudes in seinen thermischen Zonen (Abb. 34). Als thermische Zone dienen Räume bzw. Gruppen von Räumen, die gleiche Randbedingungen wie Nutzung, Verglasung oder geometrische Lage aufweisen. Der thermischen Zone wird ein Luftknoten zugewiesen mit der Wärmekapazität des Inhaltes des Zonenvolumens. Dort werden die relevanten Wärmeströme bilanziert. Sie werden in Abb. 34 einzeln aufgeführt.

In den folgenden Abbildungen werden zur Wahrung der Kompatibilität die TRNSYS-Benennungen der Variablen genutzt [Kle et al. 1996].



$$\dot{Q}_i = \dot{Q}_{\text{surf},i} + \dot{Q}_{\text{inf},i} + \dot{Q}_{\text{vent},i} + \dot{Q}_{\text{g, c, }i} + \dot{Q}_{\text{cp lg, }i}$$

$\dot{Q}_{\text{surf},i}$ *konvektive Wärmeströme von den Hüllflächen*

$\dot{Q}_{\text{inf},i}$ *Wärmeströme durch Infiltration (Undichtigkeiten)*

$\dot{Q}_{\text{vent, }i}$ *Wärmeströme über Ventilation (Lüftung)*

$\dot{Q}_{\text{g, c, }i}$ *konvektiver Anteil der internen Gewinne*

$\dot{Q}_{\text{cp lg, }i}$ *Wärmeströme durch Luftströmung aus Nachbarzonen*

Abbildung 34: Bilanzierung der Wärmeströme im Luftknoten einer thermischen Zone (nach [Kle et al. 1996])

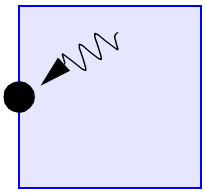
Die Wärmeströme durch Infiltration (Undichtigkeit) oder Ventilation (Lüftung) berechnen sich aus vom Anwender vorzugegebenden relativen Luftwechselzahlen pro Zeiteinheit für jede thermische Zone. Es kommt zum Austausch eines Massenstromes $\dot{m}_{\text{inf, }i}$ bzw. $\dot{m}_{\text{vent, }i}$ von Raumluft der spezifischen Wärmekapazität c_p und der Temperatur T_{i} mit Außenluft der Temperatur T_{a} im Falle der Infiltration bzw. mit Luft einer vorgegebenen Temperatur T_{v} im Falle der Ventilation. Die Wärmeströme berechnen sich entsprechend den Gleichungen (1) und (2) in Abb. 35.

$$\dot{Q}_{\text{inf},i} = \dot{m}_{\text{inf, }i} \cdot c_p \cdot (T_{\text{a}} - T_{\text{i}}) \qquad (1)$$

$$\dot{Q}_{\text{vent, }i} = \dot{m}_{\text{vent, }i} \cdot c_p \cdot (T_{\text{v}} - T_{\text{i}}) \qquad (2)$$

Abbildung 35: Wärmeströme durch Luftaustausch (nach [Kle et al. 1996])

Die Berücksichtigung der solaren Einstrahlung erfolgt durch die Bilanzierung der relevanten Wärmeströme am Oberflächenknoten der inneren Hüllflächen der thermischen Zone (Abb. 36). Die Summenbildung umfasst die Absorption der solaren Gewinne auf den inneren Hüllflächen, ihre konvektive Wärmeabgabe an den Raumluftknoten, den Strahlungsanteil möglicher interner Wärmequellen innerhalb der thermischen Zone sowie zusätzliche benutzerdefinierte Wärmegewinne innerhalb der Hüllfläche, beispielsweise durch eine Fußbodenheizung. Hinzu kommt der separat berechnete langwellige Strahlungsaustausch der Hüllflächen der thermischen Zone untereinander (Abb. 37). Die Fenster werden prinzipiell wie masselose Hüllflächen in die Bilanzierung aufgenommen. Es existieren aber auch weit detailliertere Modelle.



$$\dot{Q}_{r,\,w_i} = \dot{Q}_{g,\,r,i,w_i} + \dot{Q}_{sol,\,w_i} + \dot{Q}_{long,\,w_i} + \dot{Q}_{wall\text{-}gain}$$

$\dot{Q}_{r,\,w_i}$ *Strahlungsgewinne am Oberflächenknoten der Hüllflächen*

$\dot{Q}_{g,\,r,i,w_i}$ *Gewinne durch Strahlungsanteil der internen Gewinne*

$\dot{Q}_{sol,\,w_i}$ *solare Gewinne durch die Fenster*

$\dot{Q}_{long,\,w_i}$ *langwelliger Strahlungsaustausch der Hüllflächen*

$\dot{Q}_{wall\text{-}gain}$ *benutzerdefinierter zusätzliche Wärmegewinne*

Abbildung 36: Strahlungswärmeströme zu den Hüllflächen (nach [Kle et al. 1996])

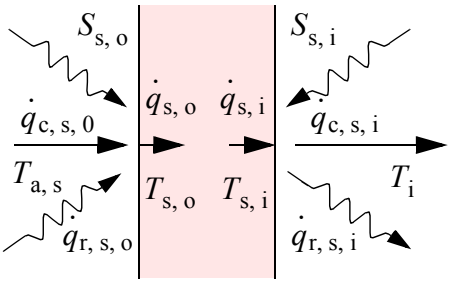
Alle Stoffwerte werden als Konstanten definiert, eine Abhängigkeit von physikalischen Zustandsgrößen wird nicht berücksichtigt.

4.2.2.2 Wandmodell

Das thermische Verhalten der Wand wird mittels der Transferfunktionsmethode [Lec-92] (Abb. 37) berechnet (siehe Anhang 9.2). Diese geht zurück auf Mitalas und Arseneault [StMi-71]. Letztendlich stellt die Transferfunktionsmethode eine Verbindung von Laplace-Transformation und z-Transformation dar. Die Laplace-Transformation wird genutzt um die partielle Differentialgleichung der Wärmeleitung zu lösen. Die z-Transformation wird genutzt um diese kontinuierliche Lösung zu diskretisieren. Die Berechnungen der Wärmeströme und Temperaturen erfolgen jeweils an der inneren und äußeren Wandoberfläche. Diese Werte stellen die Schnittstellen des Wandmodells zum restlichen Simulationsmodell dar.

Abb. 37 zeigt zunächst die Bilanzierung der Wärmestromdichten auf der Wandoberfläche. Auch dargestellt ist die Reihenentwicklung der Wärmestromdichten auf den beiden Wandoberflächen als Resultat der z-Transformation der Transferfunktionsmethode.
Die Reihenentwicklung erlaubt die Bestimmung der momentanen Wärmestromdichten aufgrund früherer Ergebnisse der Wärmestromdichten und Temperaturen zu diskreten Zeitpunkten. Die Reihe der Potenzen wird berechnet nach konstanten Zeitintervallen $t = k \cdot \Delta t$ in der Vergangenheit. Hierbei entspricht k=0 der aktuellen Zeit, k=1 ist ein Zeitschritt früher. Die Anzahl der noch zu berücksichtigenden Zeitschritte legt die Genauigkeit der Beschreibung fest. Eine Wand höherer Masse, also höherer thermischer Speicherfähigkeit, muss über eine höhere Zahl von berücksichtigten Zeitschritten verfügen als eine leichte, dünne Wand. Die Koeffizienten der Entwicklung werden vor Simulationsbeginn berechnet. Typische Zeitschritte haben die Größenordnung *1 Stunde*.

Die physikalischen Grundlagen dieser Methode werden ausführlich im Anhang 9.2 diskutiert.



$$\dot{q}_{s,i} = \sum_{k=0}^{n_{b_s}} b_s^k T_{s,o}^k - \sum_{k=0}^{n_{c_s}} c_s^k T_{s,i}^k - \sum_{k=1}^{n_{d_s}} d_s^k \dot{q}_{s,i}^k$$

$$\dot{q}_{s,o} = \sum_{k=0}^{n_{a_s}} a_s^k T_{s,o}^k - \sum_{k=0}^{n_{b_s}} b_s^k T_{s,i}^k - \sum_{k=1}^{n_{d_s}} d_s^k \dot{q}_{s,o}^k$$

$S_{s,i}$, $S_{s,o}$	*absorbierte Strahlungsgewinne*
$\dot{q}_{r,s,i}$, $\dot{q}_{r,s,o}$	*Strahlungsaustausch mit anderen Flächen*
$\dot{q}_{w,g,i}$	*benutzerdefinierter zusätzlicher Wärmefluss, z.B. d. Fußbodenheizung*
$\dot{q}_{s,i}$, $\dot{q}_{s,o}$	*Wärmefluss durch Wärmeleitung von der Wand zur Oberfläche*
$\dot{q}_{c,s,i}$, $\dot{q}_{c,s,0}$	*Wärmefluss durch Konvektion von der Wand zur Luft*
$T_{s,i}$, $T_{s,o}$	*Oberflächentemperatur*
T_i , $T_{a,s}$	*Lufttemperatur*

Index: (i) Innenseite (o) Außenseite

Abbildung 37: Wärmeflüsse und Temperaturen an einer Wand (nach [Kle et al. 1996])

4.2.2.3 Langwelliger Strahlungsaustausch

Der langwellige Strahlungsaustausch der internen Hüllflächen einer thermischen Zone wird mittels eines Einsternmodells mit zwei Temperaturknoten entsprechend Abb. 38 berechnet. Hierzu wird der fiktive Temperaturknoten mit der Temperatur T_{star} eingeführt, um die Wärmeflüsse von jeder Wand zum Raumluftknoten und zu den anderen Wänden zu bestimmen. Die zur Berechnung der Ersatzwiderstände $R_{equ,i}$ und R_{star} notwendigen Absorptionsfaktoren werden anhand von Flächenverhältnissen bestimmt. Der Zusammenhang zwischen den effektiven Wärmeströmen von den Wandoberflächenknoten zum Strahlungstemperaturknoten sowie von dort zum Raumluftknoten und der jeweils anliegenden Temperaturdifferenz wird als linear angenommen.

In gleicher Weise werden auch die Außenflächen behandelt, nur wird hier der langwellige Strahlungsaustausch mit der Umgebung durch eine Himmelstemperatur T_{sky} und einen Sichtfaktor f_{Sky} berücksichtigt.

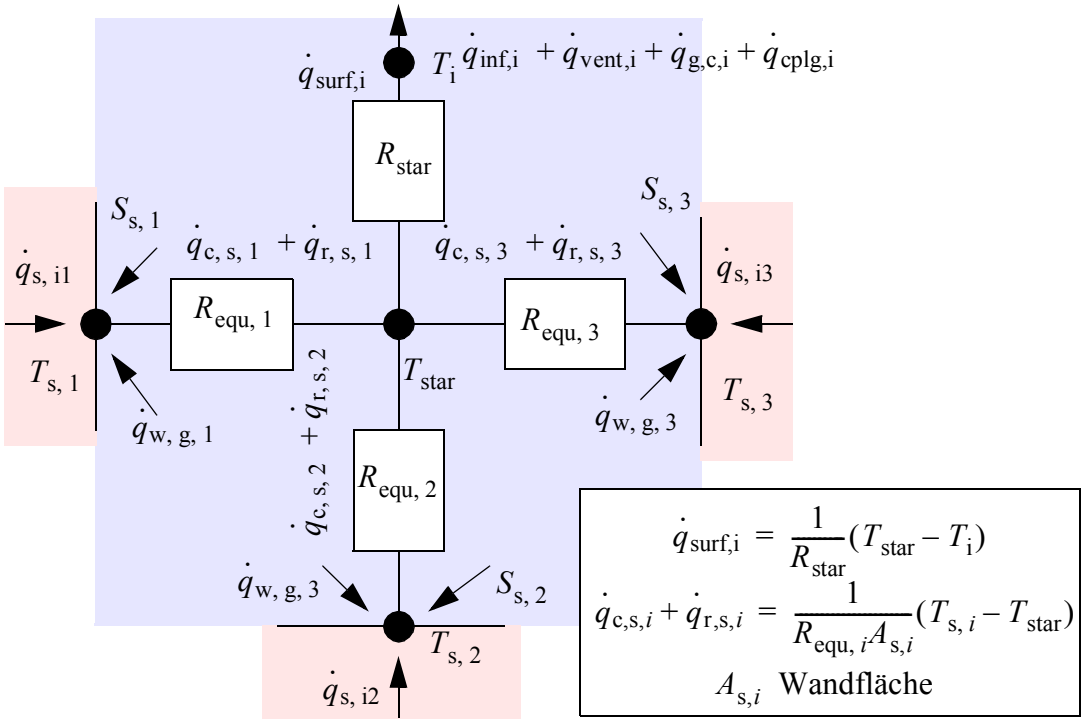


Abbildung 38: langwelliger Strahlungsaustausch der Hüllflächen einer thermischen Zone nach einem Einsternmodell (Benennungen wie in Abb. 37.) (nach [Kle et al. 1996])

Zur Berechnung der erforderlichen Energiemengen kann die Raumtemperatur jeder thermischen Zone durch eine Heizungs- und Klimaanlage mit Zweipunktreglern geregelt werden. Der erzeugte Wärmestrom wird direkt in den Raumtemperaturknoten eingespeist. Die maximal verfügbare Leistung kann limitiert werden. Es wird ein lineares Zeitverhalten des Wechsels der Raumlufttemperatur angenommen.

Das Gebäudemodell *type 56* enthält ein detailliertes Fenstermodell. Es können Fenster mit bis zu 6 Einzelscheiben und unterschiedlichen Füllgasen beschrieben werden. Jede Scheibe verfügt über einen eigenen Temperaturknoten, aber keine Speichermasse. Die innerste Scheibe ist an den Strahlungstemperaturknoten der thermischen Zone angekoppelt. Die äußerste Scheibe ist analog den Außenwänden konvektiv an die Umgebungsluft und über langwellige Wärmestrahlung an die fiktive Himmelstemperatur angeschlossen. Für jede Scheibe wird der transmittierte, absorbierte und reflektierte Anteil der diffusen und gerichteten solaren Einstrahlung berechnet. Neben der Wärmeleitung innerhalb der Scheiben werden im Scheibenzwischenraum Konvektion und langwelliger Strahlungsausgleich berücksichtigt.

Auch Verschattungselemente können beschrieben werden. Externe Verschattungselemente reduzieren im Modell die einfallende Strahlung um einen Faktor und verkleinern die Wärmeverluste durch einen zusätzlichen Widerstand. Bei internen Verschattungselementen muss zudem die Reflexion der solaren Einstrahlung in Richtung der Fensterinnenfläche sowie ihre konvektive Ankopplung an den Raumtemperaturluftknoten berücksichtigt werden.

Im Gegensatz zur langwelligen Strahlung, die alleine nach den Flächenverhältnissen im Raum verteilt wird, ist bei der solaren Einstrahlung die Absorptionsfähigkeit der Wandflä-

chen von Bedeutung. Die letztendlich in den Raum gelangte solare Einstrahlung wird daher mit dem Absorptionskoeffizienten der inneren Hüllflächen sowie der Flächengröße gewichtet und auf die Oberflächen verteilt.

Neben der Bilanzierung der Wärmeströme kann im TRNSYS-Modell *type 56* auch die relative Feuchte der Zonen bilanziert werden. Die Fähigkeit der Zonenluft Feuchte zu speichern wird in Form einer Feuchtekapazität berücksichtigt. An diese können zur Simulation des Desorptions- und Absorptionsverhaltens der Wände Oberflächen- und tiefe Feuchtespeicher angeschlossen werden. Der Feuchteaustausch zwischen zwei Zonen erfolgt alleine durch Infiltration, Ventilation und Luftströmung zwischen Nachbarzonen. Darüber hinaus können auch interne Feuchtequellen innerhalb der Zonen existieren.

4.2.3 Das Umgebungsmodell

Von großer Bedeutung für die Simulation eines Gebäudes sind die Umgebungseinflüsse. Sie werden auch in TRNSYS durch das Einbinden externer Wetterdatenfiles berücksichtigt. Dies sind im allgemeinen keine Messwerte, sondern aufbereitete Datensätze eines Jahres, die langfristige Erfahrungen nutzen. Sie sind verfügbar für viele ausgewählte Orte und in stündlichen Intervallen notiert. TRNSYS nutzt hier einen abgeänderten, vereinfachten Typ der Testreferenzjahre. In Abb. 39 ist eine typische Konfiguration des abgeänderten TRNSYS TRY Wetterdatenfiles abgebildet. Es existiert eine spezielle Subroutine *(type 9 "data reader"),* deren Aufgabe es ist, die Daten aus den externen Files einzulesen, aufzubereiten und der Simulationsumgebung verfügbar zu machen.

1	Monat
2	Tag
3	Stunde des Tages
4	direkte solare Einstrahlung (*Beam*-Strahlung) auf eine horizontale Fläche, integriert über die vorangegangene Stunde [kJ/m^2]
5	Summe aus *Beam*-Strahlung und diffuser Strahlung auf eine horizontale Fläche, integriert über die vorangegangene Stunde [kJ/m^2]
6	Umgebungstemperatur [^{o}C]
7	Windgeschwindigkeit [m/s]
8	Windrichtung [^{o}]
9	...

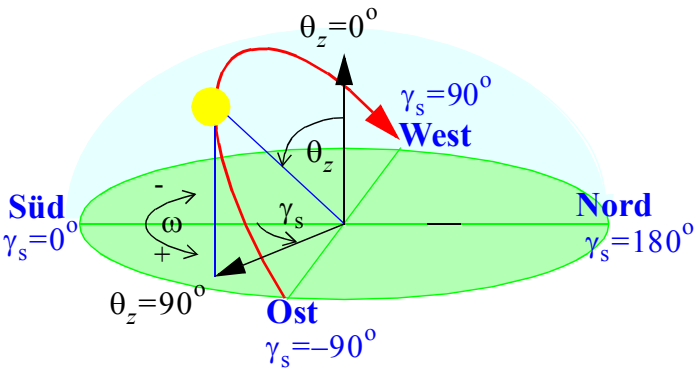
Abbildung 39: Format eines TRNSYS Wetterdatenfiles (nach [Kle et al. 1996])

Eine Sonderrolle nimmt die solare Einstrahlung ein. Für sie existiert eine spezielle Subroutine *type 16 ("solar radiation processor"*). Auch die typischen Einstrahlungsdaten aus den Wetterdatenfiles werden in stündlichen Intervallen, meist auf eine horizontale Fläche bezogen angegeben. Hier hat die Interpolation jedoch ein besonderes Gewicht.

Die Aufgabe des *type 16* ist es, diese Daten zur Ankopplung an das Gebäudemodell *type 56* geeignet aufzubereiten. So sind die stündlichen Datensätze zu interpolieren und anhand der Sonnenposition die Einstrahlung auf beliebig geneigte und orientierte Oberflächen zu berechnen. Man unterscheidet hierbei die direkte solare Einstrahlung (*Beam*-Strahlung) und die von der Atmosphäre gestreute Strahlung (diffuse Strahlung).

Gerade die Güte der Interpolation ist von überraschend großer Bedeutung für die Simulation. Beispielsweise bei Anwendung einer einfachen linearen Interpolation folgt im Bereich der Sonnenauf- und -untergänge eine resultierende solare Einstrahlung, obwohl sich die Sonne unterhalb des Horizontes befindet. Der auf die Fensternormale projizierte Anteil der *Beam*-Strahlung für Ost- bzw. Westfenster ist aber aufgrund des sehr niedrigen Sonnenstandes sehr groß. Als Artefakt werden sich zu diesen Zeiten sehr hohe solare Einträge in die entsprechenden Räume einstellen. TRNSYS vermeidet diese Problematik weitgehend, indem es den analytisch berechneten Kurvenverlauf der extraterrestrischen solaren Einstrahlung zur Interpolation der stündlichen Datensätze verwendet.

θ_z *Zenitwinkel*
γ_s *Azimutwinkel*
δ *Deklination*
ω *Stundenwinkel*
ϕ *geograph.Breitengrad*



Zenit und Azimutwinkel

$$\cos\theta_z = \sin\delta \cdot \sin\phi + \cos\delta \cdot \cos\phi \cdot \cos\omega$$

$$\sin\gamma_s = \frac{\cos\delta \sin\omega}{\sin\theta_z}$$

Abbildung 40: Berechnung der aktuellen Sonnenposition (nach [Kle et al. 1996])

Zur Berechnung der solaren Einstrahlung auf eine beliebig geneigte Oberfläche benötigt der Strahlungsprozessor als Eingangsgrößen die geeignet interpolierte diffuse Strahlung und die gerichtete *Beam*-Strahlung auf eine horizontale Fläche.

Die aktuelle Sonnenposition, spezifiziert durch den Zenit- und Azimutwinkel, folgt aus trigonometrischen Überlegungen (Abb. 40). TRNSYS zählt den Azimutwinkel γ_s vom lokalen Meridian d.h. der Südrichtung zur Projektion der Sonnensichtlinie in die Ebene. Der Stundenwinkel wird morgens *positiv*, abends *negativ* gezählt.

Basierend auf diesen Daten kann dann die Gesamteinstrahlung auf die beliebig geneigte Fläche berechnet werden. Hierbei werden die separat berechneten Anteile von *Beam*-Strahlung, Diffusstrahlung sowie der vom Erdboden reflektierten Strahlung auf die Fläche addiert. Die Orientierung der beliebig geneigten Fläche wird durch zwei Winkel beschrieben. Dies ist zum einen ihr Neigungswinkel zur Horizontalen. Zum anderen ist dies der Winkel zur Beschrei-

bung ihrer horizontalen Ausrichtung. Er wird zwischen ihrem in die Horizontale projizierten Normalenvektors und dem lokalen Meridian gemessen.

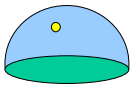
Die Berechnungen sind überwiegend durch elementar geometrische Winkelprojektionen motiviert, enthalten aber auch eine Reihe empirischer Ansätze. Dies gilt besonders für die Beschreibung des diffusen Anteils. Durch Wahl eines Systemparameters kann bei der Modellerstellung zwischen unterschiedlichen Strahlungsmodellen zu seiner Berechnung ausgewählt werden.

Während das erste Modell von einer Gleichverteilung der einfallenden diffusen Strahlung ausgeht, nimmt das zweite Modell in der Nähe der Sonnenposition eine erhöhte Intensität der diffusen Strahlung an. Der Effekt ist besonders stark bei geringer Bewölkung, weshalb die Absorption der *Beam*-Strahlung in die Berechnung aufgenommen wird. Das dritte Modell addiert zum diffusen Anteil im Bereich des Horizontes einen zusätzlichen Term hinzu, der Streueffekte berücksichtigen soll. Am aufwändigsten bezüglich der Anzahl von Parametern ist das vierte Modell. Hier werden die drei Beiträge zur diffusen Einstrahlung durch empirische Funktionen gewichtet, die von einer Reihe weiterer Parameter abhängen, die das Streuverhalten der Atmosphäre charakterisieren.

I_{bT}, I_{gT}, I_{dT}	*Beam-, Diffus, reflektierter Anteil der auf die geneigte Fläche einfallenden Strahlung*
I_b, I_d	*Beam-, Diffus- Anteil der solaren Einstrahlung auf eine horizontale Fläche*

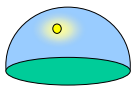
$$I_{bT} = I_b \cdot \frac{\cos\theta_z \cos\beta + \sin\theta_z \cos(\gamma_s - \gamma)\sin\beta}{\cos\theta_z} := I_b \cdot R_b$$

$$I_{dT}^{(1)} = I_d \cdot 0.5(1 + \cos\beta)$$



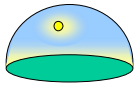
Modell 1

$$I_{dT}^{(2)} = I_d \cdot \left(0.5\left(1 - \frac{I_{bn}}{I_{on}}\right)(1 + \cos\beta) + \frac{I_{bn}}{I_{on}} R_b\right)$$



Modell 2

$$I_{dT}^{(3)} = I_d \cdot \left(0.5\left(1 - \frac{I_{bn}}{I_{on}}\right)(1 + \cos\beta) \cdot \left(1 + \sqrt{\frac{I_b}{I_b + I_d}}\left(\sin\frac{\beta}{2}\right)^3 + \frac{I_{bn}}{I_{on}} R_b\right)\right)$$



Modell 3

$$I_{dT}^{(4)} = I_d \cdot \left(0.5(1 - F_1')(1 + \cos\beta) + F_1' \cdot \frac{a}{c} + F_2' \cdot \sin\beta\right)$$

Modell 4

I_{bn}	*Beam-Strahlung bei senkrechtem Einfall*
I_{on}	*Extraterrestrische solare Einstrahlung bei senkrechtem Einfall*
$F_1', F_2', a/c$	*Empirische Funktionen bzw. Parameter zur Charakterisierung des Streuverhaltens der Atmosphäre*

$$I_{gT} = (I_b + I_d) \cdot 0.5(1 - \cos\beta)\rho_g \qquad \rho_g = \text{Reflexionsgrad des Bodens}$$

$$I_T = I_{bT} + I_{gT} + I_{dT}$$

Abbildung 41: *Beam-*, Diffus, und vom Boden reflektierte Strahlungseinträge auf eine geneigte Fläche (nach [Kle et al. 1996]).

4.3 Dymola/Modelica

4.3.1 Die Entwicklungsumgebung

4.3.1.1 Charakteristika

Die *Dymola*-Urversion wurde von H. Elmqvist schon im Jahr 1978 vorgestellt [Elm-78]. Ihre zentralen Charakteristika sind die gleichungsbasierte Implementierung der nicht berechnungskausalen Beschreibungsformen, die Definition von Modellen und Modellklassen zur wiederverwendungsorientierten Systementwicklung und vertikalen Strukturierung. Die Kausalitätszuordnung erfolgt mittels graphentheoretischer Methoden. Das Gleichungssystem wird mittels symbolischer Umformungen soweit wie möglich reduziert. Entscheidend für die Anwendbarkeit des Konzeptes waren die Arbeiten von Pantelides 1988 [Pan-88]. Er entwickelte einen Algorithmus zur Indexreduktion von DAE Systemen. Später kamen Ergänzungen zur Beschreibung hybrider Systeme hinzu.

In Abb. 42 wird ein Überblick über das *Dymola*-Programmpaket gegeben. Die Vorbereitungen zur Simulation umfassen die Erstellung des physikalischen Ersatzmodells durch den Anwender, die automatische Erzeugung des mathematischen Modells und die Code-Erzeugung. Die Modellerstellung wird durch einen graphischen Editor unterstützt. Die Zuordnung der Module des Programmpaketes zu den einzelnen Schritten wird im folgenden erläutert.

4.3.1.2 Erstellung des physikalischen Ersatzmodells

Dymola verfügt über einen graphischen Modelleditor mit dessen Hilfe das physikalische Ersatzmodell des zu beschreibenden Systems manuell vom Anwender erstellt wird. Dieses hat die Form eines Objektdiagrammes (Abb. 42). Innerhalb des Objektdiagramms befinden sich einzelne Komponenten, die letztendlich Abbilder realer technischer Komponenten des zu beschreibenden Systems darstellen. Es kann sich um einzelne Anlagenkomponenten, Aggregate, technische Maschinen oder Bauteile handeln. Diese werden, sofern sie einzelne Komponenten aggregieren, durch eigene Objektdiagramme beschrieben. Diese hierarchische Strukturierung ermöglicht eine realitätsnahe Beschreibung, die das intuitive, bausteinorientierte Verständnis einer technischen Anlage unterstützt. Am Ende der Aggregationshierarchie gelangt man zu atomaren Komponenten. Erst in diesen wird das physikalische Verhalten in Form physikalischer Gesetzmäßigkeiten spezifiziert.

Die Wechselwirkung der realen technischen Komponenten wird in den Objektdiagrammen, also auf Modellebene, durch Verbindungslinien beschrieben. Die Verbindungslinien repräsentieren echte physikalische Verbindungen wie beispielsweise Flansche, Wellen, Rohrleitungen oder Kabel. Längs dieser Verbindungslinien werden die Massen-, Energie- oder Informationsflüsse ausgetauscht. Als nicht berechnungskausale Simulationsumgebung fordert der *Dymola*-Editor an dieser Stelle keine Definition einer Flussrichtung. Allerdings kann sie im Einzelfall festgelegt werden, falls das reale System über eine offensichtliche, zwingende Flussrichtung verfügt.

Die Tätigkeit der Modellerstellung wird unterstützt durch hierarchisch strukturierte Komponentenbibliotheken. Durch Selektion einer geeigneten Komponente, ihrer Anpassung an die Charakteristika ihres aktuellen Einsatzortes - beispielsweise durch Festlegung geeigneter Systemparameter und ihrer Verschaltung im Objektdiagramm - wird die graphische Modellspezifikation erstellt. Der Anwender erstellt so das physikalische Ersatzmodell.

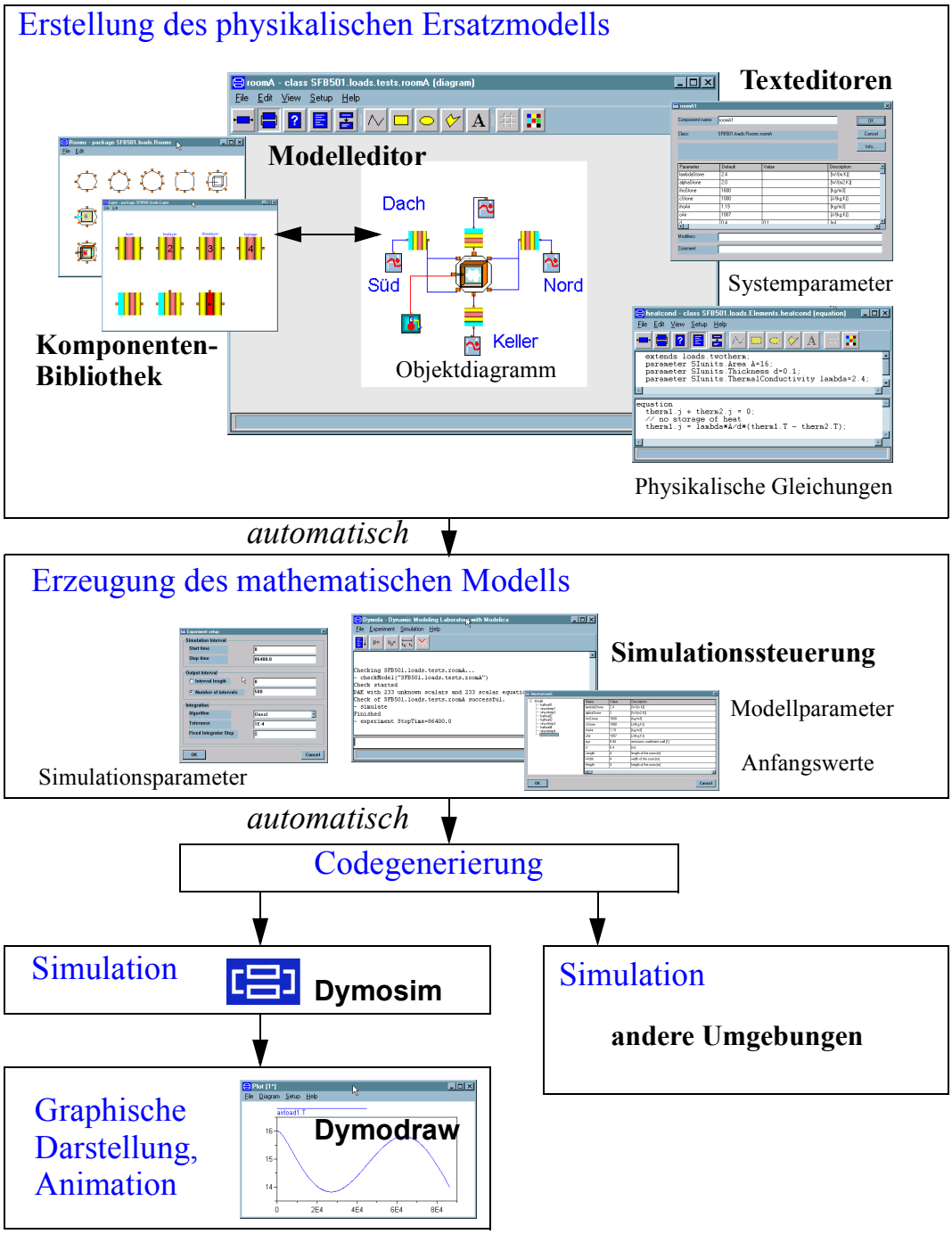


Abbildung 42: Erstellung eines mathematischen Modells unter *Dymola*

Neben dem Einsatz vorgefertigter Bibliothekskomponenten ist bei Neuentwürfen atomarer Komponenten das physikalische Verhalten zu implementieren. Hier ist neben der Unterscheidung der Schnittstellenvariablen in Potential- und Flussvariablen die gleichungsbasierte Beschreibung ihrer Zusammenhänge entsprechend allgemeingültiger physikalischer Gesetzmäßigkeiten erforderlich.

Aufgrund der Besonderheit des nicht berechnungskausalen Modellentwurfes kann sich der Anwender auf die Eingabe der physikalischen Gesetzmäßigkeiten in Lehrbuchform beschränken. Es müssen keine algebraischen Umformungen zur Auflösung einzelner Variablen getätigt werden. Eine Unterteilung in Eingangs- und Ausgangsgrößen ist nicht erforderlich. Neben der Reduktion des Aufwandes sowie möglicher manueller Fehlerquellen erhöht diese Methode die Wiederverwendbarkeit der Komponenten drastisch.

4.3.1.3 Erzeugung des mathematischen Modells

Nach dem Entwurf des physikalischen Ersatzmodells wird das mathematische Modell *automatisch* erzeugt.

Die erste Aufgabe der Simulationsumgebung ist es, alle Komponentengleichungen und Verbindungsgleichungen aufzusammeln, insbesondere beim mehrfachen Auftreten einer Komponente diese mit unterscheidbaren, individuellen Bezeichnern auszustatten und ein Gesamtgleichungssystem zu erstellen.

Tabelle 2: Benennung aller Größen des DAE-Systems

$\underline{p}$	*Parameter*
$\underline{u}(t)$	*Eingänge*
$\underline{x}(t)$	*Zustandsvariablen*
$\underline{y}(t)$	*Variablen bzw. Ausgangsvariablen*

Ausgehend von den in Tabelle 2 gezeigten Benennungen hat das resultierende differentialalgebraische Gleichungssystem (DAE-System) die Form $0 = f(\dot{\underline{x}}, \underline{x}, \underline{y}, \underline{u}, \underline{p}, t)$. Unter der Annahme, dass die Jacobi-Matrix bezüglich $\dot{\underline{x}}$ und $\underline{y}$ regulär ist, kann das Gleichungssystem durch rein algebraische Umformungen in die Zustandsform überführt werden, die numerisch integriert werden kann. Falls dies nicht möglich ist, da beispielsweise die Jacobi-Matrix nicht zu jedem Zeitpunkt t regulär ist, existieren andere Lösungsverfahren, die im Anschluss kurz erwähnt werden.

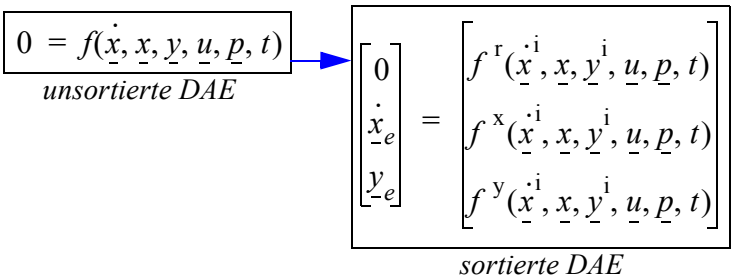


Abbildung 43: Sortieren des DAE-Systems [Ott-99d]

So ist die unsortierte DAE zu sortieren (Abb. 43). Hierzu werden die Vektoren $\dot{\underline{x}}(t)$ und $\underline{y}(t)$ in einen expliziten und impliziten Teilvektor aufgespalten. Der explizite Anteil $\underline{y}_e$ des Vektors $\underline{y}(t)$ enthält nun alle algebraischen Variablen, die explizit aus den anderen Variablen berechnet werden können. Diese können vor der numerischen Integration verborgen werden (Abb. 43). So fallen an dieser Stelle alle Schnittstellenvariablen weg, welche durch den komponentenbasierten Entwurf in die mathematische Beschreibung eingeflossen sind.

Auch können wesentlich einfacher konsistente Anfangswerte für die Zustandsvariablen bestimmt werden, da zunächst nur der implizite Anteil von $\underline{x}(t)$ in $0 = f^{\mathrm{r}}(\dot{\underline{x}}^{i}, \underline{x}, \underline{y}^{i}, \underline{u}, \underline{p}, t)$ betrachtet werden muss und die Anfangswerte des expliziten Anteils direkt berechnet werden können. Die vom sortierten DAE-System ausgehende numerische Integration kann so wesentlich effizienter erfolgen.

Block-Lower-Triangular-Transformation

Ein gängiges Verfahren zur möglichst expliziten Auflösung eines Gleichungssystems der Form $0 = f(\underline{z}, \underline{\dot{z}}, \underline{u}, \underline{p}, t)$ mit $\underline{z} = [\dot{\underline{x}}^{\mathrm{T}}, \underline{y}^{\mathrm{T}}]^{\mathrm{T}}$ ist die *Block-Lower-Triangular*-Transformation (*BLT*).

Das Verfahren beschreibt, wie durch Permutation der Gleichungen und Unbekannten eine rekursiv auflösbare Form des Gleichungssystems erstellt werden kann [Ott-99d]. Hierzu wird die Struktur des Gleichungssystems durch eine Inzidenzmatrix abgebildet (Abb. 44). In ihr wird in der k-ten Spalte und der i-ten Zeile mit 0 oder 1 markiert, ob die k-te Variable in der i-ten Gleichung vorkommt oder nicht. Durch Permutationen der Unbekannten und Gleichungen wird die Inzidenzmatrix und damit auch das Gleichungssystem in die untere Dreiecksform gebracht. Danach ist ein rekursives Auflösen möglich. Im Beispiel (Abb. 44) wurden die Variablen unterstrichen, nach denen die jeweiligen Funktionen aufzulösen sind.

Somit ist zunächst $f_2(z_1) = 0$ nach z_1 und $f_1(z_2, z_3) = 0$ nach z_3 aufzulösen, dann $f_3(z_1, z_2, z_3) = 0$ nach z_2.

Nachdem jeder Gleichung eine Variable zugeordnet werden kann, nach der diese aufzulösen ist, kann das Gleichungssystem als gerichteter Graph aufgefasst werden (Abb. 44). Die Knoten des Graphen repräsentieren die einzelnen Gleichungen. Innerhalb der Knoten bedeutet $f \rightarrow z$, dass die Funktion f nach z aufzulösen ist. Die gerichteten Kanten des Graphen bedeuten, dass die an der Pfeilspitze berechnete Variable von der Gleichung am Pfeilende benötigt wird.

Von besonderer Bedeutung für die Berechnung des Gleichungssystems ist das Auftreten von Schleifen innerhalb des gerichteten Graphen. Dies bedeutet, dass die zugehörigen Variablen voneinander abhängen und nicht rekursiv berechnet werden können. Nach ihrer Detektion werden die Schleifen zu „Superknoten“ zusammengefasst und der resultierende azyklische Graph rekursiv ausgewertet.

Automatic Tearing

Die Dimension der algebraischen Schleifen kann durch weitere Verfahren wie das *Automatic Tearing* reduziert werden. Es beruht auf geeigneten Variablensubstitutionen zur Schleifenauflösung. Ein stark vereinfachendes Beispiel wird in Abb. 44 gezeigt [ElOt-94]. Hierbei wird die Schleife des Graphen durch *tearing* (engl. *"ziehen"*) in einen linearen Graph umgewandelt.

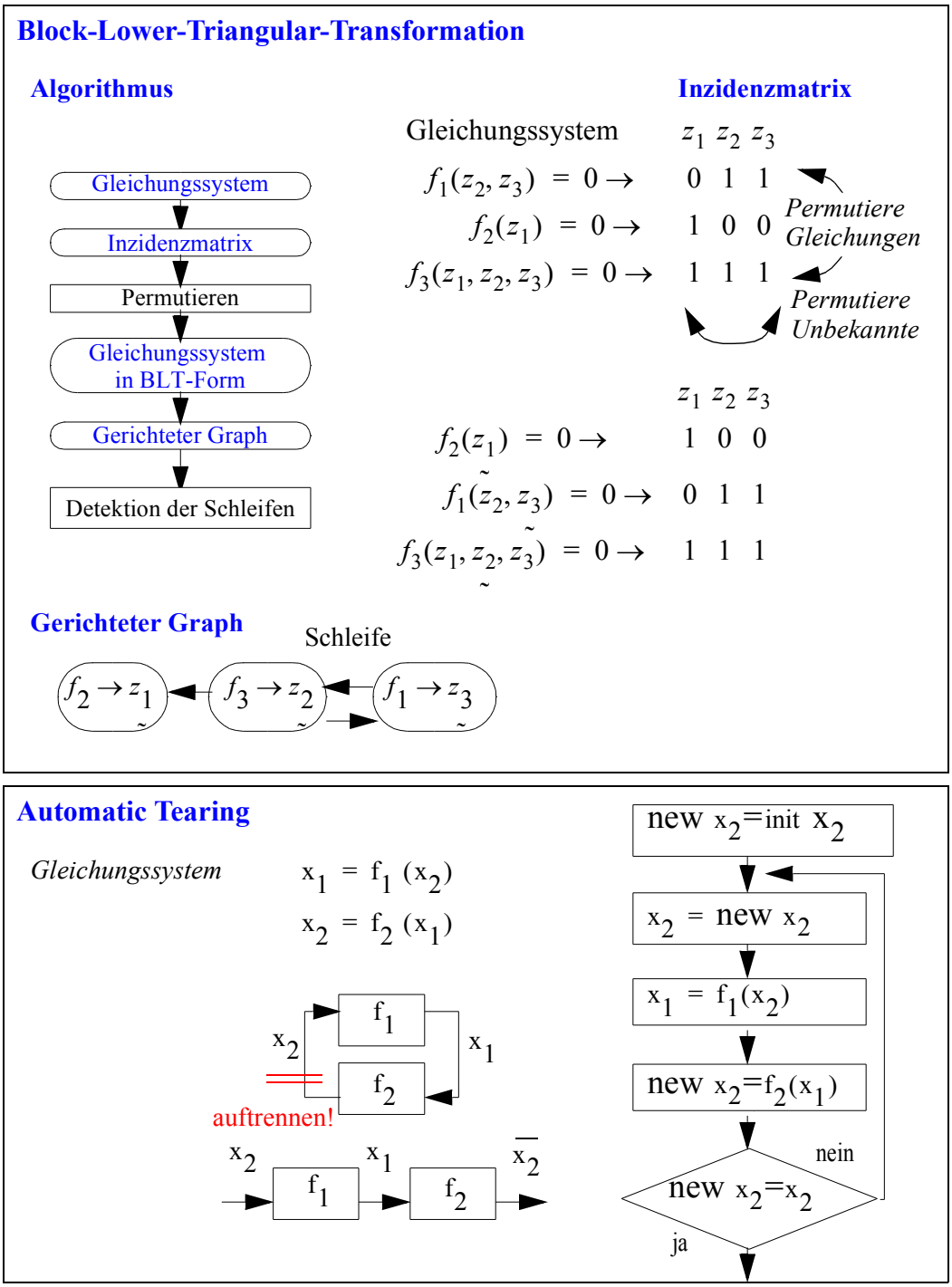


Abbildung 44: Das Block-Lower-Triangular (BLT)- und das Automatic-Tearing-Verfahren (nach [Ott-99d], [ElOt-94])

Schwieriger gestalten sich die Transformationsalgorithmen im Falle singulärer Jacobi-Matrizen. Anschaulich bedeutet dies, dass die Zustandsvariablen $\underline{x}(t)$ voneinander abhängig sind. Der Fall voneinander abhängiger Variablen $\underline{y}(t)$ braucht nicht weiter betrachtet zu werden. Entweder handelt es sich um redundante oder um widersprüchliche Gleichungen. Der erste Fall ergibt keine Probleme. Im zweiten Fall besitzt das DAE-System keine eindeutige Lösung.

Das singuläre DAE-System kann daher nicht in eine allgemeine Zustandsform umgewandelt werden, sondern allenfalls lokal entsprechend Gleichung (1) in Abb. 45. Dabei umfasst der Zustandsvektor $\underline{x}^s$ die voneinander unabhängigen Elemente, der Vektor $\underline{x}^d$ die voneinander abhängigen Anteile von $\underline{x}$. Die zeitlichen Ableitungen von $\underline{x}^d$ wurden in der Gleichung schon eliminiert. Der Vektor $\underline{x}^d$ wird in diesem Zusammenhang als *Dummy*-Zustandsgröße bezeichnet [OtBa-99a,b]. Die Wahl des Zustandsvektors $\underline{x}^s$ kann sich während der Simulation ändern und es kann notwendig sein, zwischen verschiedenen Zustandsvektoren umzuschalten.

Abbildung 45: Lokale Zustandsform im Falle eines singulären DAE-Systems [OtBa-99a]

$$\begin{bmatrix} \dot{\underline{x}}^s \\ \underline{x}^d \\ \underline{y} \end{bmatrix} = f_2(\underline{x}^s, t) \qquad (1)$$

Gerade durch die komponentenbasierte Modellbeschreibung kommt es sehr häufig zu singulären Jacobi-Matrizen. Denn die lokal definierten Komponentendifferentialgleichungen erfahren häufig durch das Verschalten der zugehörigen Komponenten Zwangsbedingungen. So kommt es dazu, dass Zustandsgrößen voneinander abhängig werden. Ein einfaches Beispiel stellt schon die direkte Parallelschaltung zweier Kapazitäten oder Wärmespeicher dar. Effektiv sollte das Simulationssystem mit nur einer Zustandsgröße weiterrechnen.

Dymola verwendet hierzu die auf dem *Pantelides-Algorithmus ([Pan-88], auch in: [OtBa-99a,b])* basierende *Dummy-Derivative-Methode*. Hierbei werden die singulären Systeme durch Differentiation von Gleichungen des DAE-Systems so aufbereitet, dass es auf die lokale Zustandsform in Abb. 45 transformiert werden kann. Der Algorithmus umfasst iterativ die im folgenden aufgelisteten Schritte 1 bis 3 [OtBa-99a,b]:

1. Schritt: Zunächst wird innerhalb des *DAE-Systems* eine möglichst kleine Untermenge von Gleichungen gesucht, bei der die Zahl von Gleichungen größer als die der Unbekannten ist. Diese Gleichungen werden als *Zwangsgleichungen* bezeichnet. Als *Unbekannte* $\underline{z}$ werden die höchsten Ableitungen der beschreibenden Variablen betrachtet. Dies sind im ersten Schritt $\dot{\underline{x}}$ und $\underline{y}$.

2. Schritt: Die so aufgefundenen *Zwangsgleichungen* werden differenziert und in das *DAE-System* aufgenommen. Durch das Differenzieren neu auftretende *Unbekannte* $\underline{z}_{Neu}$ kommen zur Menge der unbekannten Variablen hinzu. Im folgenden werden jedoch nur, wie *Schritt 1* fordert, die höchsten Ableitungen *(wie beispielsweise* $\dot{\underline{z}}_{Neu}$*)* als unbekannt angenommen. Denn $\underline{z}_{Neu}$ kann aus den *Zwangsgleichungen* berechnet werden und muss im folgenden nicht mehr betrachtet werden. Aus den *Zwangsgleichungen* müssen allerdings noch soviel *Dummy-Zustände* der auftretenden Elemente von $\underline{x}$ ausgewählt werden, bis die Gleichungen vollen Rang besitzen.

3. Schritt: Im folgenden wird das gesamte *DAE-System* - ohne die im *Schritt 2* ermittelten *Zwangsgleichungen* - bezüglich der höchsten Ableitungen analysiert. Dies bedeutet, es wird zu *Schritt 1* zurückgegangen, bis keine Untermenge von Gleichungen gefunden wird, bei der die Zahl der Gleichungen größer als die der Unbekannten ist. Der Abbruch des Algorithmus bedeutet, dass die Jacobi-Matrix des resultierenden Systems nicht mehr singulär ist.

4.3.1.4 Automatische Codegenerierung

Die automatische Codeerzeugung übernimmt das *Dymosim*-Programmpaket. *Dymosim* steht für *Dynamic model simulator*. In seinen Aufgabenbereich fällt neben der Code-Erzeugung zur kontinuierlichen Simulation die Anfangswertberechnung und Behandlung diskreter Ereignisse. Es stehen 10 verschiedene Integratoren zur Auswahl. Die ausführbaren Routinen sind im Programm *dymosim.exe* abgelegt. Zur automatischen Code-Erzeugung wird das mathematische Modell zunächst in *C*-Syntax übersetzt, compiliert und mit *dymosim.exe* gelinkt. Ein Aufruf von *dymosim.exe -i* berechnet die Anfangswerte. Der erzeugte Maschinen-Code steht dann zur Durchführung der Simulation zur Verfügung.

Die Ergebnisse der Simulation werden im *Matlab*-Format unter *dsres.mat* abgespeichert. Als Voreinstellung werden alle Variablen zu allen Zeitschritten gespeichert. Sie können dann mit dem Visualisierungsprogramm *Dymodraw* geplottet analysiert werden.

4.3.2 Die Sprache Modelica

Die Grundideen der Simulationssprache *Modelica* ([Mod-02], [Til-01]) haben ihre Wurzeln in einer ganzen Reihe von objektorientierten, nicht berechnungskausalen Simulationssprachen. Sie erlauben eine physiknahe Modellspezifikation in Form von Gleichungen, Objekten und Schnittstellen. Sie sind meist Teil einer kompletten, für sie spezifischen Simulationsumgebung, die den Anwender bei der Modellerstellung, Durchführung und Analyse unterstützen. Und sie sind im Begriff, herkömmliche, blockschaltbildorientierte Beschreibungsformen kontinuierlicher Systeme besonders bei der Beschreibung großer, komplexer Systeme zunehmend zu verdrängen [Bea-00], [BaWi-00], [CSS-00a,b], [Ott-99a,b,c,d], [OEM-99a,b], [OtBa-99a,b], [OtSc-99], [Ott-00], [Tum00a,b], [TuTi-00].

Die Tabelle 3 auf Seite 65 gibt einen Überblick über wichtigste Vorläufersprachen, die an der *Modelica*-Entwicklung Anteil hatten. Trotz der konzeptionellen Ähnlichkeit macht ihre vorwiegend syntaktische Verschiedenheit einen Austausch bereits entwickelter Modellkomponenten unmöglich. Die Idee bei der Entwicklung der Sprache *Modelica* war, dies zu ändern und einen gemeinsamen Standard zu definieren, wobei alle Konzepte der Vorgängersprachen in *Modelica* einfließen sollten. Dies bedeutet nicht, dass die Sprachen abgeschafft werden sollen, sondern sie sollen in der Lage sein ihre Modelle in der Sprache *Modelica* zu speichern und auch zu lesen. So können domänenspezifische Eigenschaften spezieller Tools auch weiterhin genutzt werden, ohne die Austauschbarkeit der Modelle zu behindern.

Tabelle 3: Vorgängersprachen von *Modelica* [Mod-02]

Sprache	Beschreibung
Dymola	*Dynamic Modeling Laboratory* Vielkörpersysteme, Antriebe, Leistungselektronik, Thermische Prozesse, *Fa. Dynasim, Lund Schweden* [Elm-78], [Dym-02]
Omola	*Objectoriented Modeling Language* Chemische Verfahrenstechnik, Kraftwerke und Versorgungstechnik *Deparment of Automatic Control, Lund Institute of Technology* [And-94, And-95]
IDA	*Indore Climate and Energy 3.0* Objektorientierter Gebäudesimulator *EQUA (früher Brisdata AB) Sundbyberg, Sweden* [Equ-02]
NMF	*Neutral Model Format* Gebäudesimulation Austauschstandard im Bereich Heizung, Klima, Lüftung, *kontrolliert von* *ASHRE (Am. Soc. for Heating Refrigerating and Air-Conditioning Engineers)*
ObjectMath	Objektorientierte *Mathematica* Erweiterung, gleichungsbasierte Modellspezifikation, Codegeneratoren für Fortran 90 bzw. C++ Code *PELAB (Programming Environment Lab), Linköping, University, Sweden* [FEV-93, ViFr-92]
SIDOPS+	Nichtlineare vieldimensionale Bond-Graphen und Blockdiagramme mit zeitkontinuierlichen und diskreten Anteilen, Mechatronic *Dep. of. Elec. Engineering, Technical University of Twente* [BrBr-97]
Smile	Scientific Simulation Environment, Energietechnik, Gebäudesimulation, *Technische Universität Berlin und Fraunhofer Institute für Computer Architektur und Software Technologie* [BiKl-95, NyBa-01]
Allan	Allgemeines Softwaretool zur Beschreibung und Simulation dynamischer Systeme *Gaz de France Research and Development Division La Plaine Saint-Denis* [JeBo-97]

Im Basissprachumfang von *Modelica* sollen häufig verwendete Bausteine enthalten sein, was die Austauschbarkeit und Verständlichkeit auch komplexer Modelle unterstützt. Diese Bausteine sind in der *Modelica Standard Library (MSL)* hinterlegt und frei erhältlich [Mod-02]. *Modelica* soll in der Elektrodynamik, Mechanik, Hydraulik, Thermodynamik, Regelungstechnik oder auch chemischen Verfahrenstechnik anwendbar sein. Auch soll es möglich sein die meisten etablierten Formalismen (Differentialgleichungssysteme, Zustandsautomaten, Petri-Netze oder Bondgraphen) dieser Anwendungsgebiete in *Modelica* zu beschreiben.

4.3.3 Die verfügbaren Modellbibliotheken

4.3.3.1 Organisation der Modelica Bibliotheken

Modelica verfügt in seinem Standardsprachumfang über Bibliotheken mit mathematischen und physikalischen Konstanten, graphischen Icons, mathematischen Standardfunktionen und vordefinierten SI-Einheiten *(Modelica Standard Library)*. Darüberhinaus existieren schon vorab vorkonfektionierte Modellbibliotheken zu verschiedenen Anwendungsbereichen. Hier werden grundlegende Bausteine mit vordefinierten und für die jeweilige Anwendung typischen Schnittstellen zur Verfügung gestellt. Sie können als Beispiel dienen, aber auch direkt eingesetzt werden. Aus ihnen konstruierte Modelle sind automatisch kompatibel zueinander. Gleiche Schnittstellen implizieren Kompatibilität. So kann beispielsweise die thermische

Schnittstelle eines Widerstandes an einen konvektiven Wärmeübergang angebunden werden. Verschiedene Schnittstellen zeigen sofort, dass hier keine Verbindung möglich ist. So kann eine elektrische Schnittstelle eines Widerstandes nur mit einer anderen elektrischen Schnittstelle verbunden werden. Die interdisziplinäre Verkopplung verschiedener Modellkomponenten erfolgt so intuitiv (Abb. 46).

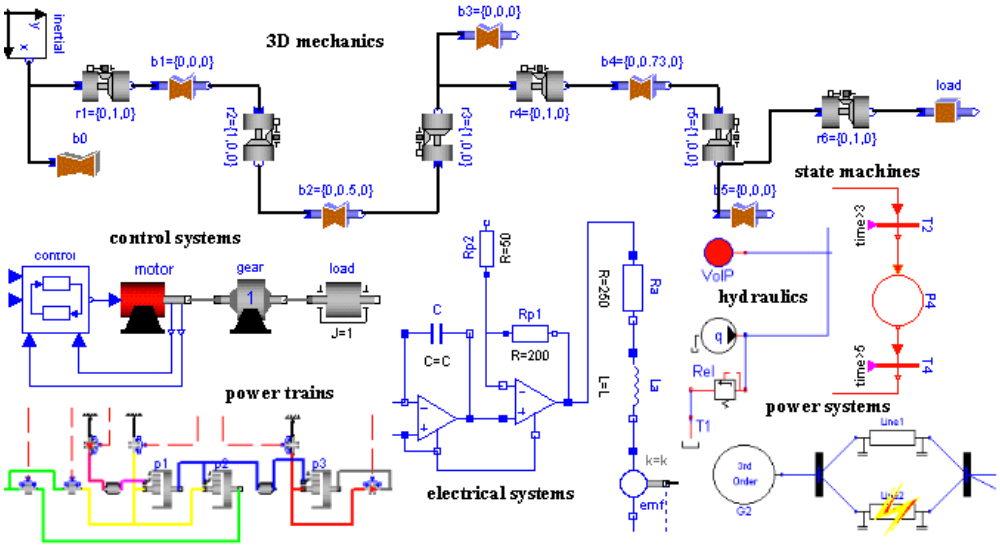


Abbildung 46: Multidisziplinäre Modellbibliotheken in *Modelica* [Mod-02]

4.3.3.2 Modelica Standard Library

Die *Modelica Standard Library* (Abb. 47) enthält blockschaltbildartige Komponenten für regelungstechnische Anwendungen. Es sind auch rein mathematische Bausteine vorhanden. Letztendlich existieren alle grundlegenden blockschaltbildartigen Standardfunktionen von *Matlab/Simulink* auch in *Modelica.*

Die Bibliothek enthält darüberhinaus nicht berechnungskausale Komponenten für elektrische Schaltkreise, mechanische translatorische und rotatorische Probleme *(*Abb. 47). Innerhalb jeder domänenspezifischen Bibliothek existiert eine Sammlung für die Anwendung geeigneter Schnittstellen *(interfaces)*.

4.3.3.3 Modelica Additions Library

Weitere Anwendungsgebiete werden durch Spezialbibliotheken verschiedener Autoren, die in der *Modelica Additions Library (MAL)* zusammengefasst sind, aufgezeigt. So existieren blockschaltbildartige Unterbibliotheken. Sie enthalten Erweiterungen für regelungstechnische Anwendungen mit Abtasthaltegliedern und Übertragungsfunktionen im z- und Laplace-Bereich. Eine ausführliche Bibliothek mit Petri-Netz-Implementierungen dient zur hybriden Modellbildung. Eine weitere Unterbibliothek befasst sich mit der Boole'schen Algebra. Sehr anwendungsspezifisch ist die Unterbibliothek zur Leistungselektronik. Einige grundlegende wärmetechnische Komponenten zur Beschreibung des eindimensionalen Wärmeflusses durch feste Körper, einfache hydraulische Komponenten sowie Bausteine zum Ein- und Auslesen externer Dateien ergänzen die Erweiterungen (Abb. 48).

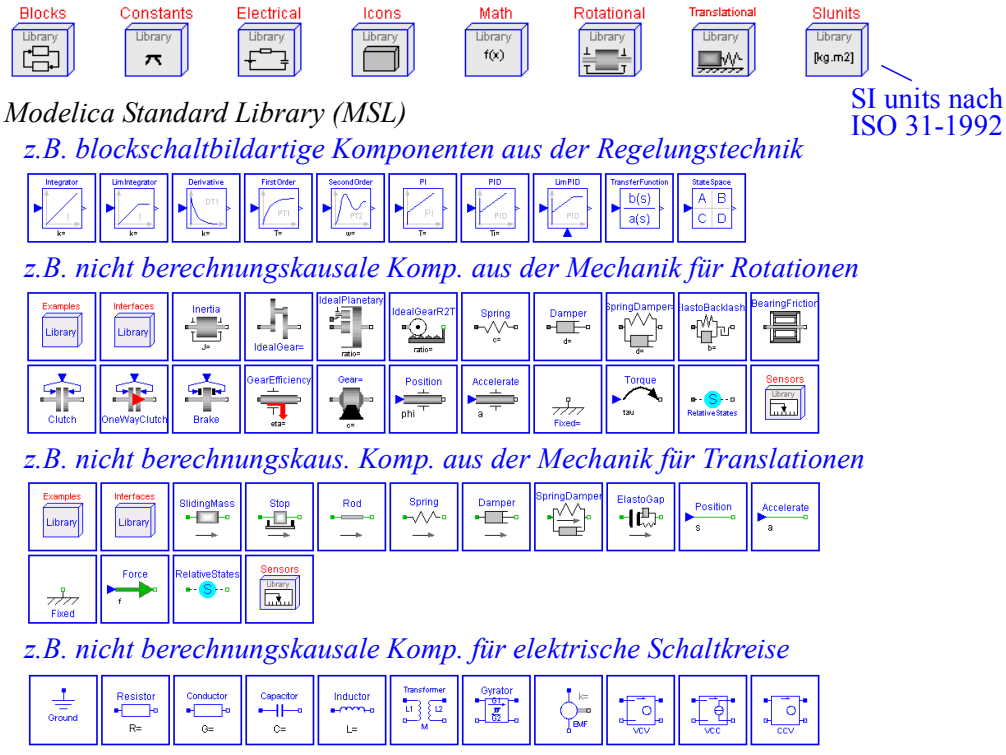


Abbildung 47: Frei erhältliche Standardbibliotheken in *Modelica* [Mod-02]

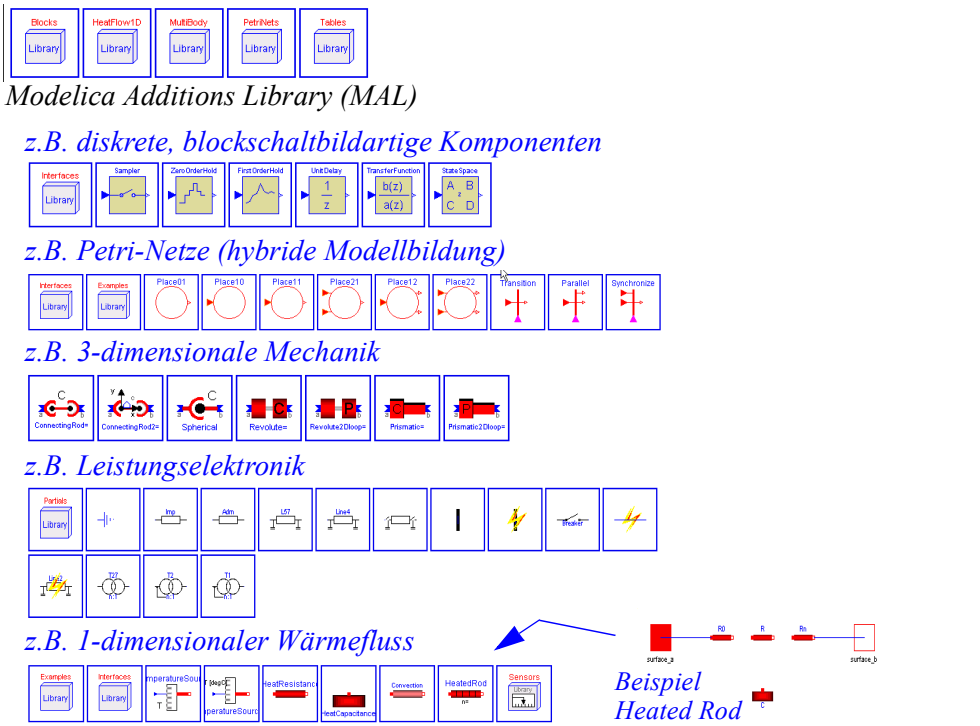


Abbildung 48: Weiterführende anwendungsspezifische Komponenten der *Modelica* Additions Library (MAL) [Mod-02]

Neben der *MSL* und *MAL*, die zunächst einen gemeinsamen Standard schaffen und die Möglichkeit der Nutzung von *Modelica* in speziellen Anwendungsgebieten aufzeigen, existieren mittlererweile auch weitere, nicht im Sprachumfang enthaltene Bibliotheken. Zum einen hat das den Grund, dass sie erst im Anwärter-Status zur Aufnahme in die *MAL* sind, zum anderen kann es sich um kommerzielle Bibliotheken handeln. Die *Modelica* Entwicklung ist im Gange. Ziel ist es, eine möglichst große Bandbreite an Spezialbibliotheken abzudecken.

4.3.3.4 Licensed Modelica Library (HyLib)

Es existieren auch erste kommerzielle Bibliotheken *(Licensed Modelica Library LML)*. Es handelt sich hierbei um die Hydraulikbibliothek *HyLib* (Abb. 49). Die Bibliothek verfügt über 90 Komponenten mit Pumpen, Rohren, Ventilen und Düsen [Bea-99, Bea-00, Mod-02]. Allerdings werden die für Heizungssysteme notwendigen thermischen Effekte nicht abgebildet. Der Schwerpunkt liegt bei der Beschreibung hydraulischer Systeme.

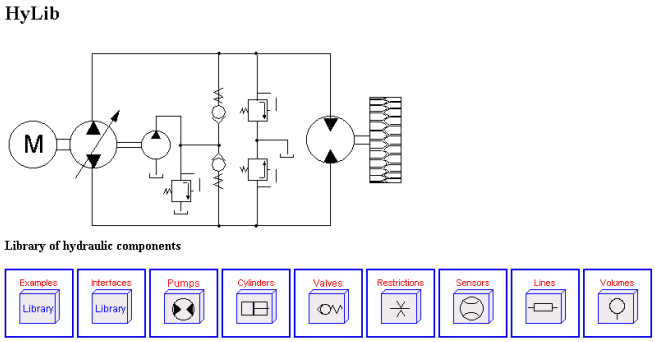


Abbildung 49: Kommerzielle Hydraulikbibliothek [Bea-99, Bea-00]

Die Entwicklungsgeschwindigkeit und Zuwachsrate an Bibliotheken und Toolboxen ist enorm. Hier ist vor allem die Zusammensetzung der *Modelica*-Association von Bedeutung. Sie umfasst die Vertreter fast aller objektorientierten Simulationssprachen, die ihre bisherige Erfahrung einbringen.

4.3.3.5 Beta-Versionen (ThermoHyd)

Frei erhältlich ist eine thermohydraulische Bibliothek [TuEb-98, Tum-00a,b,c], die sich auf mechanische sowie thermische Effekte bei Strömungsvorgängen von Flüssigkeiten (Wasser und FCKWs) und Gasen in Rohren konzentriert. Sie soll homogene eindimensionale Ein- und Zweiphasenflüsse beschreiben. Auch Wärmeabgabe- und -aufnahmeprozesse können berechnet werden. Die Bibliothek wurde zur Simulation von Motoren und Turbinen entwickelt, enthält aber auch einige für die Anwendung Gebäudesimulation sehr interessante Komponenten.

Die notwendige Modularität wird durch eine finite Volumenmethode geschaffen. Überwiegend werden konzentrierte Parametermodelle genutzt. Die Flüsse von Masse- und Energieströmen werden an den Volumengrenzen berechnet (Abb. 50). Innerhalb der Volumenelemente werden die Ströme bilanziert und die intensiven Größen berechnet. Die an den Volumengrenzen übergebenen intensiven Größen stellen Mittelwerte über den Bilanzraum dar, nicht die realen Werte an der Oberfläche. Dies ist Voraussetzung zur Beschreibung

bidirektionaler Flüsse. So können auch numerisch problematische Situationen wie Strömungsstillstand oder Rückwärtsströmung behandelt werden.

Durch Vernachlässigung der Impulsdynamik können stationäre Strömungszustände berechnet werden. Der Anwender kann bei speziellen Fragestellungen auch dynamische Impulsbilanzen in die Betrachtung aufnehmen.

Alle Komponenten erhalten ihre Eigenschaften durch dreifache Vererbung, untergliedert in einen thermischen, hydraulischen und materialspezifischen Anteil. Der thermische Anteil enthält die dynamischen Zustandsgleichungen aus der Bilanzierung der Massen- und Energieflüsse im Volumen. Die hydraulischen Eigenschaften enthalten Berechnungen der Massenflüsse aus statischer oder dynamischer Impulsbilanz. Der materialspezifische Anteil enthält alle Materialeigenschaften und spezifischen Berechnungsvorschriften.

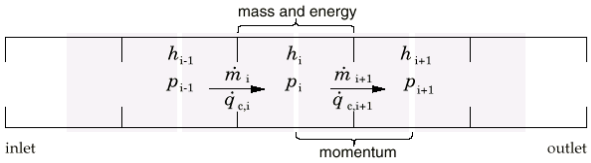


Abbildung 50: Bilanzierung von Masse- und Energieströmen im Bilanzraum [TuEb-98]

Die Bibliothek definiert eigene Schnittstellen. An den Schnittstellen der Basiselemente zur Beschreibung der Strömung eines Mediums werden als Potentialvariablen der Druck, die spezifische Enthalpie, die Dichte, die Temperatur, die spezifische Entropie und der Isentropenkoeffizient übergeben. (Der Isentropenkoeffizient beschreibt das Verhältnis der isobaren und isochoren Wärmekapazität, und ist wichtig bei isentropen Zustandsänderungen. Diese laufen bei konstanter Entropie also in adiabaten Systemen ab.) Als Flussvariablen werden der Massenstrom und der konvektive Wärmestrom übergeben. Im Falle multimedialer Ströme ist die Variablenliste zu erweitern.

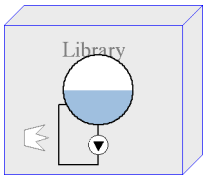
Die Bibliothek verfügt mittlererweile auch über Pumpen, Turbinen, Turbomaschinen, Rohrelemente, Ventile, Wärmetauscher und auch Heizungsboiler (Abb. 51). Zur Zeit ist nur eine experimentelle β-Version verfügbar.

Eine spätere Verknüpfung der Hydraulikkomponenten von Tummescheit ([TuEb-98], [Tum-00a,b,c]) mit den Gebäudemodellen der vorliegenden Arbeit ist über Adapterelemente denkbar aber problematisch.

Die Modellkomponenten von Tummescheit sind sehr flexibel und vielseitig. Sie können Gemische verschiedener Medien, Phasenwechsel, chemische Reaktionen der Medien untereinander und auch kompressible Medien beschreiben.

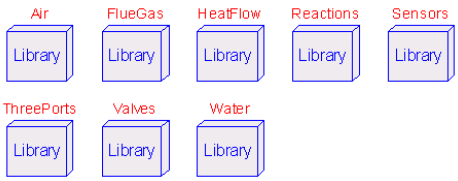
Diese Vielseitigkeit erfordert komplexe, aufwändige Modellbausteine, die auch mehr Simulationszeit benötigen als einfachere Bausteine. Sie erfordern zudem aufwändige Parametrierungen bei der Modellerstellung.

Die Vielseitigkeit der Modellkomponenten ist für die Berechnung der Wärmeverteilung in einer Heizungsanlage nicht erforderlich. Kombinierte Simulationsmodelle werden daher eine unnötige Komplexität aufweisen. Dies erklärt die Notwendigkeit der Entwicklung einer eigenen, anwendungsspezifischen Thermohydraulik-Bibliothek zur Beschreibung konventioneller Warmwasserheizungen von Gebäuden (siehe Kapitel 5.2).

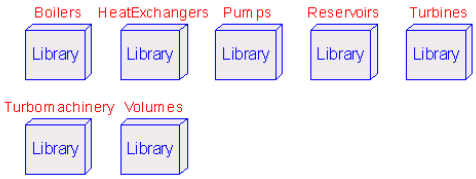


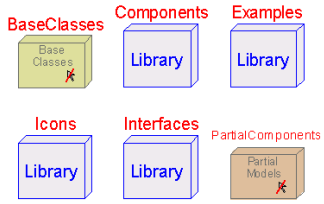
ThermoHyd -Bibliothek

z.B. *Components*-Bibliothek

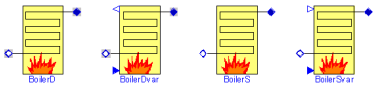


z.B. *Water*-Bibliothek

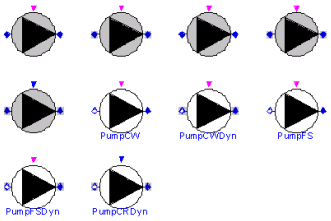




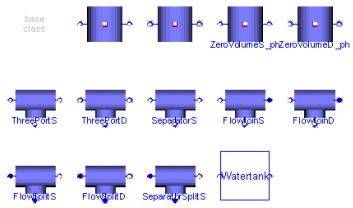
z.B.*Boiler*-Bibliothek



z.B.*Pumpen*-Bibliothek



z.B.*Volume*-Bibliothek



z.B.*Heat-Exchanger*-Bibliothek

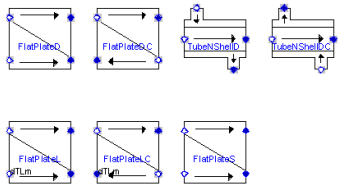


Abbildung 51: Thermohydraulische Bibliothek [TuEb-98, Tum-00a,b,c]

4.4 Andere Umgebungen

Die Auflistung erhebt keinerlei Anspruch auf Vollständigkeit. Wie die Zusammenstellung der *Modelica*-Vorläufersprachen (Tabelle 3 auf Seite 65) schon andeutet, existiert eine Vielzahl innovativer, sehr interessanter weiterer Konzepte, die sich auch mit der Simulation thermischen Gebäudeverhaltens befassen (ColSim [WHS-01], IDA [Equ-02], Smile [NyBa-01], PSiGene [Zim-01], u.v.a.). Die vorliegende Darstellung beschränkt sich auf die für das Projekt zum konzeptionellen oder simulativen Vergleich genutzten Umgebungen.

5 Erstellung der Modellbibliothek

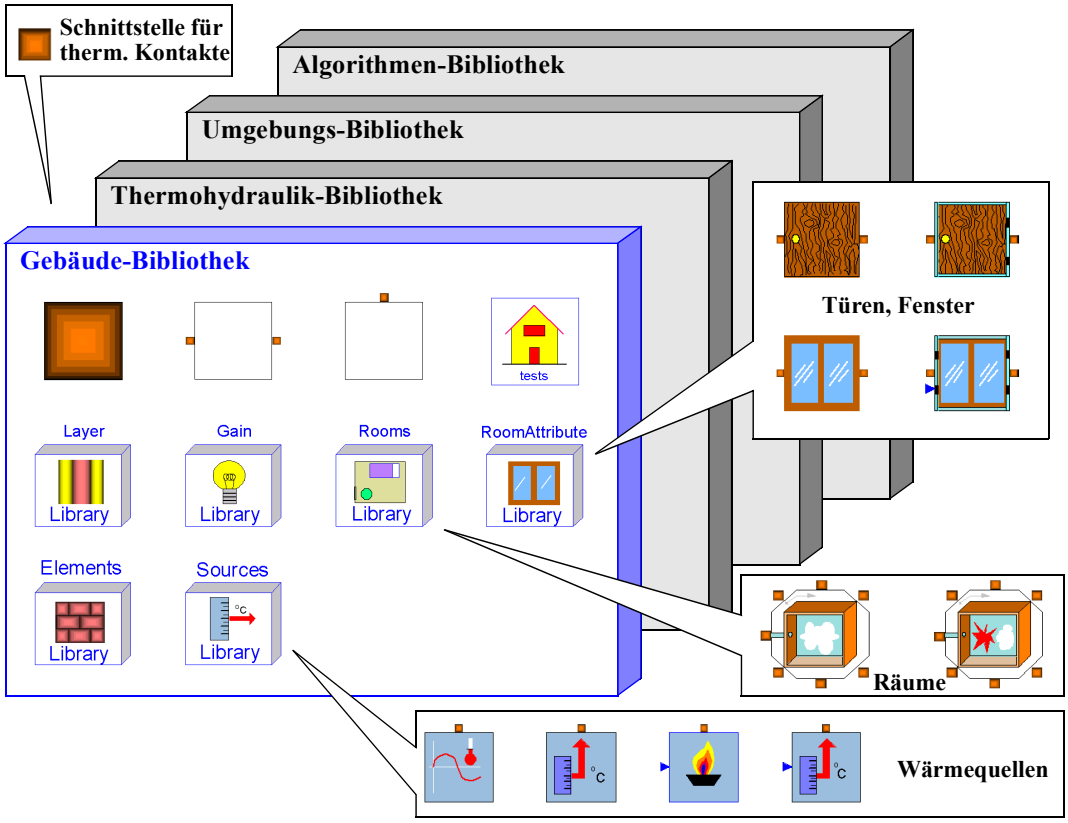


Abbildung 52: Modellbibliothek zur Gebäudesimulation
[See-00], [Sit-00],
[FeMe-01], [MeLi-01], [SiMe-01],
[Agu-02], [Cla-02], [Fel-02]

Es wurde eine Modellbibliothek (Abb. 52) zur Simulation des thermischen Gebäudeverhaltens in *Modelica* entwickelt ([FeMe-01], [MeLi-01], [SiMe-01], [Fel et al. 02]).

Die Bibliothek wurde unter Beteiligung studentischer Hilfskräfte, Diplom- und Masterkandidaten ([See-00], [Sit-00], [Agu-02], [Cla-02], [Fel-02]) erstellt (Kap. 5) und validiert (Kap. 6). Erfahrungen des Lehrstuhls zur mathematischen Beschreibung konventionell beheizter Gebäude wurden genutzt [LiKö-95].

Die Bibliothek untergliedert sich in die Abschnitte Gebäude-Bibliothek (Abschnitt 5.1), Thermohydraulik-Bibliothek (Abschnitt 5.2), Umgebungs-Bibliothek (Abschnitt 5.3) und Algorithmen-Bibliothek (Abschnitt 5.4). Dieses Spektrum deckt typische Fragestellungen der thermischen Gebäudesimulation ab.

Der anwendungsdomänenübergreifende, multidisziplinäre Charakter von *Modelica* eröffnet darüberhinaus weitere Perspektiven. So können bei speziellen Fragestellungen sofort andere Bausteine aus dem Bereich der Regelungstechnik, elektrische, mechanische oder pneumatische Komponentenmodelle auch von anderen *Modelica* Anwendern zur Beschreibung spezieller Aspekte angekoppelt werden.

5.1 Die Gebäude-Bibliothek

5.1.1 Die Begriffsbildung

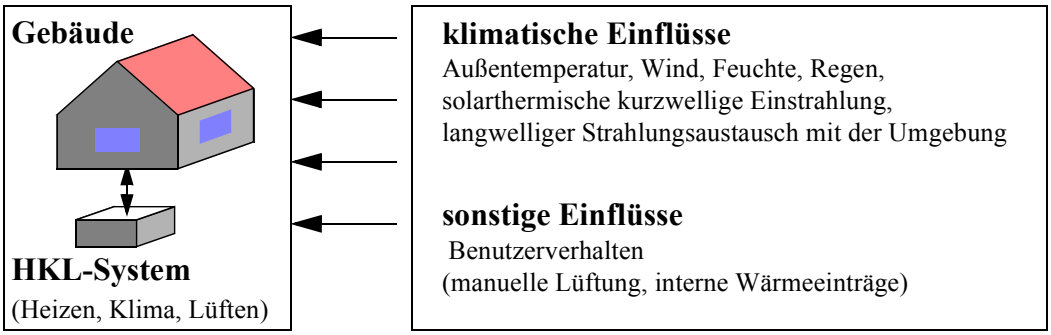


Abbildung 53: Die Wechselwirkung des Gebäudes mit äußeren Einflüssen

Primäre Aufgabe bei der Erstellung eines Gebäudemodells ist es, die Geometrie und physikalische Beschaffenheit eines Gebäudes in ein mathematisches Modell zu übertragen. Die hierbei angewendete Grundidee ist es, die einzelnen Räume, aber auch die sie verbindenden Wände in ein räumliches Gitter zu übertragen, wobei den einzelnen Gitterpunkten physikalische Eigenschaften wie Temperatur, Wärmekapazität oder auch Wärmeleitfähigkeit zugeordnet werden. Das thermische Verhalten des Gebäudes, die zeitliche Veränderung seiner Energieströme kann so analog einem elektrischen Netzwerk aus Widerständen und Kapazitäten dargestellt werden.

Die Komponentenbausteine werden in einer Bibliothek hinterlegt. Sie enthalten die physikalische Verhaltensbeschreibung des realen Bauteils. Die nicht berechnungskausale Modellerstellung erlaubt eine so physiknahe Implementierung, dass im folgenden die relevanten physikalischen Effekte allein durch die Vorstellung und Diskussion der Modellklassen erläutert werden können. Bei der Auswahl der zu berücksichtigenden Effekte wurde auf die VDI Richtlinie [VDI-6020] zurückgegriffen.

Alle Modellkomponenten sind objektorientiert modelliert. Vererbungshierarchien definieren gemeinsame Grundstrukturen mit standardisierten Schnittstellen. Sie stellen gemeinsam benötigte Materialeigenschaften zur Verfügung.

Bei der Modellerstellung durch den Anwender sind Bausteinklassen aus der Bibliothek zu selektieren, zu instantiieren und individuell zu parametrieren. Gegebenenfalls kann auch eine Adaption erforderlich sein. Die Komponenten werden in einem Objektdiagramm zu einem Gebäudemodell verschaltet. Neben den elementaren Bausteinen existieren in der Bibliothek komplexe Bausteine, die komplette architektonische Strukturen repräsentieren. Ihre vertikale Aggregationshierarchie folgt der Strukturierung eines realen Gebäudes.

Das System unterliegt äußeren Störungen u.a. in Form der äußeren klimatischen Einflüsse. Hierzu zählt beispielsweise die solarthermische Einstrahlung, der Strahlungsaustausch der Hüllflächen mit der Umgebung, die Außenlufttemperatur und alle die Wärmeübergangszahl der äußeren Gebäudehülle beeinflussenden Größen. Dies ist der Wind, aber auch die herrschende Feuchte oder der Regen. Darüber hinaus beeinflusst der Wind maßgeblich den Luftaustausch des Gebäudes mit seiner Umgebung durch Fugen und Ritzen. Weitere Störungen sind das Benutzerverhalten bezüglich Lüftung oder zusätzliche interne Wärmeeinträge. Die meist in Form von Datenfiles vorhandenen klimatischen Einflüsse und eine Spezifikation des Benutzerverhaltens werden während der Simulationszeit gelesen und eingekoppelt.

5.1.2 Ein Beispiel

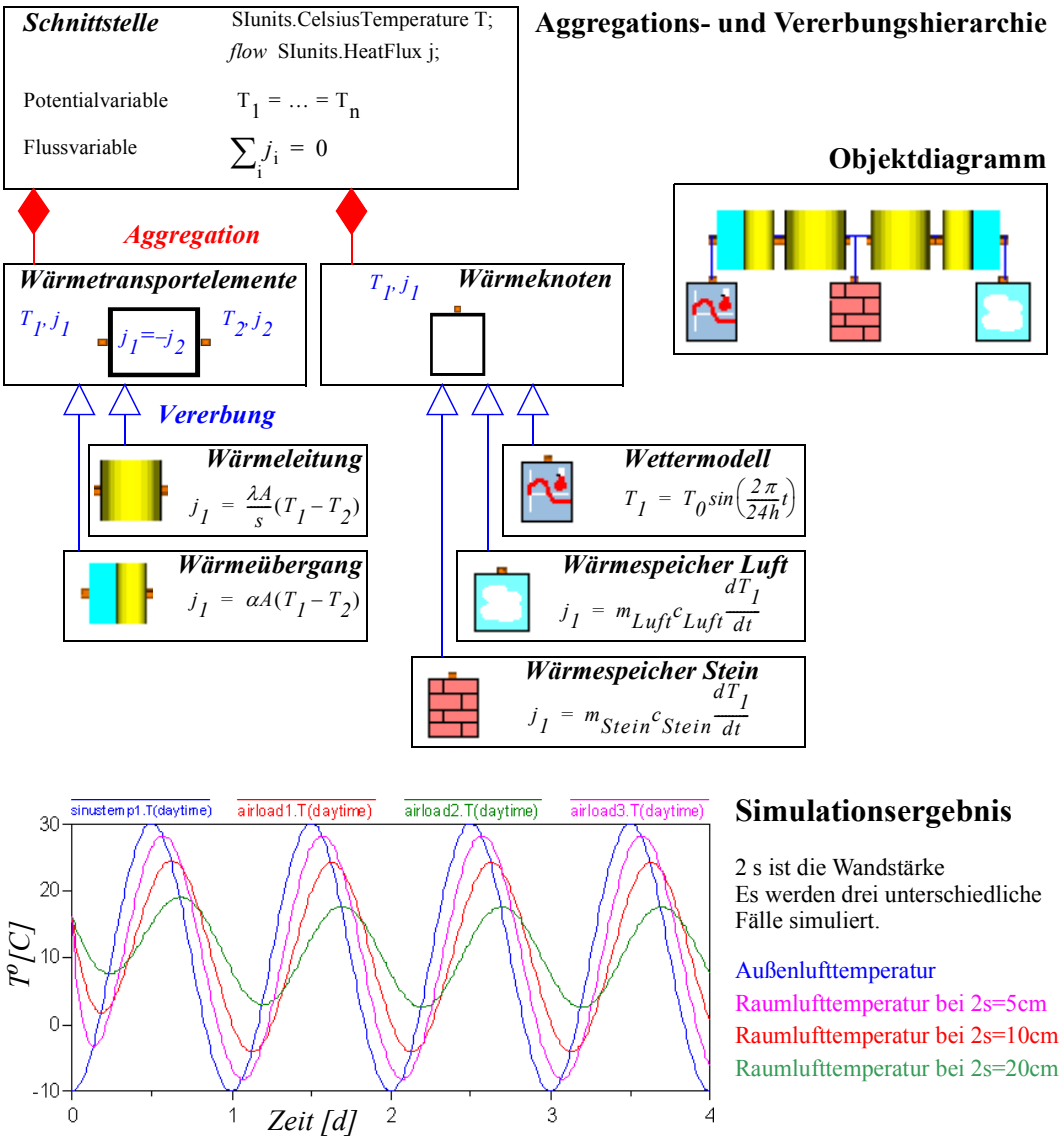


Abbildung 54: Definition der Komponentenbausteine und des Objektdiagramm eines Einraumhauses

Abb. 54 zeigt die Implementierung eines Einraumhauses in *Modelica*. Es soll die Raumlufttemperatur bei unterschiedlichen Wandstärken in Folge einer sinusförmig variierenden Außenlufttemperatur berechnet werden. Das Modell besteht aus zwei thermischen Speichern, den Wänden und der Raumluft. Abb. 54 enthält das Objektdiagramm, die Aggregations- und Vererbungshierarchie zur Definition der Komponentenbausteine und das Simulationsergebnis.

Ausgehend von einer einheitlichen Schnittstellendefinition für thermische Kontakte werden die unvollständigen Basisklassen *Wärmetransportelement* und *Wärmeknoten* definiert. Hierbei dient die Temperatur T an der Kontaktfläche als Potentialvariable $(T_1 = ... = T_n)$, der Wärmestrom j als Flussvariable $(\sum_i j_i = 0)$. Die Wärmeströme werden so automatisch an den Schnittstellen bilanziert.

Während die *Wärmetransportelemente* über zwei Schnittstellen verfügen, besitzen die *Wärmespeicher* nur eine Schnittstelle. Die Wärmetransportelemente werden als nicht speichernd angenommen; daher fordert man $j_1 = -j_2$. Hierbei bezeichnen j_1 bzw. j_2 die Flussvariable des Wärmestroms der linken bzw. rechten Schnittstelle. (Alle Verknüpfungspunkte (orange) sind Schnittstellen.)

Die Basisklassen vererben ihre Eigenschaften an die Modellklassen *Wärmeleitung*, *Wärmeübergang*, *Wärmespeicher Stein*, *Wärmespeicher Luft* und *Wettermodell*. Dort findet sich das physikalische Verhalten nahezu in Lehrbuchform. Parametrierte Instanzen dieser Klassen sind im Objektdiagramm verschaltet.

5.1.3 Das Wandmodell

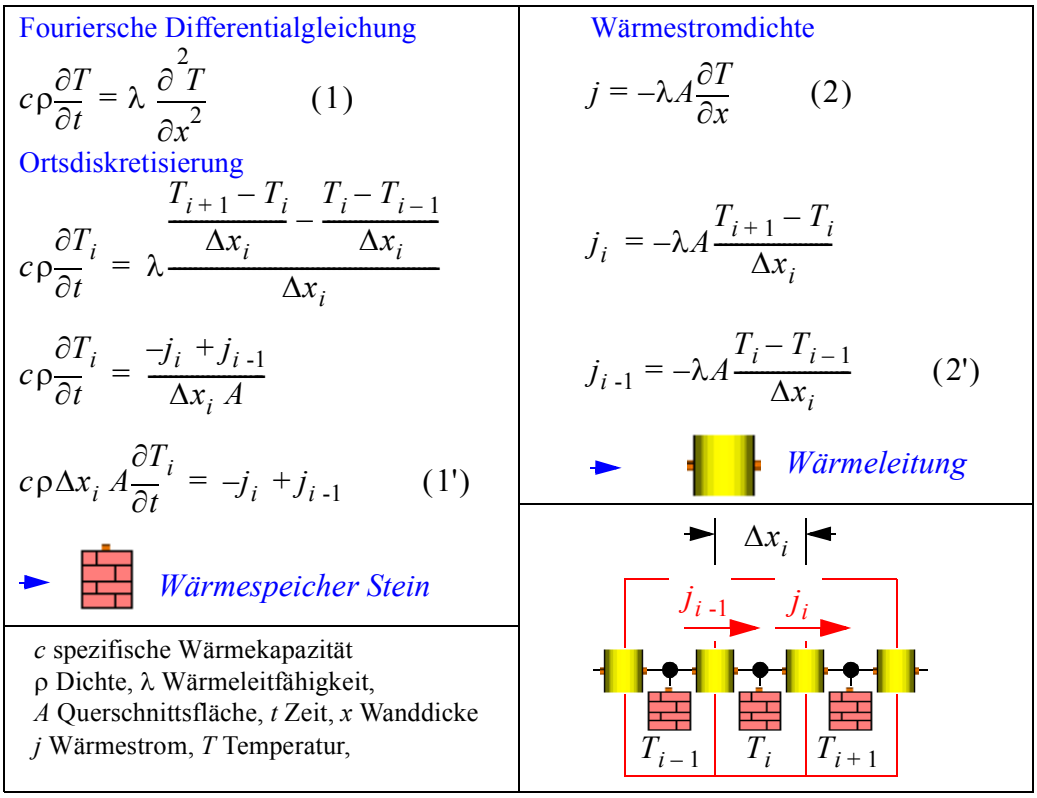


Abbildung 55: Ortsdiskretisierung der Fourierschen Differentialgleichung

Die im Beispiel Abb. 54 genutzte Methode zur Beschreibung des wärmetechnischen Verhaltens einer Wand nutzt das Beuken-Modell *([Beu-36], z.B. in [RoZi-97])*. Die Wand wird durch einen von Wärmewiderständen und konvektiven Wärmeübergängen umgebenen Speicher einheitlicher Temperatur dargestellt. Mathematisch entspricht dies der zunächst sehr groben Ortsdiskretisierung der Fourierschen eindimensionalen, partiellen Differentialgleichung zur Beschreibung von Wärmeleitungsprozessen in festen Stoffen (Gleichung (1) in Abb. 55).

Das Wandmodell ist entsprechend der Fragestellung skalierbar, denn durch die Verschaltung mehrerer Komponenten des Types *Wärmeleitung* bzw. des Types *Wärmespeicher* lässt sich die Anzahl der örtlichen Stützstellen erhöhen. So kann außer der groben Bilanzierung der Wärmeströme auch beispielsweise das Temperaturprofil innerhalb einer Wand berechnet wer-

den. Auch können komplexe Wandaufbauten mit Schichten unterschiedlicher physikalischer Eigenschaften bzw. auch die konvektiven Wärmeübergänge an der Wandoberfläche beschrieben werden.

Die in den Wandmodellen verschalteten Komponenten (*Wärmeleitung, Wärmeübergänge* und *Wärmespeicher*) können durch die Farbcodierung der den Modellklassen zugeordneten Icons unterschieden werden (Abb. 56). *Gelb* (□) symbolisiert ein *Wärmeleitungselement, rot* (□) einen *Wärmespeicher*, *blau* (□) einen konvektiven *Wärmeübergang*. Der im Icon enthaltene generische Parameter (siehe Abschnitt 2.3) gibt die Anzahl der in Reihe geschalteten elementaren Schichtbausteine an.

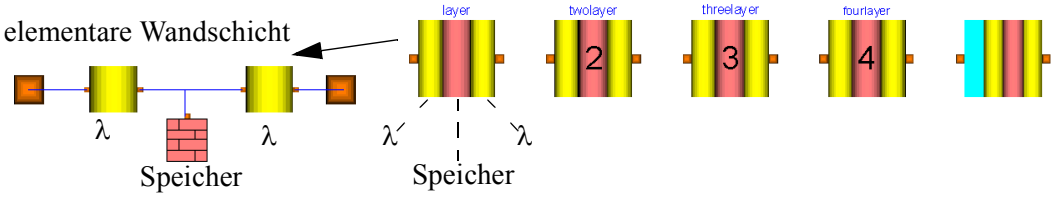


Abbildung 56: Verschaltung einzelner Wandschichten zu Halbwänden bzw. Wänden.

5.1.4 Die thermische Zone

5.1.4.1 Abgrenzung und Definition

Die grundlegende Strukturierungsidee eines Gebäudemodells resultiert in der Notwendigkeit, möglichst klar definierte Module zu erzeugen. Den ersten Anhaltspunkt bietet die Unterteilung des Gebäudes in Räume. In Übereinstimmung mit der VDI Richtlinie [VDI-6020] werden zum Raum alle wärmespeichernden Raumumschließungen (Wände, Fußboden, Decke), alle nicht wärmespeichernden Raumumschließungsflächen (Fenster), die Raumluft, sowie alle inneren Speichermassen (Mobiliar und Personen) gerechnet. Häufig werden Räume mit gleichem Benutzerverhalten und ähnlicher Exposition bezüglich der äußeren klimatischen Einflüsse zu thermischen Zonen zusammengefasst und mit einheitlichen physikalischen Eigenschaften beschrieben. Die Bilanzierung der Wärmeströme erfolgt so im Raumluftknoten der thermischen Zone. Besteht eine thermische Zone aus mehreren Räumen, so müssen auch die inneren Trennwände als innere Speichermasse aufgefasst werden.

Die Aufteilung eines Gebäudes in thermische Zonen erfolgt bei der hier angewendeten Methode durch Zerschneidung aller Wände innerhalb der Wandmitten. Jede der Halbwände wird der näherliegenden thermischen Zone zugerechnet. Die so zu einer thermischen Zone gehörigen Halbwände werden im folgenden als Innenwandhälfte bezeichnet. Sie besitzen individuelle Speichertemperaturen und stehen nach außen nur durch den direkten Kontakt, also die Wärmeleitung im Festkörper, mit den anschließenden Wandhälften im Austausch. Die verbleibenden äußeren Halbwände aller Außenwände des Gebäudes können auch individuelle Speichertemperaturen besitzen und stehen über noch näher zu beschreibende Mechanismen mit der Außenwelt im Austausch (Abb. 57).

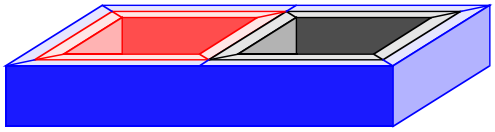


Abbildung 57: Aufteilung eines Gebäudes in thermische Zonen, die Boden und Dekkenflächen wurden nicht dargestellt.

Zur Abschätzung der zu erwartenden Modellordnung gelangt man unter Annahme eines würfelförmigen Gebäudes mit *n* würfelförmigen *thermischen Zonen* zu einer Modellordnung kleiner als 13 *n* (Abb. 58).

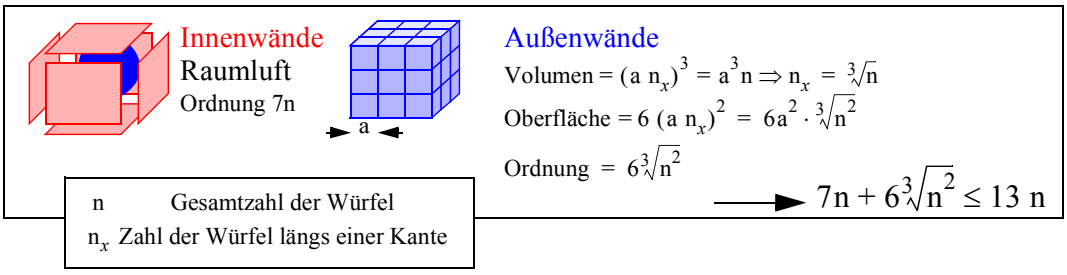


Abbildung 58: Abschätzung der Modellordnung

5.1.4.2 Bilanzierung der Wärmeströme

Die Bilanzierung der Wärmeströme erfolgt im Raumluftknoten. Die zu berücksichtigenden Wärmeströme umfassen entsprechend der VDI Richtlinie [VDI-6020] die konvektiven Wärmeströme von den Hüllflächen, die Wärmeabgabe und Aufnahme der inneren Lasten, Wärmeströme durch Infiltration (Undichtigkeiten) und Ventilation (Lüftung). Hinzu kommt die konvektive Wärmeübergabe von Heiz- und Kühlgeräten durch innere Wärmequellen wie Personen, Beleuchtung oder Maschinen. Die gleichen Wärmeströme werden auch im TRNSYS-Gebäudemodell (Abb. 34) zur Bilanzierung der Wärmeströme im Raumluftknoten berücksichtigt.

5.1.4.3 Wärmeaustausch der Innenwände

Innerhalb der thermischen Zone stehen die Wandflächen vorwiegend über zwei Mechanismen miteinander im Austausch.

Konvektion

Die natürliche bzw. erzwungene Konvektion der Raumluft kann bei bekannten Wärmeübergangskoeffizienten entsprechend dem einführenden Beispiel aus Abschnitt 5.1.2 beschrieben werden. Der Raumluftknoten befindet sich im Zentrum einer sternförmigen Verschaltung aller Innenwände. Die sternförmige Verschaltung greift auf der Seite der konvektiven Wärmeübergänge an den Innenwänden an.

Langwelliger Strahlungsaustausch

Die Innenwände tauschen als graue Temperaturstrahler nach dem Stefan-Boltzmann-Gesetz Energie durch langwelligen Strahlungsaustausch aus [HMS-89]. Die zwischen zwei beliebig orientierten Flächen ausgetauschte Strahlungsenergie ergibt sich durch Integration des photometrischen Grundgesetzes über die Strahlaustauschflächen A_1 und A_2 [Fei-94]. Der Abstand der Flächen sei *r*. Der Winkel zwischen der Flächennormalen und der Strahlrichung sei β_1 bzw. β_2. Der ausgetauschte Wärmestrom J_{12} ist unter Vernachlässigung von Reflexionen in Abb. 59 aufgeführt.

Eine Beschreibung des langwelligen Strahlungsaustausches nach dieser Methode ist aufwändig. Denn neben der numerischen Berechnung der zweifachen Integration ist eine exakte Parametrierung der geometrischen Lagedaten aller Flächen notwendig.

Dimensionslose Einstrahlzahl $$\varphi_{1\to 2} = \frac{1}{\pi A_1}\int_{A_1}\int_{A_2}\frac{cos\,\beta_1 cos\,\beta_2}{r^2}dA_1 dA_2$$

Stefan-Boltzmann-Konstante $$\sigma = 5.670 \cdot 10^{-8}\frac{W}{m^2 K^4}$$

Resultierender Wärmestrom zwischen A_1 und A_2 $$J_{12} = A_1\varepsilon_1\varepsilon_2\sigma\varphi_{1\to 2}\cdot(T_1^4 - T_2^4)$$

A_1 A_2	Fläche
ε_1 ε_2	Emissionsgrad
β_1 β_2	Winkel zwischen Flächennormale und Strahlrichtung
r	Abstand der Flächen

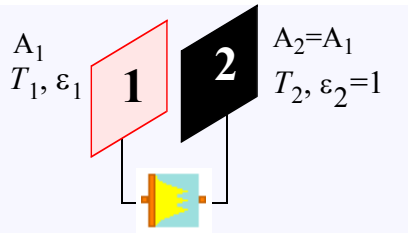
Abbildung 59: Berechnung des Strahlungsaustausches zweier beliebig geneigter Flächen

Man wendet daher meist die Näherung des sogenannten Zweisternmodells [Fei-94] an, zu mal die Ergebnisse für eine nicht ortsauflösende Bilanzierung der Wärmeströme in den Speicherknoten ausreichend ist.

Hierbei wird der Strahlungsaustausch der Hüllflächen untereinander nicht mehr direkt, sondern nach Zwischenabsorption an einem den ganzen Raum ausfüllenden, masselosen, ideal schwarzen Körper unendlicher Wärmeleitfähigkeit beschrieben.

Aufgrund der Annahme eines schwarzen Körpers gilt für den Absorptionsgrad $\alpha_2^{abs} = \varepsilon_2 = 1$. Da er den Raum vollständig ausfüllt, kann $A_2 = A_1$ und $\varphi_{1\to 2} = 1$ angenommen werden.

Der Strahlungsaustausch zwischen jeder der Wandflächen A_i und dem Strahlungsknoten berechnet sich nach der Gleichung $J_{i\,Stern} = A_i\varepsilon_i\sigma\cdot(T_i^4 - T_{Stern}^4)$ und kann über einen Strahlungskoppler (Abb. 60) berechnet werden. Die sternförmige Verschaltung der einzelnen Strahlungskoppler beschreibt so eine Aufteilung des Wärmestroms entsprechend den Produkten $A_i\varepsilon_i$.



Strahlungskoppler

$$J_{1 \leftrightarrow 2} = A_1 \varepsilon_1 \sigma \cdot (T_1^4 - T_2^4)$$

$\varepsilon_1, \varepsilon_2$ = Emissionsgrad (=Absorptionsgrad)

$\sigma = 5.67 \cdot 10^{-8} \mathrm{W/m^2K^2}$

Abbildung 60: Modell eines Strahlungskopplers: er beschreibt den langwelligen Strahlungsaustausch zwischen der Fläche 1 und einer parallelen, gleich großen schwarzen Fläche 2.

5.1.4.4 Implementierung

In Abb. 61 ist die Implementierung des Komponentenmodells einer fensterlosen, thermischen Zone mit 6 möglichen Nachbarn in Form der graphischen Darstellung sowie des Objektdiagrammes abgebildet. Die Hüllflächen werden entsprechend der Aufteilung der Wände in Halbwände auf Halbwandmodelle mit einer zum Raum zeigenden, konvektiven Seite abgebildet. Die Zahl der möglichen Nachbarn folgt aus der Zahl verfügbarer äußerer Schnittstellen. Im Zonenmodell befinden sich die Innenwandhälften, die Raumluftknoten sowie die zusätzlichen inneren Lasten als speichernde Elemente.

Der masselose Strahlungsknoten ist in Abb. 61 als rote Verbindungslinie zu erkennen. Er wird sternförmig über Strahlungskoppler (Abb. 60) mit den Innenwänden verschaltet. Für den Strahlungsaustausch der Innenwände ist die Oberflächentemperatur relevant. Die Verbindung der Strahlungskoppler an die Wände muss daher „hinter“ dem konvektiven Wärmeübergang erfolgen. Die Wände besitzen daher separate Schnittstellen, die den konvektiven Wärmeübergang umgehen.

Der Raum verfügt so über zwei vollständig voneinander getrennte Knoten, den Raumluftknoten und den Strahlungsknoten.

Ein direkter Anschluss an den Raumluftknoten wird als Schnittstelle nach außen geführt. Dies eröffnet die Möglichkeit eine externe Raumluftheizung, einen Temperatursensor oder den Luftaustausch mit einer Nachbarzone anzubinden.

Durch Aufnahme weiterer Komponenten lassen sich die Modelle der thermischen Zonen detaillieren. So ist in Abb. 62 ein Raummodell mit 7 möglichen Nachbarn, internen Gewinnen sowie einem Fenster dargestellt. Die internen Gewinne werden durch vom Raum aggregierte Bausteine eingebracht. Der konvektive Anteil der internen Gewinne wird in den Raumluftknoten eingespeist. Der Strahlungsanteil wird entsprechend den Absorptionskoeffizienten und Flächengrößen den Oberflächen der Innenwandhälften zugeführt.

Die Behandlung der kurzwelligen solaren Einstrahlung.
Die vom Fenster transmittierte kurzwellige, solare Einstrahlung wird über eine eigene gerichtete Schnittstelle, den *Strahlungscut (* *),* in den Raum geführt. Die Einstrahlung wird nicht in den Strahlungsknoten eingespeist, sondern über einen zentralen Verteiler auf die Innenwandoberflächen der thermischen Zone geführt. Dies hat den Vorteil, dass für die kurzwellige, solare Einstrahlung eigene Absorptionskoeffizienten eingestellt werden können. Auch ist es möglich bei besonderen geometrischen Verhältnissen Wichtungen einzuführen.

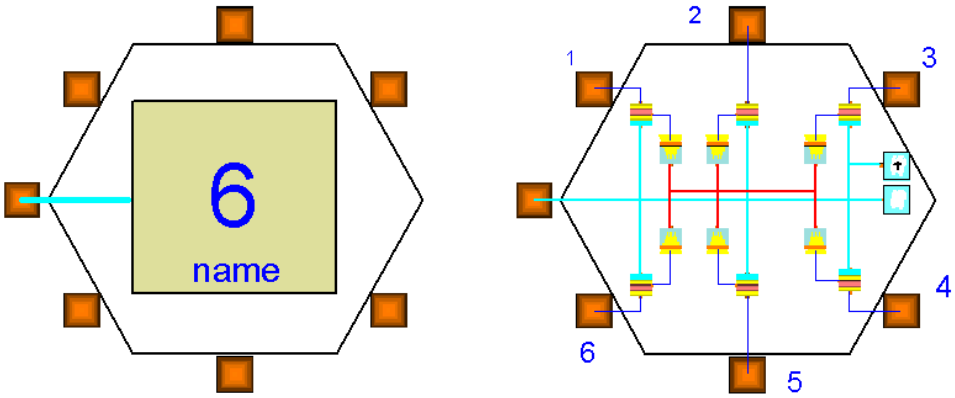


Abbildung 61: Graphische Darstellung und Objektdiagramm einer thermischen Zone mit 6 möglichen Nachbarn.

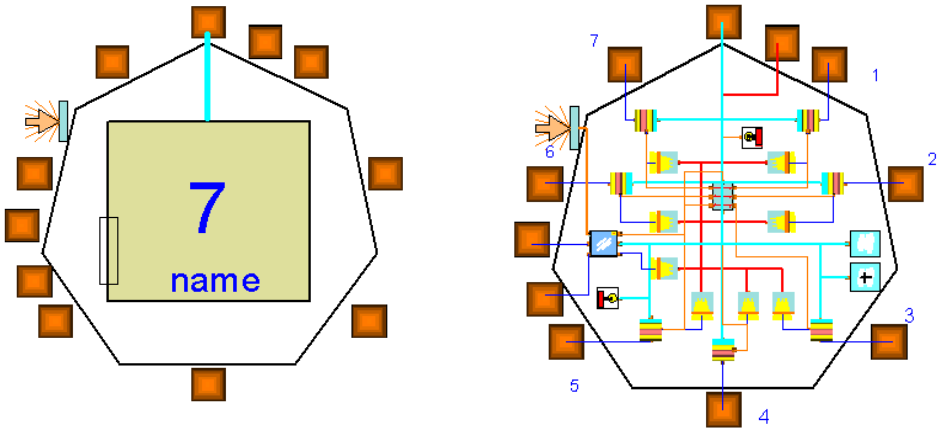


Abbildung 62: Graphische Darstellung und Objektdiagramm einer thermischen Zone mit 7 möglichen Nachbarn und einem Fenster

5.1.5 Das Fenstermodell

Die quantitative Beschreibung dieser Wärmeströme erfolgt innerhalb des Fenstermodells. Die Implementierung des Fenstermodells ist in Abb. 64 dargestellt.

5.1.5.1 Wärmeaustausch

Da das Fenster zu den Außenflächen des Gebäudes gehört, steht es über konvektive Wärmeübergänge im Austausch mit der Außenluft und der Raumluft. Es verfügt aufgrund der geringen Masse über keinen Wärmespeicher. Dieser meist mit einem k-Wert spezifizierte Wärmeverlustpfad findet sich in Abb. 63.

Darüber hinaus befindet sich das Fenster als *„Temperaturstrahler“* durch langwellige Strahlung im Austausch mit seiner äußeren Umgebung sowie den Innenwandoberflächen der thermischen Zone. Die Schnittstellen des langwelligen Strahlungsaustausches benötigen die Oberflächentemperatur und greifen unter Umgehung der konvektiven Wärmeübergänge (α) über die in Abb. 63 rot eingezeichneten Verbindungslinien direkt am Wärmewiderstandselement (λ) an. Die äußere Schnittstelle zur Außenwelt wird, wie im Falle aller Außenflächen, zum Modellbaustein *fiktive Himmeltstemperatur* geführt, die innere Schnittstelle zum Strahlungsknoten der thermischen Zone.

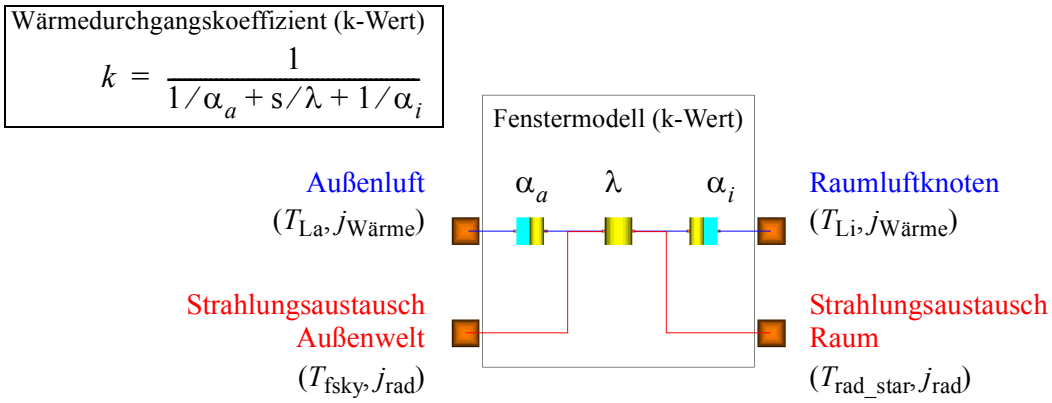


Abbildung 63: Fenstermodell mit k-Wert und Strahlungsaustausch

5.1.5.2 Luftaustausch

Jedes reale Gebäude steht im Luftaustausch mit seiner Umgebung. Hierfür sind zum einen Undichtigkeiten der Gebäudehülle wie beispielsweise Fugen und Ritzen im Bereich der Fenster und Türen verantwortlich. Die Stärke des Luftaustausches durch Fugen und Ritzen hängt von den lokalen Druckverhältnissen vor der Gebäudefassade und damit von der aktuellen Windgeschwindigkeit und Richtung bezüglich der Lage der betreffenden Fenster oder Türen ab. Zum anderen erfordert ein gesundes Raumklima den Austausch verbrauchter Luft durch Frischluft. Während diese Aufgabe in modernen Bürogebäuden meist einer Belüftungsanlage mit Wärmetauschern übertragen ist, erfolgt die Lüftung in Einfamilienhäusern des Bestandes meist durch manuelles Öffnen der Fenster.

Beide Vorgänge bewirken einen Austausch warmer Raumluft durch kalte Außenluft oder umgekehrt. Den bisherigen Wärmeverlustpfaden ist in Abb. 64 ein neuer Pfad zur Beschreibung der Lüftungswärmeverluste parallel geschaltet. Die vorgegebene Frischluftrate wird mittels eines *Input-Cuts* in die Modellkomponente eingekoppelt.

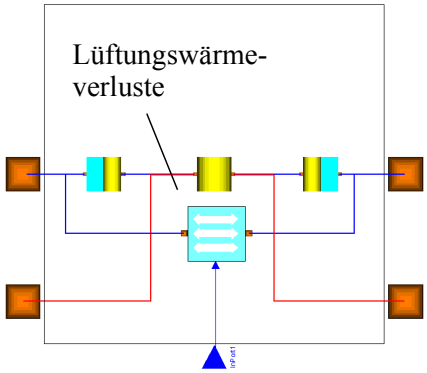
Lüftungswärmeverluste

$j_{\mathrm{Li}\rightarrow\mathrm{La}} = c_{\mathrm{L}} \cdot \rho_{\mathrm{L}} \cdot V_{\mathrm{L}}(t) \cdot (T_{\mathrm{Li}} - T_{\mathrm{La}})$

c_{L} = spez. Wärmekap. von Luft $[\mathrm{J/kg}]$

ρ_{L} = Dichte von Luft $[\mathrm{kg/m^3}]$

$V_{\mathrm{L}}(t)$ = Frischluftrate $[\mathrm{m^3/h}]$



Zeitschema-Cut $V_{\mathrm{L}}(t)$ (Input)

Abbildung 64: Fenstermodell mit k-Wert, Strahlungsaustausch und Lüftungswärmeverlusten

5.1.5.3 Solare Einstrahlung

Zur Bestimmung der solaren Einstrahlung werden gleich orientierte Außenflächen zu Gebäudefassaden, spezifiziert durch ihren Normalenvektor, zusammengefasst. Die in Form diffuser und gerichteter (*Beam*-) Strahlung auf eine horizontale Fläche angegebenen Strahlungsdaten kommerzieller Wetterdatenfiles sind zunächst geeignet aufzubereiten. Diese Aufgabe übernehmen die *Solarmodelle* (siehe auch Abschnitt 5.3.2). Von ihnen wird aus der Simulationszeit die solare Zeit und in der Folge der aktuelle Sonnenstand in Form des Azimut- und Zenitwinkels berechnet. Diese Information genügt, um aus der momentanen solaren *Beam*-Strahlung auf eine horizontale Fläche die Bestrahlungsstärke für jede Gebäudefassade längs ihrer Normalen pro Flächeneinheit durch Projektion auszurechnen. Der Anteil der diffusen Strahlung wird hinzuaddiert, wobei im einfachsten Fall von einer isotropen Verteilung ausgegangen wird. Es stehen jedoch auch alle beispielsweise von TRNSYS genutzten Strahlungsmodelle (siehe Abb. 41) zur Verfügung. Die Summe der geeignet auf die Außenfläche fallenden Strahlungsdichten in $\mathrm{W/m^2}$ wird über den gerichteten *Input-Strahlungscut* dem Fenster zugeführt.

Um zu vermeiden, dass pro Fassadenfläche ein eigenes Projektionselement eingesetzt werden muss, wird in einem übergeordneten Projektionselement außerhalb des Gebäudes die solare Einstrahlung für eine gewisse Anzahl vordefinierter Flächenorientierungen ausgerechnet. Die Ergebnisse können so als Vektor gepackt und in gleicher Weise mit einer gemeinsamen Leitung an alle Außenflächen des Gebäudes weitergereicht werden.

Die Komponenten des Vektors charakterisieren die solare Einstrahlung auf die vorab als möglich deklarierten Vorzugsorientierungen. Zur intuitiveren Parametrierung ist es möglich signifikante Benennungen wie *SüdVertikal*, *NordVertikal* oder *SüdGeneigt* zur Charakterisierung der Vorzugsrichtungen einzusetzen. Da diese Definition sowohl dem Projektionselement als auch allen Außenflächen des Gebäudemodells zur Verfügung stehen sollte, ist sie in einer übergeordneten Hierarchieebene zu definieren. Die Anbindung der solaren Einstrahlung an das Gebäudemodell kann so durch eine einzelne vektorwertige Verbindung zwischen dem Projektionselement und allen betroffenen Außenflächen realisiert werden.

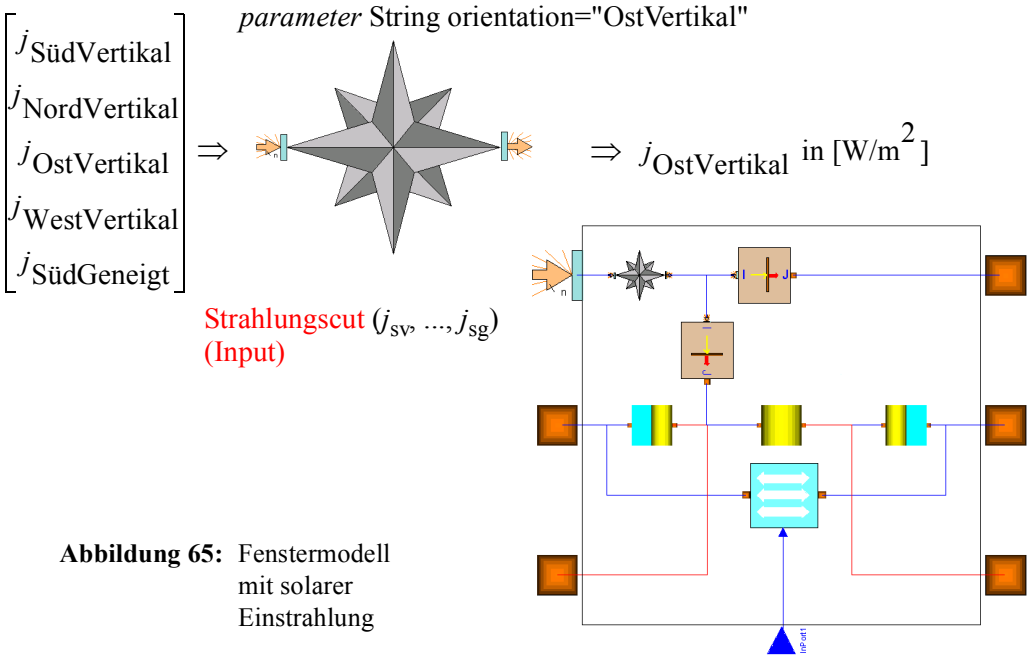


Abbildung 65: Fenstermodell mit solarer Einstrahlung

Das Fenster kennt seine Zugehörigkeit zu einer der möglichen Fassadenrichtungen und kann die richtige Komponente des Strahlungsvektors auswählen. (Dies erfolgt in dem rosettenförmigen Baustein in Abb. 65.)

An dieser Stelle ist zu berücksichtigen, dass das Fenster auch einen Anteil der Strahlung absorbiert. Während der transmittierte Anteil der Strahlung in den Raum gelangt, wird der absorbierte Anteil der Strahlung dem Fenster selbst zugeschlagen. Er verteilt sich im Fenster, das masselos und somit nicht speichernd angenommen wurde und fließt über die konvektiven Wärmeübergänge bzw. auch über den langwelligen Strahlungsaustausch ab.

Aus der Gesamtstrahlungsdichte in $[W/m^2]$ wird zunächst die absolute transmittierte und absorbierte Strahlung in [W] berechnet. Dies geschieht in den *Transmissions*- und *Absorptionselementen* anhand des Absorptionskoeffizienten und der Flächengröße des Fensters. Der nach außen reflektierte Anteil braucht nicht weiter betrachtet zu werden, da dort keine energetische Bilanzierung erfolgt.

5.1.6 Die ideale Heizung

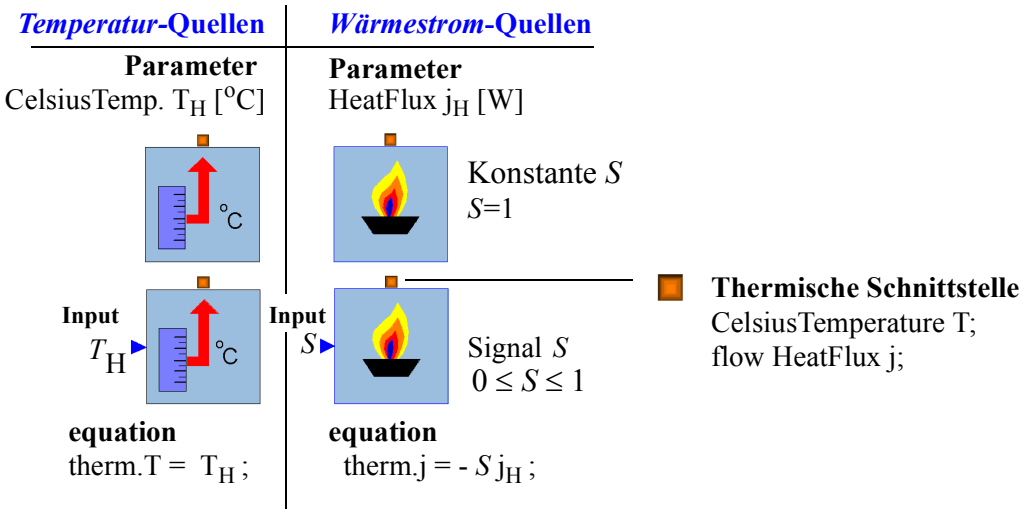


Abbildung 66: Modell einer idealen Heizung

Die Aufgabe der Heizungsbausteine (Abb. 66) ist es, einen positiven oder auch negativen Energieeintrag innerhab der thermischen Zone zu beschreiben. Dies kann durch Festlegung des einfließenden Wärmestroms an der Schnittstelle geschehen. Dann sollte jedoch über die Temperatur an der Schnittstelle keine Aussage getroffen werden. Sie wird sich passend einstellen. Dies Modell beschreibt eine im Zeitverhalten ideale, aber bezüglich der Heizleistung limitierte Heizung. Andererseits kann auch die Temperatur an der Schnittstelle festgelegt werden; dann darf jedoch keine Aussage über den Wärmestrom erfolgen. Beide Konzepte ähneln der Beschreibung eines elektrischen Schaltkreises. Hierbei unterscheidet man Spannungsquellen und Stromquellen. Daher werden die beiden unterschiedlichen Wärmequelletypen im folgenden auch *Temperaturquellen* und *Wärmestromquellen* genannt.

Zur praktischen Nutzung gibt es von beiden Wärmequellen eine konstante und eine steuerbare Version. Bei letzterer wird der Ausgang proportional mit einem Steuersignal gefahren. Das Zeitverhalten kann über ein vorgeschaltetes Verzögerungsglied in die Beschreibung aufgenommen werden. Daher wird für den Signaleingang eine *Modelica-Input-Schnittstelle* gewählt, die kompatibel zu den Blockschaltbildkomponenten der *Modelica-Standard*-Bibliothek ist.

5.2 Die Thermohydraulik-Bibliothek

Die *Thermohydraulik*-Bibliothek (Abb. 68) wurde zur Beschreibung einer Warmwasserzentralheizung entwickelt. Die Rohrströmung wird vorwiegend statisch wie ein elektrischer Schaltkreis behandelt. Dies bedeutet: die Rohrleitungselemente werden analog Widerständen, die Pumpen analog elektrischen Stromquellen beschrieben. Darüberhinaus existieren Verzweigungs- und Verbindungsrohre, Ventile, ein Mischer zum Aufbau eines zweistufigen Heizkreises und Sensoren. Das Rohrmodell kann auch als Wärmetauscher eingesetzt werden, um so als Radiator-Heizkörper oder als Teil einer Warmwasserfußbodenheizung zu fungieren. Allen thermohydraulischen Komponenten sind die in Abb. 67 dargestellten *thermohydraulischen Schnittstellen* gemein. Hierbei wird die Temperatur und der Druck des Heizwassers an

der Schnittstelle als Potentialvariable übergeben, der Volumenstrom als Flussvariable. Da in Rohrleitungssystemen meist eine Vorzugsrichtung existiert wurden im Gegensatz zur *thermischen Schnittstelle* der *Gebäude-Bibliothek* zwei unterschiedlich markierte, ansonsten aber identische Schnittstellen definiert.

„Hydraulische“ Schnittstellen

CelsiusTemperature T;
Pressure p;
flow VolumeFlowRate q;

Abbildung 67: Einheitliche Schnittstellendefinition:
Thermohydraulische Schnittstellen

Die Wärmetauscher verfügen über zusätzliche *thermische Schnittstellen* der *Gebäude-Bibliothek*. Die Konsequenz dieses Konzeptes ist, dass eine Rohrleitung zwar einfach in zwei Zweige aufzuspalten ist, aber das Zusammenführen nur über ein eigenes Bauteil geschehen kann. Denn falls das Heizwasser nach Durchlaufen der beiden Zweige unterschiedliche Temperaturen aufweist, würde die Potentialvariable identische Werte erzwingen; es muss jedoch eine mit der Größe des jeweiligen Massenstroms gewichtete Mittelwertbildung erfolgen. Diese Aufgabe übernehmen die Zusammenführungen (Abb. 68).

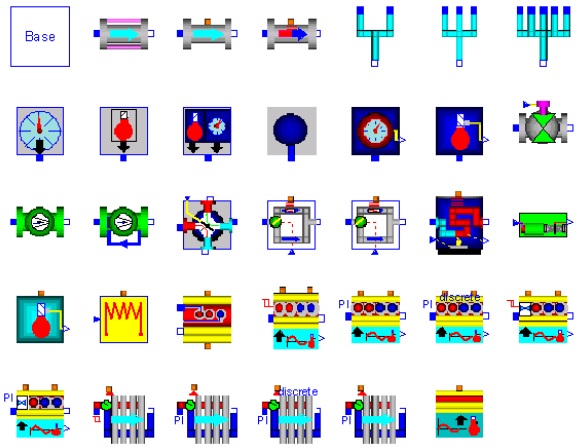


Abbildung 68: Hydraulik-Bibliothek zur Beschreibung einer Warmwasserheizungsanlage

5.2.1 Das Rohrmodell

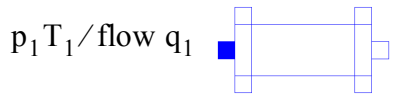
In der *thermohydraulischen* Bibliothek befinden sich unterschiedliche Rohrmodelle zur Beschreibung verschiedener Aspekte. Die Vererbungshierarchie (Abb. 71) erlaubt eine Auftrennung der physikalischen Beschreibung in einen hydraulischen und einen thermischen Anteil.

Abbildung 69: Vererbungshierarchie am Beispiel des Rohrmodells

5.2.1.1 Hydraulischer Anteil

Die Beschreibung des hydraulischen Anteils erfolgt in Analogie zu einem elektrischen Schaltkreis. Die Strömung wird als stationär und inkompressibel angenommen. Im Falle der laminaren Rohrmodelle berechnet sich der Druckabfall längs eines Rohrstücks in Abhängigkeit des Volumenstromes nach dem Hagen-Poisseuille'schen Gesetz (Abb. 70) [HMS-89]. Andere Rohrmodelle bestimmen zunächst anhand der Reynoldszahl den vorliegenden Strömungszustand, und wählen eine geeignete Beschreibungsform aus (Abb. 71). Darüberhinaus ist die Temperaturabhängigkeit der dynamischen Viskosität (η) zu berücksichtigen. Es wird eine abschnittsweise lineare Näherung genutzt [Glü-91].

Problematisch ist neben der Signifikanz der ausgewählten physikalischen Beschreibung im Grenzgebiet zwischen turbulenter und laminarer Strömung die numerische Problematik des diskreten Umschaltens. Jede ereignisgesteuerte Abfrage unterbricht kurzzeitig die kontinuierliche Simulation, um das Eintreten des Ereignisses zu überprüfen, und führt damit zu einer verlängerten Simulationszeit. Gerade bei großen Systemen empfiehlt es sich daher, ein festes Strömungsmodell auszuwählen.



$$q_1 = -q_2$$

$$p_1 - p_2 = \frac{8\eta l}{\pi R^4} \cdot q_1 \quad \text{(Hagen-Poiseuille-Gesetz)}$$

$p_{1,2}$ = Druck
$T_{1,2}$ = Temperatur
$q_{1,2}$ = Volumenstrom
η = dyn. Viskosität
R, l = Rohrradius, -länge

Abbildung 70: Hydraulisches Rohrmodell unter Annahme einer laminaren Strömung

$$Re = \frac{\rho \bar{v} D}{\eta}$$

Re < 2320
laminare Strömung

2320 < Re < 10^5
turbulente Strömung

turbulente Strömung

$$p_1 - p_2 = \lambda \cdot \frac{l}{2R} \cdot \frac{1}{2} \rho \bar{v}^2$$

Re = Reynoldszahl
ρ = Dichte
$\bar{v}$ = mittl. Geschwindigkeit = $q_1 / \pi R^2$
D = charakt. Länge, z.B. = 2R
η = dyn. Viskosität
R, l = Rohrradius, -länge
λ = $0.3164 / \sqrt[4]{Re}$
Rohrreibungszahl nach Blasius

Abbildung 71: Laminare und turbulenter Strömung

5.2.1.2 Thermischer Anteil

Bei der Beschreibung des thermischen Verhaltens kann je nach Aufgabenstellung zwischen verschiedenen Modellen ausgewählt werden. Ihre Entwicklung geht von einer partiellen Differentialgleichung aus, die der Bilanzierung der Energieströme in einem infinitesimal kleinen Scheibchen des Rohrvolumens entstammt (Abb. 72). Orts- und zugleich zeitabhängige Differentialgleichungen sind in *Modelica* nicht direkt zu implementieren. Es bieten sich zwei Lösungsmethoden (*A, B* in Abb. 73) an.

A) Bei der Ortsdiskretisierung durch Modularisierung, ähnlich der Wandbeschreibung nach dem Beukenmodell, wird jedes infinitesimal kleine Bilanzvolumen auf eine elementare Modellkomponente abgebildet. Die Anzahl der Modellkomponenten entspricht der Anzahl der örtlichen Stützstellen.

B) Für größere Anlagen ist der Einsatz dieser Modelle jedoch sehr aufwändig. Da das Speichervermögen des Heizwassers in den Rohren im Vergleich zum Speichervermögen der massiven Bauteile (Betonplatte der Fußbodenheizung, Radiatorkörper, Mauerwerk) relativ klein ist, kann häufig eine stationäre Näherung eingesetzt werden. Die partielle Differentialgleichung geht über in eine reine zeitliche Differentialgleichung, die analytisch lösbar ist und ein exponentielles Temperaturgefälle beschreibt. Diese Näherung wird insbesondere bei langen Rohrleitungssystemen (z.B. Heizschlangen der Fußbodenheizung) eingesetzt.

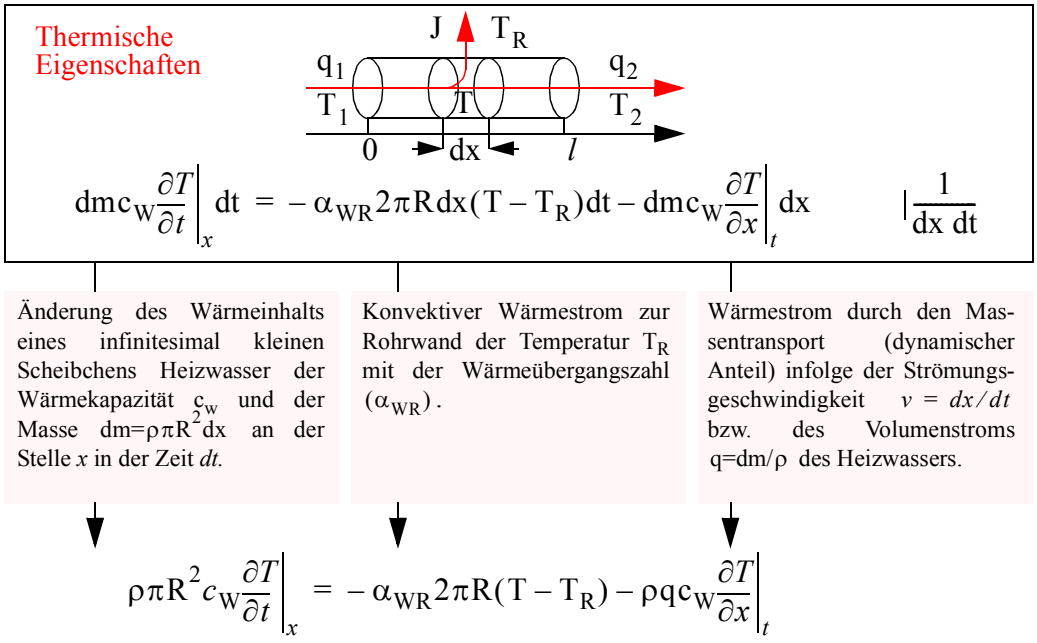
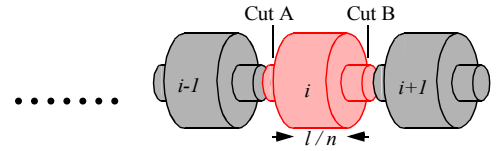


Abbildung 72: Partielle Differentialgleichung zur Beschreibung des thermischen Verhaltens

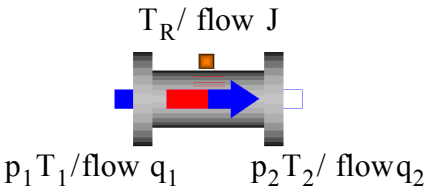
A) Lösung durch Ortsdiskretisierung in n Module der Länge l/n



$$\rho\pi R^2 c_W \frac{dT_{CutB}}{dt} = -\alpha_{WR} 2\pi R (T_{CutB} - T_R) - \rho q c_W \frac{T_{CutB} - T_{CutA}}{l/n}$$

B) Lösung durch stationäre Näherung, d.h. $\left.\frac{\partial T}{\partial t}\right|_x = 0$

$$\int_0^l \alpha_{WR} 2\pi R \, dx = \int_{T(0)=T_1}^{T(l)=T_2} \frac{-\rho q_1 c_W}{T - T_R} dT \rightarrow \alpha_{WR} 2\pi R \, l = -\rho q_1 c_W \ln \frac{T_2 - T_R}{T_1 - T_R}$$

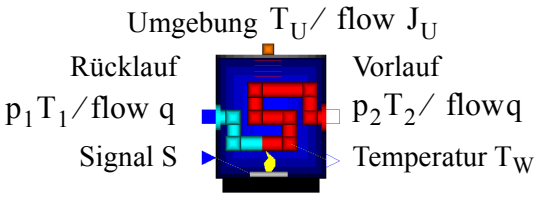


$$T_2 - T_R = (T_1 - T_R) e^{-\frac{\alpha_{WR} 2\pi R l}{\rho q_1 c_W}}$$

$$J = \rho q c_W \frac{T_{CutB} - T_{CutA}}{l}$$

Abbildung 73: Lösungsmethoden A) und B) zur Berechnung des thermischen Verhaltens.

5.2.2 Das Kesselmodell



A_{Kessel} = Oberfläche Kessel
K_{Kessel} = k-Wert Kessel
P_{Heiz} = max. Heizleistung
J_U = Umgebungsverluste

$$m_W c_W \dot{T}_W = P_{Heiz} \cdot S - \rho q c_W \cdot (T_2 - T_1) - J_U \qquad J_U = K_{Kessel} A_{Kessel} (T_W - T_U)$$

Abbildung 74: Das Kesselmodell

Das Kesselmodell (Abb. 74) besitzt einen hydraulischen Anschluss für den Vorlauf und Rücklauf der Heizkreise, eine thermische Schnittstelle für Wärmeverluste an die Umgebung, einen Signaleingang [0,1] zur Ansteuerung des Brenners und einen Sensorausgang zur Übermittlung der momentanen Wassertemperatur im Kessel. Die maximale Heizleistung ist limitiert. Meistens wird zur Regelung der Kesseltemperatur ein Zweipunktregler mit Hysterese genutzt.

5.2.3 Das Pumpenmodell

Die grundlegenden Pumpenmodelle erzeugen eine feste Druckdifferenz *(rot* in Abb. 75) entsprechend einer konstanten Spannungsquelle oder einen festen Volumenstrom (*grün* in Abb. 75) entsprechend einer konstanten Stromquelle. Neben fest parametrierbaren Pumpen existieren ansteuerbare Pumpen mit einem Signaleingang. Weiterentwickelte Pumpenmodelle beschreiben eine zunehmende Leckage von der Hochdruck- zur Niederdruckseite bei ansteigendem Strömungswiderstand des Heizkreises. Zu ihrer Beschreibung ist es notwendig sogenannte konzentrierte Volumina einzuführen, die als Druckspeicher fungieren [Bea-00] und in diesen die Annahme der Inkompressibilität des Mediums aufzugeben. Parallel zur idealen Pumpe, die unabhängig vom Strömungswiderstand des Heizkreises einen konstanten Volumenstrom fördert, wird eine rückführende Rohrleitung mit festem Strömungswiderstand geschaltet. Der im äußeren Kreis fließende Strom wird so abhängig vom Widerstandsverhältnis des Heizkreises und dem des rückführenden Leckage-Rohrelementes. Auch kann die Pumpe, entsprechend der Realität, nur noch einen begrenzten äußeren Staudruck aufbauen. Diese mit Leckage behafteten, realen Pumpenmodelle sind erforderlich, falls ein Heizkreis durch ein Ventil komplett unterbrochen werden kann.

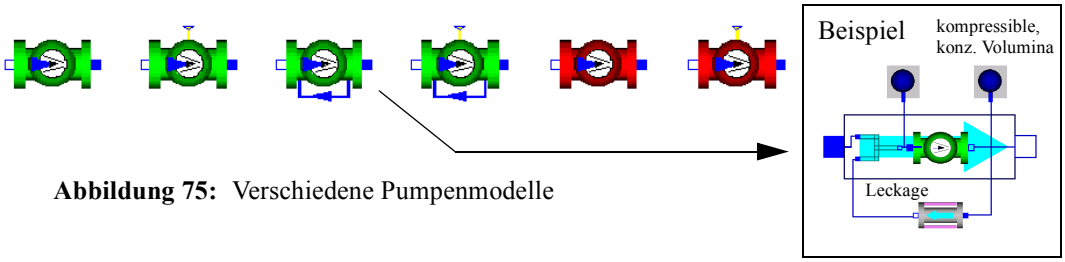


Abbildung 75: Verschiedene Pumpenmodelle

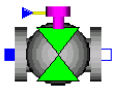
5.2.4 Die Ventilmodelle

Die Aufgabe der einfachen Absperrventile ist es, als Aktoren den Heizwasserstrom und damit auch die übertragene Wärme zu beeinflussen. Sie werden vorwiegend in den Heizkörpern als Aktoren lokaler Regelkreise, sogenannter Raumthermostate, eingesetzt. Die Ventile entsprechen Rohrleitungen mit steuerbarem Querschnitt. Um numerische Probleme zu vermeiden fahren die Ventile nie vollständig zu, sondern reduzieren den Heizwasserfluss um einen Faktor 1000.

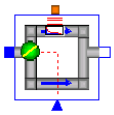
Gerade bei in Reihe geschalteten Heizkörpern benötigt man sogenannte 3 Wegeventile. Sie erlauben es, das Heizwasser an dem abgeschalteten Heizkörper vorbeizuführen. Dies wurde realisiert über eine spezielle Komponente bei der die konvektive Wärmeabgabe über einen Signaleingang mit einem Faktor [0..1] moduliert werden kann. Das hydraulische Verhalten entspricht einem einfachen Rohr. Möchte man den unterschiedlichen Strömungswiderstand des Heizkörpers und der Bye-Pass Leitung berücksichtigen, muss die Schaltung aus modularen Komponenten aufgebaut werden.

Höhere Anforderungen sind von dem 4-Wege-Mischerventil zu erfüllen. Es wird insbesondere bei gemischten Systemen aus Heizwasserradiatoren und Fußbodenheizungen eingesetzt. Die Radiatoren benötigen eine hohe Vorlauftemperatur (z.B. 60-70^{o}C), die nahezu der Kesseltemperatur entspricht, um trotz ihrer kleineren Wärmeaustauschfläche effektiv arbeiten zu können. Die Fußbodenheizung sollte jedoch aus hygienischen Gründen und um Rissbildung im Estrich vorzubeugen mit niedrigeren Temperaturen betrieben werden. (Die Fußbodenoberflächentemperatur darf 35^{o}C nicht übersteigen.) Die Aufgabe des Mischers ist es, kühles Rücklaufwasser aus den Heizkreisen dem heißen Vorlaufwasser unter Umgehung des Kessels

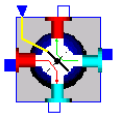
beizumischen und so die Vorlauftemperatur der Fußbodenheizung zu reduzieren. Auch bietet die von der Kesselregelung unabhängige Regelung den Vorteil, über den Mischer kurzfristige Wärmeanforderungen ausregeln zu können, ohne gleich die Kesseltemperatur durch Starten des Brenners zu erhöhen. Aufgrund der Inkompressibilität des Wassers gilt sowohl für den Heizkreis als auch für den Verbraucherkreis, dass die Wassermenge des Vorlaufes gleich der Wassermenge des Rücklaufes ist. Die konzentrierten Volumina der Pumpen werden vernachlässigt. Es ist somit nur ein Wärmestrom vom Heizkreis auf den Verbraucherkreis zu übertragen. Bei einem Signaleingang von 0 sind beide Kreise entkoppelt. Hierbei ist darauf zu achten, dass zumindest die Mischerklappe einen konvektiven Übergang zu beiden Kreisen besitzt, um dem Verbraucherkreis eine Starttemperatur zu übertragen. Bei einem Signal von 1 läuft das Vorlaufwasser der Heizkreise direkt in die Verbraucherkreise ohne Beimischung von kälterem Rücklaufwasser. Beliebige kontinuierliche Zwischenstellungen sind möglich.



Absperr- bzw, Dosierventil



Dreiwegeventil mit Bypass



4 Wegemischer

Abbildung 76: Verschiedene Ventile

5.3 Die Umgebungs-Bibliothek

5.3.1 Die atmosphärische Gegenstrahlung

5.3.1.1 Begriffsbildung

Die Oberfläche des Gebäudes steht als *„Temperaturstrahler“* im Austausch mit der Umgebung. Neben Gebäuden oder der Geländestruktur wirkt vor allem die das Gebäude umgebende Atmosphäre. Gerade die Beschreibung der thermischen Abstrahlung der Atmosphäre ist schwierig zu fassen. Die Atmosphäre stellt eine dicke, für Infrarotstrahlung jedoch relativ transparente Gasschicht dar. Sie besteht überwiegend aus gleichatomaren und damit in erster Näherung infrarotinaktiven Gasmolekülen.

Bei klarem Himmel, besonders in klaren Nächten ist sie teiltransparent. Dies bedeutet: sie strahlt effektiv weniger zurück als ein fester Körper oder anschaulich ausgedrückt: „man *blickt ein wenig auf das kalte Universum hindurch [Kel-97]"*. Im bewölkten bzw. im Nebelzustand verhält sie sich aufgrund des Dipolmoments des kondensierten Wassers anders. Sie verhält sich dann wie ein fester Körper, dessen Temperatur gleich der Außenlufttemperatur ist. Entsprechend strahlt sie auf das Gebäude zurück.

Eine mögliche Beschreibungsform ist daher, einen langwelligen Strahlungsaustausch mit einem fiktiven Körper anzunehmen. Bei starker Bewölkung oder Nebel besitzt der Körper etwa Außenlufttemperatur. Mit sinkender Feuchte ist seine fiktive Temperatur zu reduzieren, bis sie bei klarem Wetter etwa 10-15 K unter der Außenlufttemperatur liegt.

5.3.1.2 Himmelstemperatur

Die Temperatur des fiktiven Körpers wird im allgemeinen Himmelstemperatur (T_{sky}) genannt. Sie hängt ab von der Umgebungstemperatur (T_a) und der relativen Feuchte (φ), dem Bewölkungsgrad des Himmels (C_{Cover}) und dem momentanen Luftdruck (p_{atm}). Die Berechnungen folgen dem in TRNSYS angewendeten Verfahren [Aue-94] und werden in der Komponente *Himmelstemperatur* spezifiziert.

Falls die Wetterdaten den Bewölkungsgrad des Himmels nicht enthalten, kann C_{Cover}, wie Abb. 77 zeigt, aus den aktuellen Daten der solaren Einstrahlung abgeschätzt werden. Nachts liegen jedoch keine Einstrahlungsdaten vor. Der nächtliche Faktor wird daher als konstant angenommen und auf dem letzten bekannten Wert gehalten. *(TRNSYS verwendet hier einen Mittelwert über den vorausgegangenen Nachmittag.)*

Die Himmelstemperatur (T_{sky}) lässt sich unter Anwendung der barometrischen Höhenformel ($p_{atm}(h)$) und eines Näherungsausdruckes für den Emissionsgrad (ε_0) des klaren Himmels berechnen, wie in Abb. 77 gezeigt wird. Hierbei wurde zur Charakterisierung der Feuchte die Sättigungstemperatur T_{Sat} genutzt. Die Berechnung der Sättigungstemperatur (T_{Sat}) erfolgt in einer eigenen Komponente.

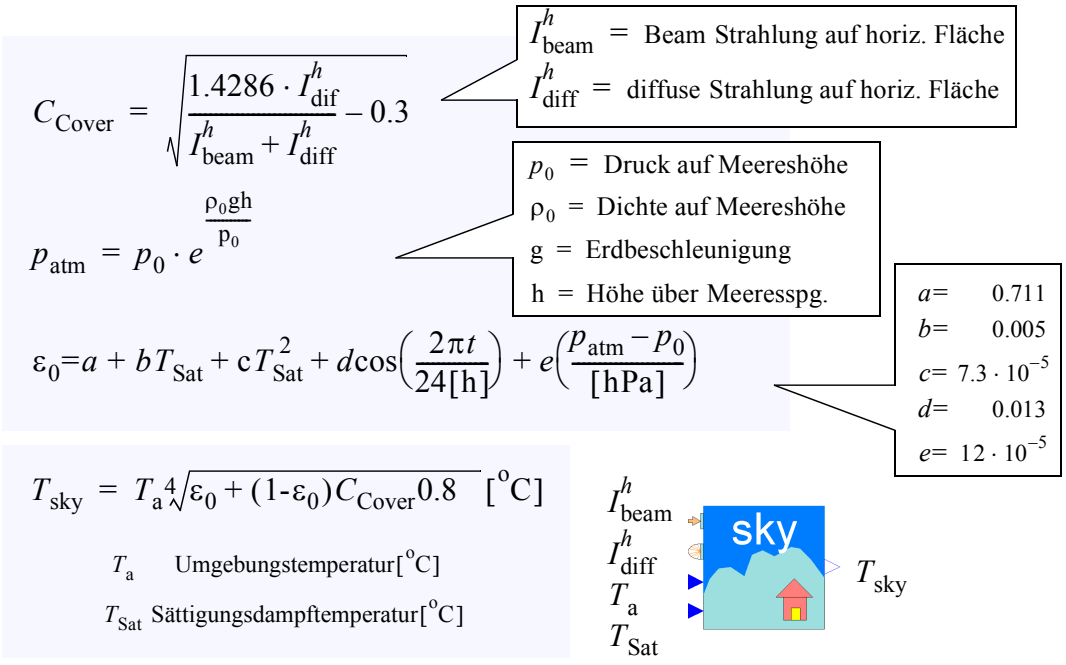


Abbildung 77: Komponente zur Berechnung der Himmelstemperatur

5.3.1.3 Sättigungstemperatur

Die Aufgabe dieser Komponente ist die Berechnung der Sättigungsdampftemperatur der Außenluft bei vorgegebener Umgebungstemperatur (T_a) und relativer Feuchte (φ). Eingangsgrößen sind die relative Feuchte (φ) und die Umgebungstemperatur (T_a). Beide Werte sind in typischen Wetterdatenfiles enthalten.

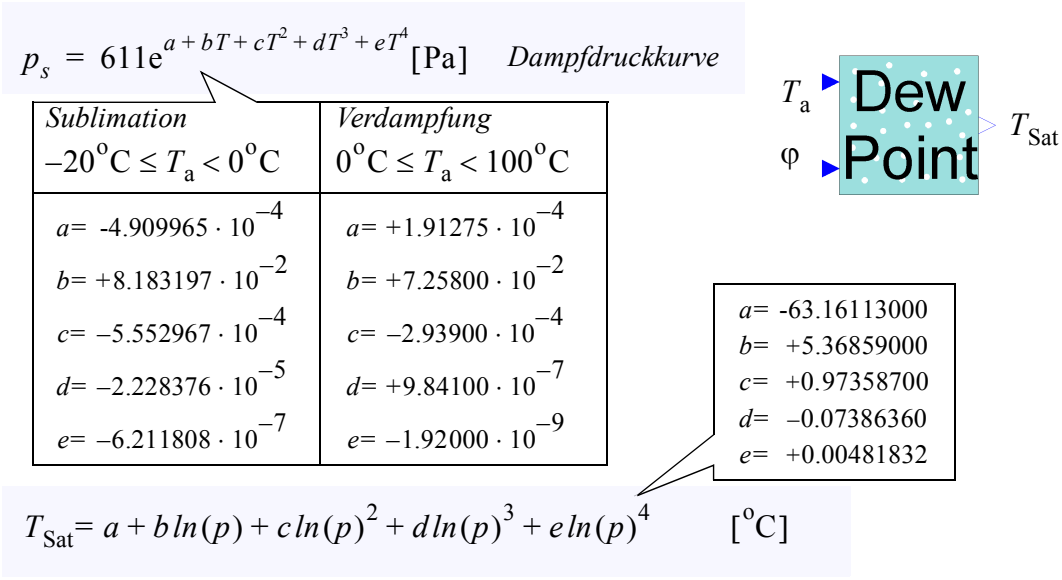


Abbildung 78: Komponente zur Berechnung der Sättigungstemperatur

Zunächst ist aufgrund der Umgebungstemperatur (T_a) zu entscheiden, ob die Verdampfungs- oder Sublimationskurve zur Berechnung des Sättigungsdampfdrucks (p_s) anzuwenden ist [Glü-91]. Aus dem Sättigungsdampfdruck (p_s) und der relativen Feuchte (φ) folgt der Wasserdampfpartialdruck nach $p = p_s \cdot \varphi$ [Loh-95]. Die Sättigungstemperatur (T_{Sat}) kann durch eine Potenzreihe in *ln(p)* beschrieben werden [Glü-91].

5.3.1.4 Fiktive Himmelstemperatur

Für eine Außenfläche des Gebäudes wird der freie Himmel jedoch nur unter einem Bruchteil des gesamten Raumwinkels $4\pi[sr]$ erscheinen. Daher ist die Himmelstemperatur durch den geometrischen Sichtfaktor (f_{sky}) zu korrigieren. Die resultierende fiktive Himmelstemperatur (T_{fsky}) kann unter Berücksichtigung des Emissions- bzw. Absorptionsgrades (ε_W) der Außenfläche (A_W) genutzt werden, um den langwelligen Strahlungsaustausch mittels eines Strahlungskopplers nach der Gleichung $j = \sigma A_W \varepsilon_W \cdot (T_W^4 - T_{fsky}^4)$ zu berechnen.

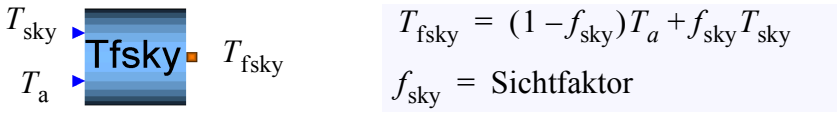


Abbildung 79: Komponente zur Berechnung der fiktiven Himmelstemperatur

5.3.2 Die solare Einstrahlung

Die zur Berechnung der solaren Einstrahlung benötigten Strahlungsdaten liegen meist in Form stündlicher Messwerte, getrennt nach direkter und diffuser Strahlung auf eine horizontale Fläche, vor.

Die Aufgabe des Srahlungsmodells ist es, zunächst aufgrund der wahren Ortszeit die aktuelle Sonnenposition zu berechnen. Dann ist durch trigonometrische Projektionen die Einstrahlung auf eine vorab festgelegte Anzahl vorab definierter Flächenorientierungen zu bestimmen.

Bei diesen Berechnungen sind die stündlichen Daten geeignet zu interpolieren. Die Bestimmungsgleichungen zur Berechnung des aktuellen Sonnenstandes sind etabliert [DuBe-74] und auch in der VDI Richtlinie [VDI-6020] aufgeführt. Die Modelle wurden kompatibel zu Standard Wetterdatenfiles entwickelt, wie sie auch TRNSYS benutzt. Später sollen auch eigene Messdaten einer Wetterstation [Mer-01] genutzt werden können. Die Darstellung und Implementierung folgt der Richtlinie VDI-6020.

5.3.2.1 Berechnung der wahren Ortszeit (WOZ)

Die aktuellen Sonnenposition kann nur aus der wahren Ortszeit (WOZ) berechnet werden. Zunächst ist zu berücksichtigen, dass innerhalb einer Zeitzone die lokale mittlere Ortszeit (MOZ) für einen geographischen Längengrad (L_{loc}) durch die Standardzeit der Zeitzone ersetzt wurde. Im Falle der mitteleuropäischen Zeit (MEZ) ist dies die mittlere Ortszeit des Standardlängengrades 15^{o} [DuBe-74]. Auch der Unterschied zwischen der Sommer- *(MESZ)* und Winterzeit *(MEZ)* ist zu berücksichtigen.

$$MEZ = MOZ + \frac{1}{15}(15^{o} - L_{loc}) \qquad MEZ = MESZ - 1$$

Durch die Excentrizität der Erdbahn kommt es zu einer Verschiebung zwischen der für die Berechnung der aktuellen Sonnenposition relevanten wahren Ortszeit *(WOZ)* und der mittleren Ortszeit *(MOZ)*. Die Differenz wird durch die Zeitgleichung *(Zgl)* beschrieben. Ihre Beschreibung ist auch aufgrund der Einflüsse der anderen Planeten sehr schwierig. Man findet in der Literatur verschiedene Näherungsansätze zur analytischen Beschreibung. Die VDI 6020 gibt die Zeitgleichung in Abhängigkeit von $J' = 360^{o} \cdot J/365$ an, wobei J der Tag des Jahres ist. Im Falle eines Schaltjahres muss $J' = 360^{o} \cdot J/366$ gewählt werden.

$$Zgl(J) = 0.0066 + 7.3525\cos(J' + 85.9^{o}) + 9.9359\cos(2J' + 108.9^{o}) + 0.3387\cos(3J' + 105.2^{o})$$

Der Stundenwinkel ϖ wird vom Meridian aus zum Nachmittag positiv und zum Vormittag negativ gezählt. Es gilt $\varpi = (12.00\text{h} - WOZ) \cdot 15^{o}/\text{h}$.

5.3.2.2 Berechnung der Deklination

Bei der Berechnung des aktuellen Sonnenstandes muss die Deklination δ berücksichtigt werden. Auch hier findet man verschiede analytische Näherungsgleichungen in der Literatur. Es wurde der Ansatz nach VDI 6020 ausgewählt.

$$\delta(J) = 0.3948 - 22.2559\cos(J' + 9.1^{o}) - 0.3915\cos(2J' + 5.4^{o}) - 0.1764\cos(3J' + 26.0^{o})$$

5.3.2.3 Berechnung des Zenit- und Azimutwinkels

Der Zenitwinkel wird von der Normalen aus gemessen. Der Azimutwinkel a_s wird von der Nordrichtung aus im Uhrzeigersinn positiv gezählt. Diese Konvention ist anders als im Falle des in TRNSYS als γ_s bezeichneten Azimutwinkels der zum lokalen Meridian gemessen wird. Die in Abb. 80 gezeigte Darstellung nutzt die Benennungen der VDI-6020.

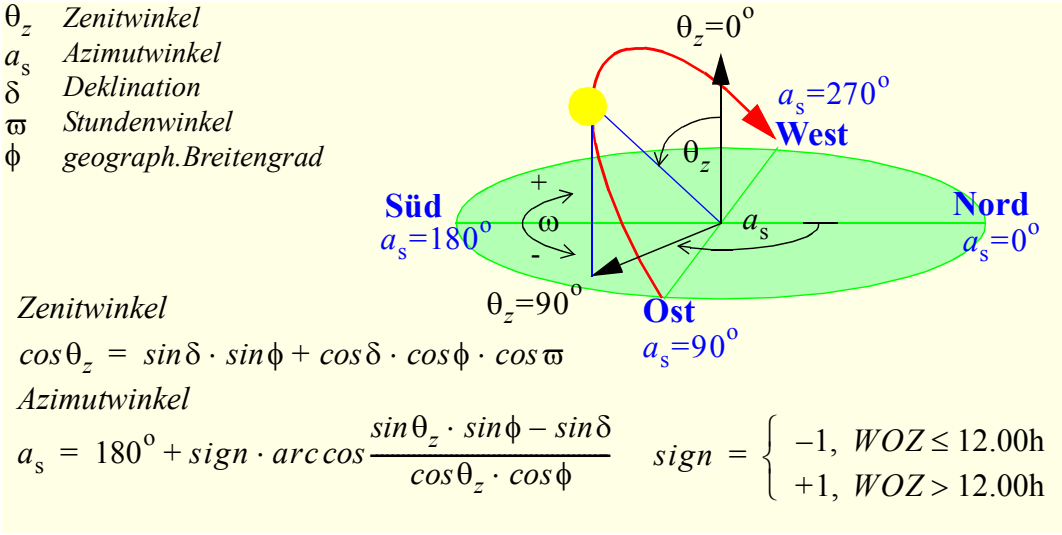


Abbildung 80: Berechnung der aktuellen Sonnenposition [VDI-6020]

5.3.2.4 Projektion auf beliebig geneigte Flächen

In typischen Wetterdatenfiles werden die Strahlungsdaten getrennt in direkte und diffuse Strahlung auf eine horizontale Fläche (I_b, I_d) in stündlichen Intervallen zur Verfügung gestellt.

Die direkte solare Strahlung (*Beam*-Strahlung) auf eine beliebig geneigte Fläche kann aus einer Projektion berechnet werden. Misst man die horizontale Flächenorientierung (γ) wie den Sonnenazimutwinkel (a_s) zur Nordrichtung, so ergibt sich nach VDI-6020 die in Tabelle 4 gezeigte Beziehung.

Problematisch sind aufgrund des $\tan\theta_z$-Terms Sonnenaufgang und Sonnenuntergang. Hier ist die *Beam*-Strahlung auf einen sinnvollen Einstrahlungswert längs der Flächennormalen zu begrenzen.

Schwieriger ist die Berechnung bei der von der Atmosphäre gestreuten Strahlung (diffuse Strahlung). Die Umrechnung der auf eine horizontal orientierte Fläche einfallenden diffusen Strahlung auf eine beliebig orientierte Fläche hängt im Gegensatz zur Winkelprojektion der *Beam*-Strahlung von der aktuellen Wetterlage ab, genauer vom Bedeckungsgrad *B*. Bei gleichmäßig bedecktem Himmel kann man von einer rotationssymmetrischen Verteilung der Strahlung um die Normale ausgehen.

Der Umrechnungsfaktor *R* ist eine Funktion der Neigung der betrachteten Fläche (β). Bei klarem Himmel ist zusätzlich die Differenz des Sonnenazimutwinkels (a_s) und der horizontalen Ausrichtung der Fläche (γ) von Bedeutung.

Beam-Anteil	$I_{bT} = I_b \cdot (cos\beta + cos\lvert a_s - \gamma \rvert sin\beta \cdot tan\theta_z)$
Diffusanteil, bei gleichmäßig bedecktem Himmel	$I_{dT} = I_d \cdot 0.182\left[1.178(1+cos\beta) + \left(\pi - \frac{\beta\pi}{180^o}\right) cos\beta + sin\beta\right]$
Diffusanteil, bei klarem Himmel	$I_{dT}^{klar} = I_d \cdot R(\theta_z, \beta, \lvert a_s - \gamma \rvert)$
Diffusanteil, bei teilweise bedecktem Himmel	$I_{dT} = (1\text{-}B) \cdot I_{dT}^{klar} + B \cdot I_{dT}^{bedeckt}$
Am Boden reflektierter Anteil	$I_{gT} = (I_d + I_b) \cdot 0.5\rho_U(1\text{-}cos\beta)$

Tabelle 4: Berechnung der Einstrahlung auf eine beliebig geneigte Fläche aufgeteilt in *Beam*-, Diffus- und am Boden reflektierten Anteil nach [VDI-6020]

Die Richtlinie VDI-6020 gibt die in Tabelle 4 gezeigten Umrechnungsvorschriften vor, wobei der Umrechnungsfaktor *R* einer weiteren Tabelle *(in VDI 2078)* zu entnehmen ist. Für den Reflexionsgrad des Bodens kann $\rho_U = 0.2$ angenommen werden. Im Rahmen unserer Modellbibliothek wurden verschiedene Beschreibungen implementiert, auch alle von TRNSYS genutzten Strahlungsmodelle (vgl. Abb. 41). *(Während die TRNSYS Modelle in ihrer Darstellung des Diffusanteils von der VDI-6020 abweichen, stimmen sie im Falle des Beam- und reflektierten Anteils mit ihr überein (vgl. Abb. 41).)*

5.3.2.5 Interpolation der Strahlungsdaten

Die Strahlungsdaten des Wetterdatenfiles sind in einem stündlichen Raster $(\Delta T = 1\text{h})$ in Form stündlicher Mittelwerte $\overline{I_{d,b}}$ der solaren Einstrahlung auf eine horizontale Fläche in [W/m^2] aufgezeichnet. Dabei entspricht der Wert zum Tabelleneintrag T_i dem Integral der Strahlung über das zurückliegende Zeitintervall $(\Delta T = 1\text{h})$ dividiert durch die Länge des Zeitintervalles (Abb. 81).

$$\overline{I_{d,b}}(t) = \frac{1}{\Delta T}\int_{T_{i-1}}^{T_i} I_{d,b}(t)dt \qquad \text{mit } T_{i-1} < t \le T_i$$
$$\Delta T = T_i - T_{i-1} = 1\text{h}$$
$$i = 1,2,3,...$$

Abbildung 81: Strahlungsdaten *(diffus, und Beam)* in Form stündlicher Mittelwerte, wie sie einem Wetterdatenfile entnommen werden können.

Es zeigte sich, dass eine lineare Interpolation der Strahlungsdaten, insbesondere im Bereich des Sonnenaufganges und Sonnenunterganges, ungeeignet zur Berechnung von $I(t)$ ist.

Daher wird die Interpolation unter Berücksichtigung des theoretischen zeitlichen Verlaufes der solaren Einstrahlung durchgeführt. Dieser berechnet sich aus der Solarkonstanten und einer Winkelprojektion des gerichteten Strahlungsvektors auf die Normale.

Die Solarkonstante $I_{sol} = 1352\ \mathrm{W/m^2}$ gibt an, welche Strahlungsenergie pro 1s auf eine senkrecht zur Ausbreitungsrichtung ausgerichtete Fläche von $1\mathrm{m}^2$ ohne Berücksichtigung der Erdatmosphäre fällt. Hierbei wird angenommen, dass der Abstand der Fläche zur Sonne dem mittleren Erdbahnradius entspricht [DuBe-74]. Den theoretisch auf eine horizontal orientierte Fläche am Erdboden gelangenden Anteil erhält man durch die Projektion des gerichteten Strahlungsvektors auf die Normalenrichtung als $I_{ex}(t) = I_{sol} cos\theta_z(t)$. Der Mittelwert des theoretischen Verlaufes $(\overline{I_{ex}}(t))$ im Intervall $[T_{i-1}, T_i]$ wird durch die Integration über dieses Zeitintervall und Division durch seine Zeitdauer berechnet (Abb. 82).

$$\overline{I_{ex}}(t) = \frac{1}{\Delta T}\int_{T_{i-1}}^{T_i} I_{ex}(t)dt \qquad \text{mit } T_{i-1} < t \le T_i,\ \Delta T = T_i - T_{i-1} = 1\mathrm{h},\ i = 1,2,3,...$$

Abbildung 82: Berechnung stündlicher Mittelwerte der extraterrestrischen Strahlung

Als Näherung für die einfallende Strahlung $I_b(t)$ bzw. $I_d(t)$ kann man den Quotienten aus dem theoretischen Strahlungswert $I_{ex}(t)$ zum betrachteten Zeitpunkt durch dessen Mittelwert $\overline{I_{ex}}(t)$ im betrachteten Tabellenintervall multipliziert mit dem Tabellenwert $\overline{I_{d,b}}(t)$ betrachten. Die Abschwächung der extraterrestrischen Strahlung beim Durchqueren der Atmosphäre kann in erster Näherung über das stündliche Intervall als konstant angenommen werden und entfällt, da sie gleichermaßen im Zähler und Nenner als Produkt vorkommt. Gerade im Bereich des Sonnenaufganges und Sonnenunterganges ist diese Näherung problematisch, da das horizontal einfallende Sonnenlicht häufig niedrig liegende Dunstschichten durchqueren muss (Abb. 83).

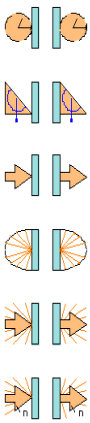
$$I_{d,b}(t) \approx I_{ex}(t) \cdot \frac{\overline{I_{d,b}}(t)}{\overline{I_{ex}}(t)}$$

Abbildung 83: Interpolation der *Beam*- und Diffusstrahlung unter Berücksichtigung des Verlaufes der extraterrestrischen Strahlung.

Neben der extraterrestrischen Strahlung zum Zeitpunkt t wird zur Berechnung des Mittelwertes das Integral der extraterrestrischen Strahlung im aktuellen Zeitintervall ΔT benötigt. Die Berechnung des Integrals wird in einer eigenen Modellkomponente durchgeführt (siehe Abb. 86.: *Modellkomponente Integration*).

5.3.2.6 Implementierung

Alle Strahlungsmodelle besitzen einheitliche, gerichtete Schnittstellen. Alle Schnittstellenvariablen werden als Potentialvariablen übergeben. Da es bei den Wärmeströmen um Bestrahlungsstärken in $[\mathrm{W/m^2}]$ handelt, werden auch die Wärmeströme als Potentialvariablen übergeben.



Zeit *(Tageszeit, Wochentag, Tag des Monats, Monat, Jahr)*

Winkel ($sin\,\theta_z$, $cos\,\theta_z$, $sin\,\gamma_s$, $cos\,\gamma_s$)

Beam-Strahlung in [W/m^2] (I_b)

diffuse Strahlung [W/m^2] (I_d)

totale Strahlung [W/m^2] (I_t)

Strahlungsvektor, totale Strahlung [W/m^2] auf Fläche *i* (I_{t1}, ... , I_{ti} , ... , I_{tn})

Abbildung 84: Schnittstellen der Strahlungsmodelle

Die Bibliothek (Abb. 85) umfasst Komponenten aus denen für den konkreten Anwendungsfall ein geeignetes Solarmodell erstellt werden kann. Die Funktionsweise wird später an einem Beispiel (Abb. 86) veranschaulicht.

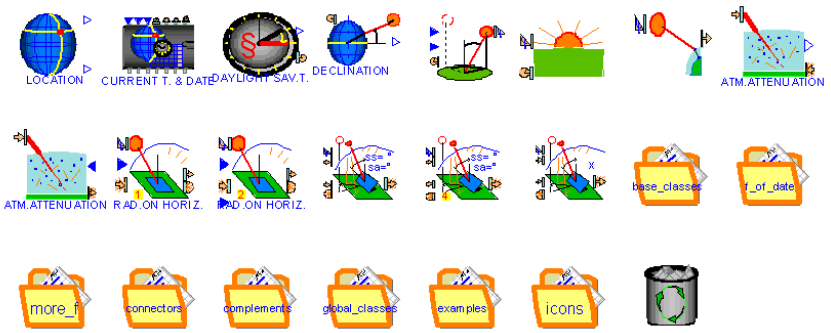


Abbildung 85: Solarmodelle als Teil der Umgebungs-Bibliothek

5.3.2.7 Ein Beispiel eines vollständigen Umgebungsmodells

Das in Abb. 86 gezeigte vollständige Umgebungsmodell hat die Aufgabe, ausgehend von einem Wetterdatenfile im Standard-TRNSYS-Format, die für die Simulation des thermischen Verhaltens eines Gebäudes notwendigen Daten geeignet aufbereitet und interpoliert zur Verfügung zu stellen.

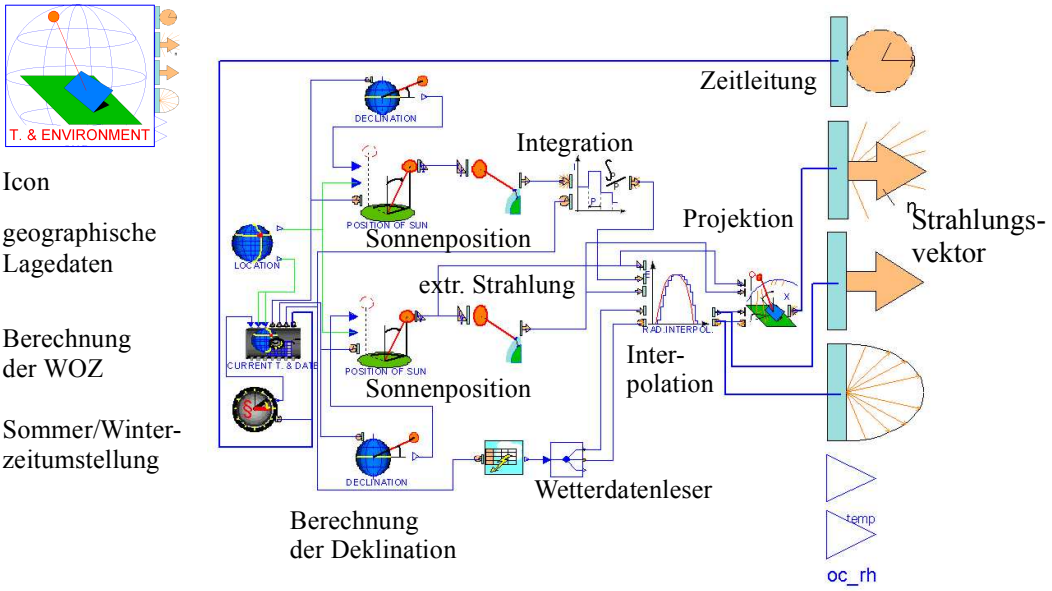


Abbildung 86: Ein Beispiel eines vollständigen Umgebungsmodells

Aufgrund der geographischen Lagedaten des Gebäudestandortes wird die wahre Ortszeit berechnet. Hierbei ist eine eventuelle Umstellung auf Sommerzeit zu berücksichtigen. Die Lagedaten verschiedener Städte sind in einer Tabelle gespeichert und können ausgewählt werden. Das initiale Wetterdatenfile muss die gerichtete *Beam*- und diffuse Strahlung auf eine horizontale Fläche, die Umgebungstemperatur und die relative Feuchte in Form von stündlichen Mittelwerten enthalten.

Das Modell soll zum einen die stündlichen Mittelwerte des Wetterdatenfiles geeignet interpolieren und ausgeben. Zum anderen soll es aufgrund des Sonnenstandes die totale Bestrahlungsstärke vorab definierter möglicher Fassadenorientierungen berechnen und zu einem Vektor zusammenfassen. Ein Fenster muss dann nur seine Zugehörigkeit zu einer Fassade kennen und kann dann durch Auswahl der korrekten Vektorkomponente seine aktuelle Bestrahlungsstärke bestimmen.

Das Modell Abb. 86 verfügt über zwei Zweige mit jeweils einem Element zur Sonnenpositionsberechnung, zur Berechnung der Deklination und zur Berechnung der extraterrestrischen Strahlung. Dies ist notwendig zur Berechnung des Mittelwertes der theoretischen Bestrahlungsstärke über das aktuelle Zeitintervall. Beide Zweige unterscheiden sich in der Zeit. Der Strahlungsprozessor gibt auf einen Zweig die aktuelle Simulationszeit, auf den anderen die um 1 *Stunde* in der Zukunft liegende Zeit. Im Integrator wird die Integration über das Zeitintervall spezifiziert, wobei ein algorithmischer Anteil die Integrationsgrenzen mit den Tabellenzeiten synchronisiert. Im Interpolationselement werden die Ergebnisse der zwei Zweige zusammengeführt und die Interpolation für den diffusen Anteil und den *Beam*-Anteil durchgeführt.

Die Projektion erfolgt entsprechend dem genutzten Strahlungsmodell (Tabelle 4 auf Seite 94) auf die vorab ausgewählten Vorzugsorientierungen innerhalb des Projektionselementes. Die Strahlungsdaten für diese möglichen Orientierungen werden zu einem Vektor zusammengefasst und auf die Ausgangsschnittstellen gegeben.

Neben den Strahlungsdaten stellt das Modell auch die aus den Wetterdaten gelesenen Daten für Außentemperatur und Feuchte zur Verfügung. Hierbei genügt eine einfache lineare Interpolation.

Die in Tag, Woche, Jahr aufbereitete Zeit wird auf eine Zeitleitung gegeben und nach außen geführt, um sie in anderen Modellkomponenten, die nach einem festen Zeit- oder Kalenderschema vorgehen, nutzen zu können. So ist ein täglich, wöchentlich oder an bestimmte Kalendertage gebundener wiederkehrender Ablauf einfach zu spezifizieren.

Die *Beam*- und diffuse Strahlung auf eine horizontale Fläche wird auch nach außen geführt und steht so u.a. zur Berechnung der Himmelstemperatur zur Verfügung.

5.4 Die Algorithmen-Bibliothek

5.4.1 Die Reglerbausteine

Die hier implementierten Regler sollen es ermöglichen auch geregelte technische Prozesse mathematisch beschreiben zu können. Es handelt sich daher vorwiegend um parametrierbare, komponierbare Grundbausteine ([Föl-93], [Föl-94], [Föl-00], [Lit-01a], [Lun-96], [Lun-97], [MeLi-00b]).

5.4.1.1 Kontinuierlicher PID-Regler

In vielen Anwendungsfällen genügt der Einsatz eines *PID*-Reglers. In Abb. 87 ist der Zusammenhang zwischen der Stellgröße u(t) als Ausgang des Reglers und der Regelabweichung e(t) als Eingangsgröße im Laplace-Bereich durch den proportional wirkenden *P*-Anteil, den integrierend wirkenden *I*-Anteil und den differenzierend wirkenden *D*-Anteil dargestellt.

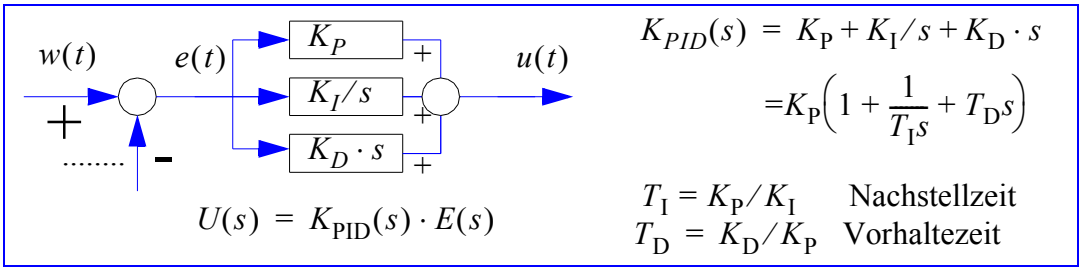


Abbildung 87: Kontinuierlicher PID Regler in Parallelstruktur
[Föl-94], [Lun-96], [Lit-01a], [MeLi-00b]

Der *P*-Anteil bewirkt eine proportionale Zunahme der Stellgröße u(t) mit der Regelabweichung e(t). Der *I*-Anteil erzielt die stationäre Genauigkeit. Die Stellgröße u(t) wird solange verändert, wie die Regelabweichung $e(t) \neq 0$ ist. Das System ist so erst bei $e(t) = 0$ vollständig eingeschwungen und stationär. Der *D*-Anteil reagiert sofort auf Veränderungen der Regelabweichung und verhindert damit schon das Entstehen großer Absolutwerte der Regelabweichung e(t).

5.4.1.2 Diskreter PI -Regler

Kontinuierliche Algorithmen sind wenig geeignet für softwaretechnische Realisierungen. Die periodische Abtastung der Istwerte legt die Nutzung eines rekursiven Algorithmus nahe. In Abb. 88 wird unter Nutzung der z-Transformation aus einer kontinuierlichen Beschreibung im Laplace-Bereich ein rekursiver *PI*-Algorithmus abgeleitet. Auf den *D*-Anteil wurde im Beispiel (Abb. 88) verzichtet, da insbesondere bei den in der Anwendungsdomäne Gebäudeautomation auftretenden langsamen thermischen Prozessen meist *PI*-Regler zur Anwendung kommen.

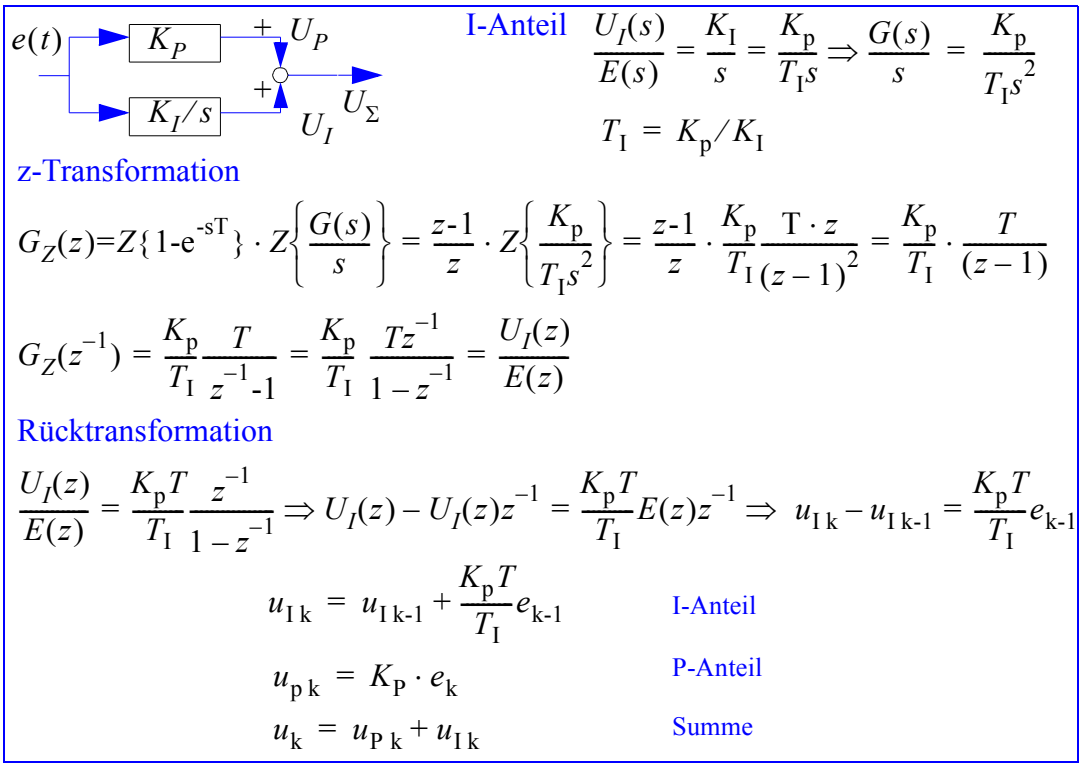


Abbildung 88: Rekursiver, diskreter PI-Algorithmus [Föl-00], [Lit-01a], [Lun-97], [MeLi-00b]

5.4.2 Die Erweiterungen der klassischen Reglerbausteine

5.4.2.1 Glättungsglied

In praktischen Anwendungsfällen ist ein sogenannter idealer *PID*-Regler, wie bislang beschrieben, nicht einsetzbar. Die Hauptproblematik ist das Auftreten hochfrequenter Störungen *d(t)* bei realen Strecken. Beim realen PID-Regler (Abb. 89) filtert daher ein zusätzliches Glättungsglied die hochfrequente Störung im D-Zweig aus ([Lun-96], [Lit-01a], [MeLi-00b]).

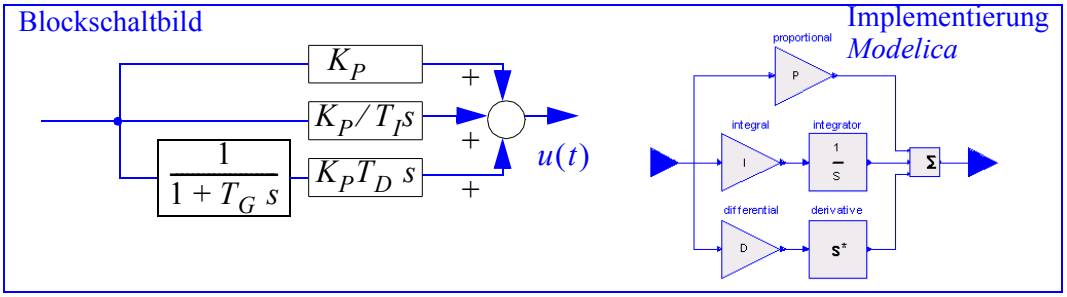


Abbildung 89: Blockschaltbild und Implementierung eines realer PID-Reglers mit Glättungsglied und Glättungszeitkonstante T_G im D-Anteil ($T_D \gg T_G$) [Lun-96], [Lit-01a], [MeLi-00b], [Mod-02].

Eine Implementierung des realen *PID*-Reglers in *Modelica* findet sich in Abb. 89. Die Darstellung zeigt die blockschaltbildartige Summendarstellung des PID-Reglers mit drei Summanden (*P, I,* geglätteter *D*-Anteil). Man wählt die Glättungszeitkonstante *(T_G)* wesentlich kleiner als die Vorhaltezeit *(T_D)* ($T_G \ll T_D$). Zur Veranschaulichung der Wirkungsweise wird in Abb. 90 ein Einheitssprung auf den Regler gegeben und die Ausgangsgröße bei unterschiedlicher Parametrierung beobachtet.

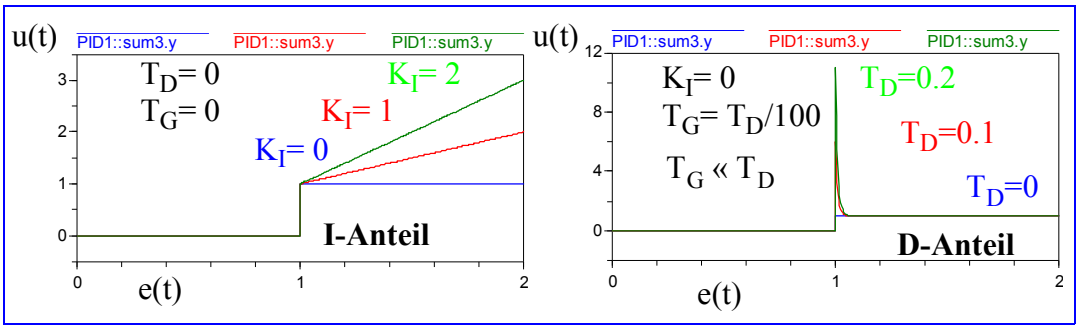


Abbildung 90: Veranschaulichung der I-, D- Anteile

5.4.2.2 Nichtlineare Erweiterungen

Um eventuelles Messrauschen des Regelfehlers nicht auf die Stellgröße zu übertragen und die Aktorik damit zu belasten, wird in praktischen Anwendungen ein Totzonen-Element [Föl-94] (Abb. 91) eingesetzt. Kleine Fluktuationen der Regeldifferenz führen so zu keiner Änderung der Stellgröße, da diese innerhalb eines Intervalles kleiner Regeldifferenzen konstant 0 bleibt.

Auch muss die vom Regler erzeugte Stellgröße aus technischen Gründen stets beschränkt sein, denn die angesteuerte Aktorik kann nur einen beschränkten Wertebereich verarbeiten. Diese Eigenschaft simuliert das Begrenzerelement (Abb. 91).

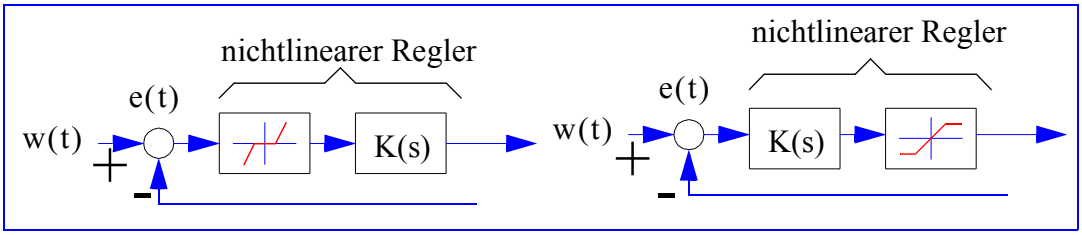


Abbildung 91: Totzonen- und Begrenzerelement als nicht lineare Bausteine der Reglerbibliothek [Föl-94], [Lit-01a], [MeLi-00b]

5.4.2.3 Anti-wind-up-Maßnahme

Das Zusammenwirken eines Reglers mit *I*-Anteil und eines Begrenzers führt zur sogenannten *Wind-up*-Problematik (Abb. 93). Der *I*-Anteil ($|u_I|$, $|u_\Sigma|$) wächst bei $e(t) > 0$ oder $e(t) < 0$ betragsmäßig grenzenlos (*Wind-up*).

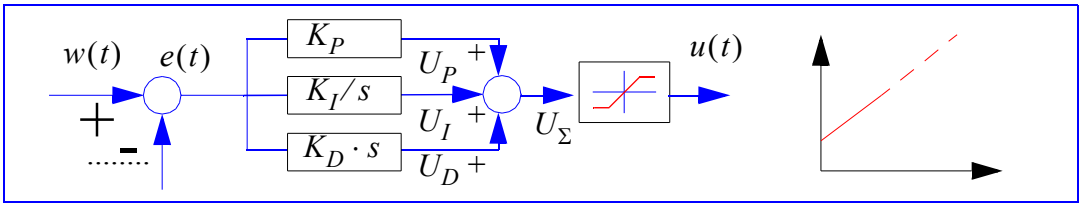


Abbildung 92: Wind-up-Problematik des *I*-Anteils [Lun-96], [Lit-01a], [MeLi-00b]

In Abb. 93 ist die Problematik anhand eines einfachen Streckenmodelles veranschaulicht. Als Aktor steht eine Wärmestromquelle zur Verfügung. Ein kontinuierlicher PID Regler hat die Aufgabe, die Raumtemperatur auf einen Sollwert einzuregeln. Ohne Anti-wind-up-Maßnahme kommt es bei Sollwertänderung zu drastischen Überschreitungen des neuen Sollwertes. Der während der Aufheizphase bei $e(t) > 0$ stetig zunehmende Integral-Anteil muss nach Erreichen des Sollwertes erst abgebaut werden.

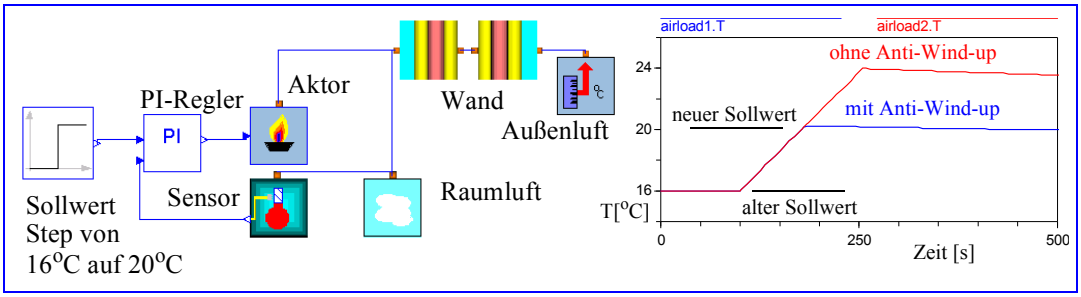


Abbildung 93: Beispiel einer typischen Regelstrecke (Beheizung eines Einraumhauses mit PI-Raumthermostat)

In Abb. 94 wird eine Anti-wind-up-Maßnahme beim kontinuierlichen, in *Modelica* implementierten PI-Regler gezeigt. Erkennt der Algorithmus, dass die Stellgröße ihren maximalen

(oder auch minimalen) Wert erreicht hat, wird der Integralanteil abgeschaltet. Dies verhindert ein weiteres Anwachsen des Integralanteils.

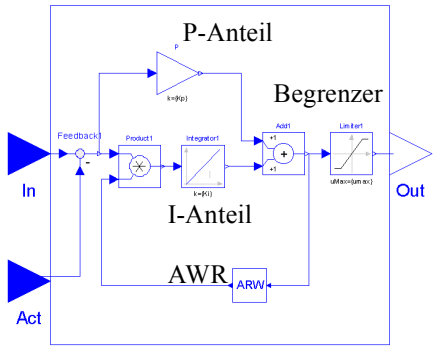


Abbildung 94: Kontinuierlicher PI-Regler mit AWR (Anti-Wind-up-Reset)

Abb. 95 zeigt in Form eines Flussdiagramms eine Anti-wind-up-Maßnahme für den diskreten PI-Regler mit rekursivem Algorithmus aus Abb. 88. Die Größe e_{k-1} stellt die Ursache der wind-up-Problematik dar.

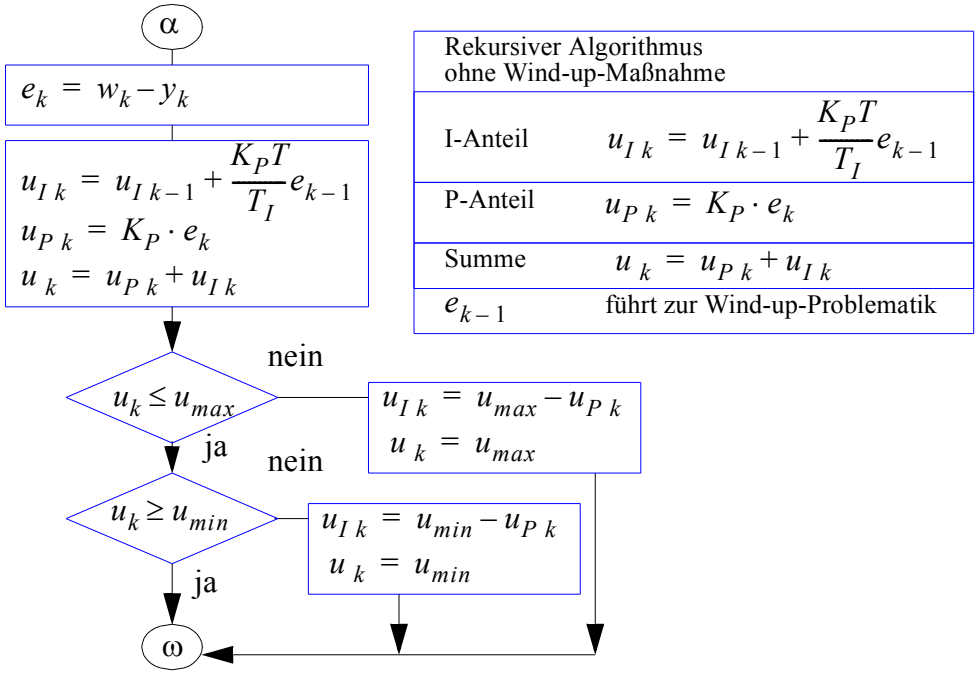


Abbildung 95: Dynamischer Anti-wind-up-Reset [Föl-93], [Lit-01a], [MeLi-00b]

5.4.2.4 Betriebsartensteuerung

Neben der Realisierung der funktionalen Eigenschaften erfordern die Reglerspezifikationen auch eine Behandlung sogenannter nicht-funktionaler Eigenschaften. So muss es möglich sein, einen Regler außer Betrieb zu nehmen, in Betrieb zu nehmen und zu parametrieren. Das Umschalten sollte u.a. zur Schonung der Aktorik stoßfrei geschehen. Abb. 96 zeigt einen Lösungsansatz für den rekursiven Algorithmus aus Abb. 88.

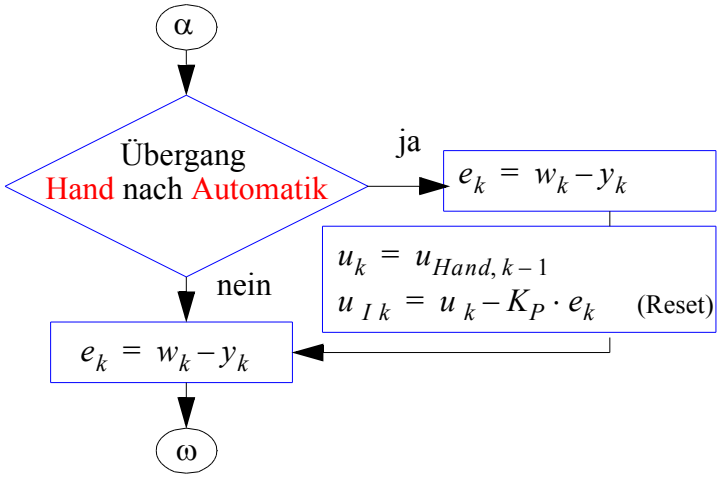


Abbildung 96: Reset des Integrators zur Erzielung von Stoßfreiheit [Lit-01a], [MeLi-00b]

5.4.3 Ein Beispiel eines vollständigen Reglers

Bei dem in Abb. 98 gezeigten Regler handelt es sich um einen diskreten, realen PID-Regler mit Abtastsimulation, dynamischem Wind-up-Reset und Betriebsartenumschaltung blockschaltbildartig implementiert in *Modelica*.

Der aktuelle, kontinuierliche Istwert wird im Element *A/D-Wandler* zu festen Zeiten t_i abgetastet und entsprechend der eingestellten Geräteauflösung diskretisiert. Der *A/D-Wandler* unterteilt das maximale Inputintervall in 2^{bits} Intervalle. Erst nach Überschreiten der nächsten Intervallgrenze wird ein neuer Wert angenommen. Die Abfrage erfolgt zudem nur zur Abtastzeit, dazwischen werden die Werte gehalten.

Zur Implementierung wird die *Modelica sample( )*-Funktion [Til-01] genutzt.

Als Argument einer *when*-Anweisung der Form:

„when Sample(SampleOffset, SampleInterval) then u_{out} *= .... endwhen“*

ändert sich der u_{out}-Wert ausschließlich zu den eingestellten Abtastzeiten. Dies sind hier ganzzahlige Vielfache von *SampleInterval*, beginnend zum Zeitpunkt *SampleOffset*.

Abbildung 97: Implementierung einer Abtastung in *Modelica*.

Die Regeldifferenz wird zwischen dem abgetasteten, diskretisierten Istwert und der abgetasteten, diskretisierten Führungsgröße berechnet. Sie wird nach Beruhigung durch ein Totzonenelement auf den diskreten *PID*-Algorithmus gegeben.

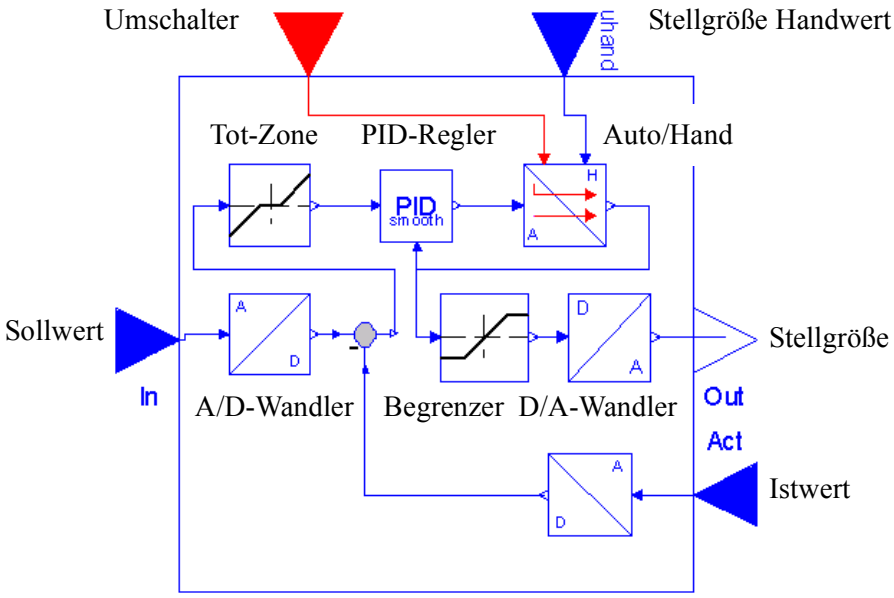


Abbildung 98: Modularer *PID*-Regler mit Abtastsimulation, dynamischem Wind-up-Reset sowie Betriebsartenumschaltung in *Modelica*

Diesem folgt der Betriebsartenumschalter. Der momentane Zustand *("Auto oder Hand")* wird über die Signalleitung *(rot in Abb. 98)* eingestellt. Im Handmodus wird die am *Handwert-Eingang* u_{hand} eingehende Stellgröße direkt ausgegeben. Im Automatikmodus wird die Stellgröße durch den *PID*-Regler bestimmt.

Steht der Regler im Handmodus, muss der Integrator des *PID*-Reglers rückgesetzt werden, um eine Wind-up-Problematik zu vermeiden und stossfreies Umschalten zu gewährleisten. Eine Rückführung zwischen Betriebsartenumschalter und PID-Regler ist daher erforderlich. Die erzeugte Stellgröße wird über einen *DA*-Wandler sowie ein Begrenzungselement auf den Ausgang gegeben.

5.5 Ein Beispiel eines vollständigen Gebäudemodells

In Abb. 99 ist die exemplarische Verschaltung eines fensterlosen 11 Raumhaus mit thermohydraulischer Heizungsanlage abgebildet. Das Haus verfügt über einen gemeinsamen Flur und 5 Zimmer auf jeder Seite des Flurs. Unter jedem Raum ist ein unbeheizter Keller. Die Außenwände werden zu einer Südfassade, Nordfassade und Dachfläche zusammengefasst. Die Außentemperaturen der verschiedenen Gebäudeseiten werden im Tagesverlauf sinusförmig mit einer Periode von *24h* angenommen.

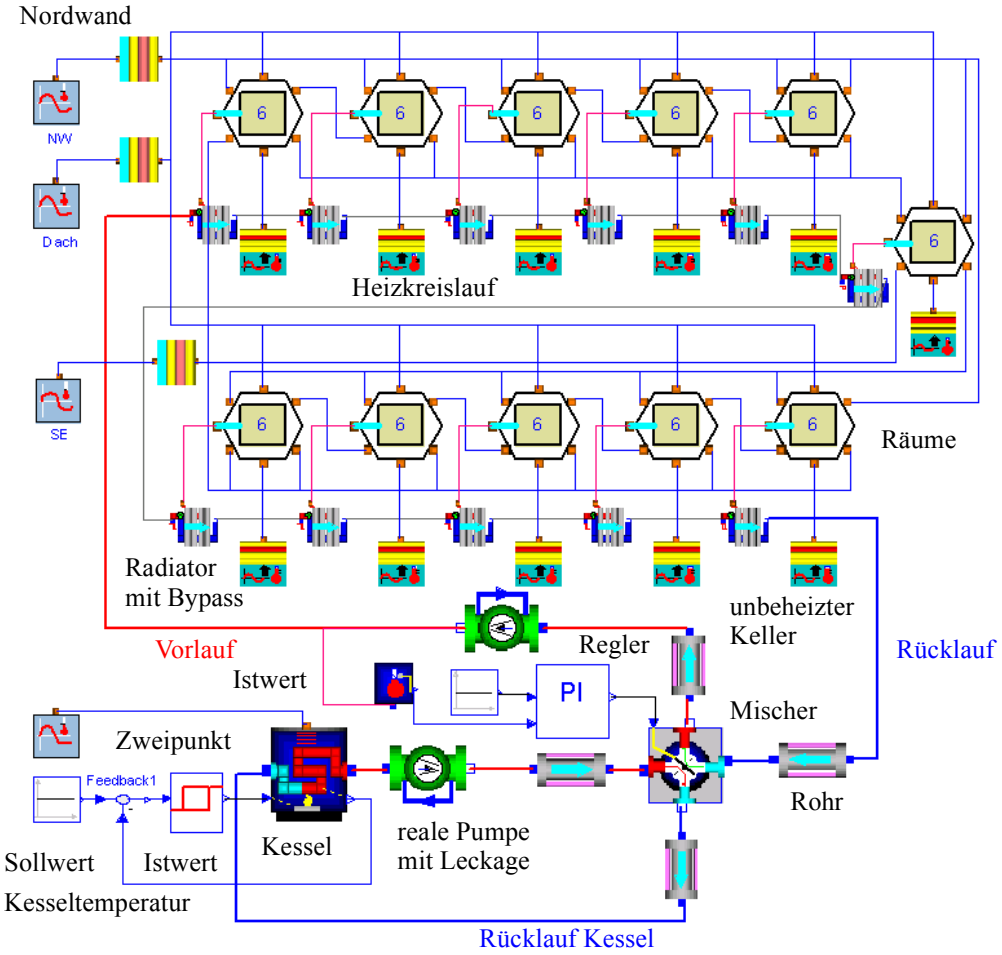


Abbildung 99: Beispiel eines 11-Raum-Hauses mit vollständiger Warmwasserheizung.

Der Heizkreis ist durch einen Mischer in zwei Stufen untergliedert. In jedem der Kreise befindet sich eine Pumpe. Jeder Raum wird mit einem Radiatorheizkörper beheizt. Die Radiatoren sind in Reihe geschaltet und verfügen über Raumthermostate. Es werden auf Raumebene Zweipunktregler mit Hysterese eingesetzt, als Aktoren dienen Dreiwegeventile in Bypassschaltung. Die Vorlauftemperatur der Heizkreise wird im gezeigten Beispiel über einen *PI*-Regler mit Anti-Windup-Reset eingestellt. Als Aktor dient das Mischerventil. Die Heizkesseltemperatur wird über einen Zweipunktregler geregelt, der über ein binäres Steuersignal den Kesselbrenner einschaltet.

6 Die Validierung der Modellbibliothek

6.1 Eine Simulation topologisch verschieden strukturierter Modelle

Im folgenden wird am Beispiel des Einraumhauses aus Abschnitt 3.2 ein Vergleich der drei unterschiedlichen Methoden zur topologischen Strukturierung mathematischer Modelle durchgeführt. Die Signalfluss- (3.2.2) und Energieflussverknüpfung (3.2.3) können aufgrund der gerichteten Verbindungen mit *Matlab/Simulink* implementiert werden. Die phänomenologische Verknüpfung (3.2.4) erfordert jedoch eine nicht berechnungskausale Simulationsumgebung wie *Modelica/Dymola*.

Die unterschiedlichen Implementierungen werden verglichen und die Simulationsergebnisse werden einander gegenübergestellt (Abb. 100). Im Beispiel soll die resultierende Raumlufttemperatur aufgrund der zeitlich variierenden Außenlufttemperatur berechnet werden. Das *Simulink*-Modell besteht aus drei Subsystemen. An alle Subsysteme werden die gleichen Eingangsgrößen angelegt.

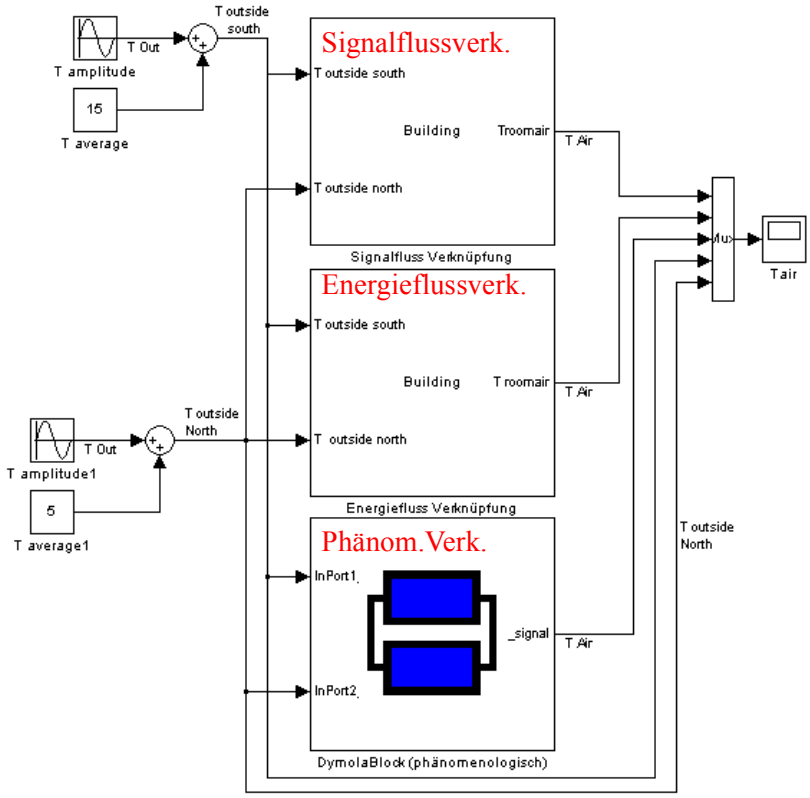


Abbildung 100: *Simulink*-Modell bestehend aus drei Subsystemen. Die Subsysteme sind nach den drei unterschiedlichen topologischen Strukturierungsmethoden implementiert (Signalfluss-, Energiefluss- und phänomenologische Verknüpfung).

Das *Modelica*-Modell ist als *Dymola-Block* in die *Simulink*-Simulationsumgebung eingebunden. So können die Simulationsergebnisse direkt unter *Simulink* analysiert werden. Hierbei ist die *Matlab/Simulink* imanente Kausalität dem *Modelica*-Modell auf höchster Hierarchieebene durch die Definition äußerer Ein-/Ausgabeschnittstellen aufgeprägt. Innerhalb des *Modelica*-Modells, also auf allen niedrigeren Hierarchieebenen, verbleibt die Modellierung nicht berechnungskausal. Vor der eigentlichen Simulation wird *Dymola* als Preprozessor aufgerufen. *Dymola* erstellt dann nach Auffinden der Kausalität aus dem nicht berechnungskausalen Modell den in *Simulink* einzubindenden Simulationscode.

Das Haus wird wieder in Form konzentrierter Speicher abgebildet. Als Vorgabe wird angenommen, dass die Außenlufttemperaturen auf der Nord- und Südseite aufgrund solarer Einstrahlung am Tag große Temperaturdifferenzen aufweisen und sinusförmig mit einer Periode von 24 h variieren. Dies bildet den täglichen Wechsel des Außenlufttemperaturverlaufes nach. Wärmeaustauschprozesse innerhalb des Hauses erfolgen rein konvektiv unter Vernachlässigung der langwelligen Wärmestrahlung.

Die Systemordnung ergibt sich aufgrund der Speicher *(Nordwand, Raumluft, Südwand)* zu 3.

6.1.1 Die Implementierung der Signalflussverknüpfung

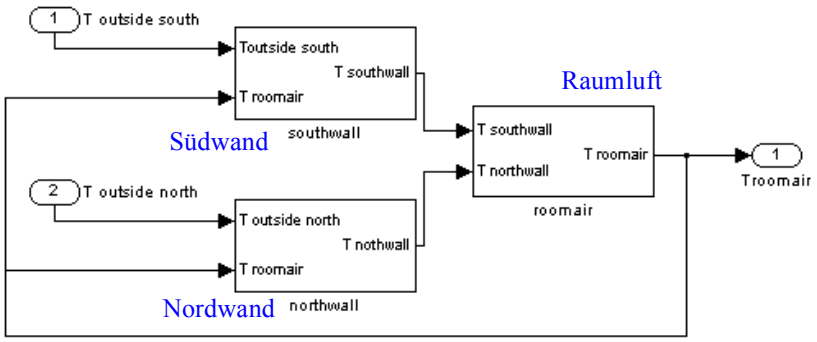


Abbildung 101: Signalflussverknüpfung: Gesamtsystem

Abb. 101 zeigt die blockschaltbildartige Implementierung unter Anwendung der Signalflussverknüpfung (siehe 3.2.2) in *Simulink*. Das Hausmodell besitzt auf oberster Hierarchieebene drei Blöcke (*Nordwand, Südwand, Raumluft*). Jeder der Blöcke erhält als Eingangsgröße die Kerntemperaturen der mit ihm in Verbindung stehenden Blöcke. Seine Ausgangsgröße ist die eigene Kerntemperatur.

In *Simulink* bietet es sich an, auch die Komponenten selbst graphisch zu implementieren (Abb. 102). So aggregiert jeder Block den physikalischen Speicher und zwei Elemente zur Berechnung der Wärmeströme zu seinen beiden Nachbarn. Aufgrund der Strukturierung muss jedes Wärmeübergangselement zweimal vorhanden sein: innerhalb des Blockes, aus dem ein Wärmestrom ausfließt, und innerhalb des Blockes, in den der Wärmestrom einfließt. So genügt es, an den Schnittstellen nur die Kerntemperaturen zu übergeben. Jeder Wärmestrom ist daher zweimal zu berechnen: zunächst mit positivem Vorzeichen im empfangenden Block, dann mit negativem Vorzeichen im abgebenden Block. Die Vorzeichenkonvention kann entsprechend einer beliebigen, aber festen Zählpfeilkonvention erfolgen.

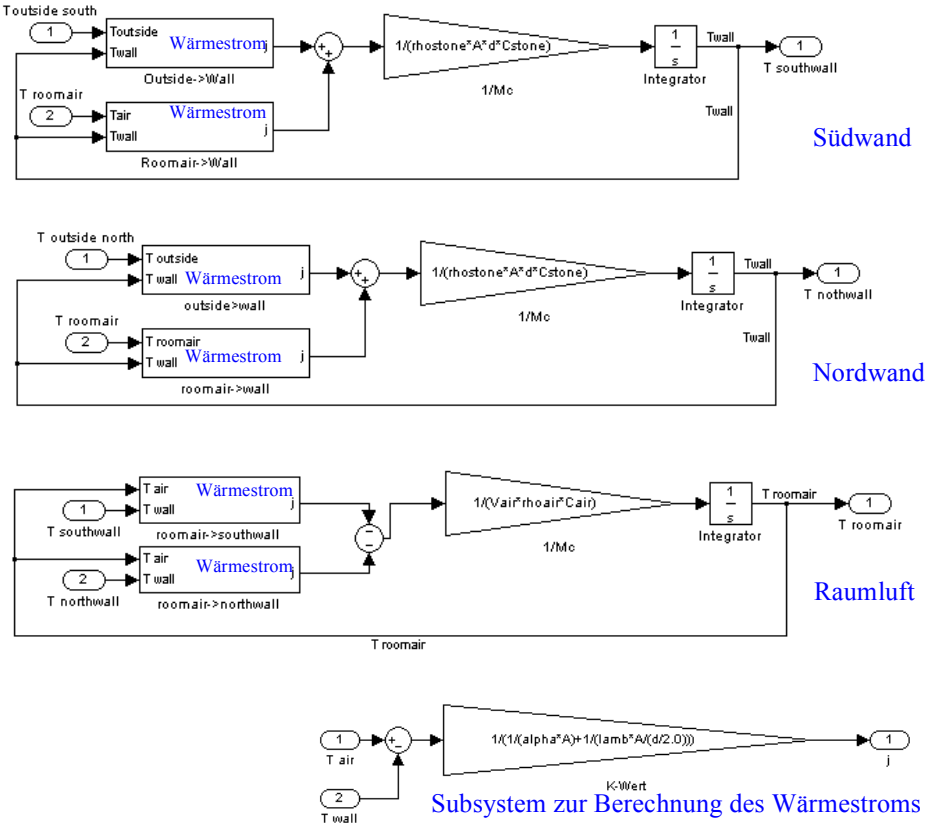


Abbildung 102: Signalflussverknüpfung: Komponentenmodelle

6.1.2 Die Implementierung der Energieflussverknüpfung

Bei der Energieflussverknüpfung (siehe 3.2.3) benötigt man drei Komponentenbausteine und vier Koppelbausteine. In den Koppelbausteinen werden die zwischen zwei Komponenten fließenden Wärmeströme berechnet. Hierzu müssen die Temperaturen der beiden Komponenten auf die Koppelbausteine (Abb. 103) rückgeführt werden. Die Ausgangsgrößen werden dann an den Eingängen der Komponentenbausteine vorzeichenrichtig addiert. Die Eingangsgrößen werden über spezielle Bausteine eingekoppelt.

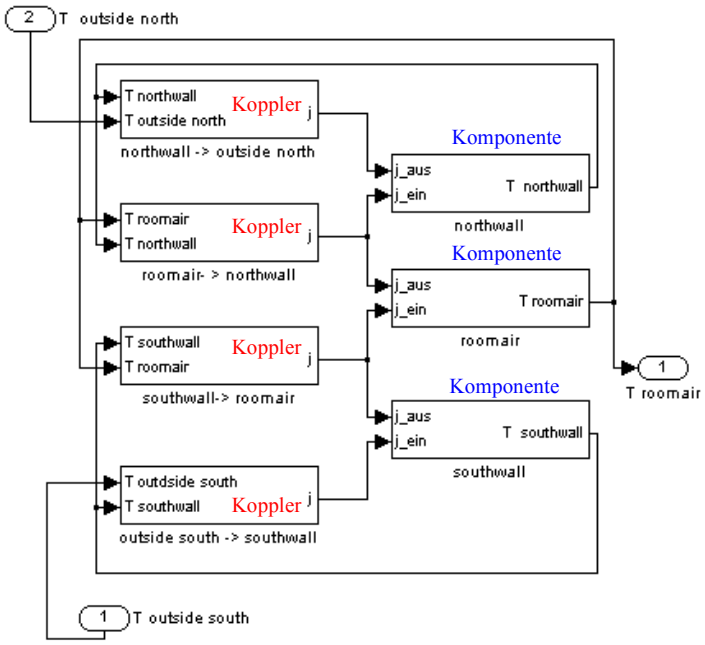


Abbildung 103: Energieflussverknüpfung: Gesamtsystem

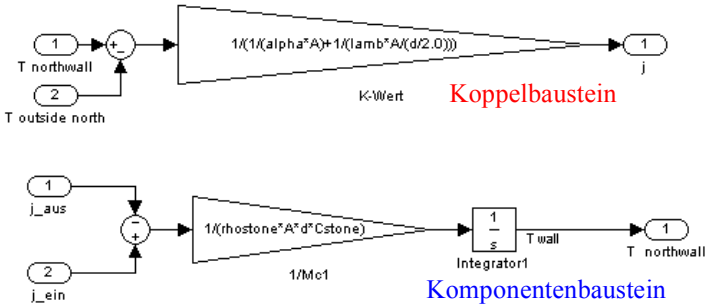


Abbildung 104: Energieflussverknüpfung: Komponentenbausteine und Koppelbausteine

6.1.3 Die Implementierung der phänomenologischen Verknüpfung

Abb. 105 zeigt die phänomenologische bzw. nicht berechnungskausale Implementierung des Hausmodells in *Modelica*. Die äußeren *I/O-Schnittstellen* stellen die Verbindung zu *Simulink* her. Die Kausalität im Modell geht auf den Einsatz von Aktor- (ideale Heizung) und Sensorbausteinen (Thermometer) zurück. Die Wärmeleitungs- und Wärmeübergangselemente dienen als Koppelbausteine. Die Komponentenbausteine sind der Wand- und Raumluftspeicher.

Die Parametrierung des Modells erfolgt über die bei der Compilierung erzeugten Parameterfenster.

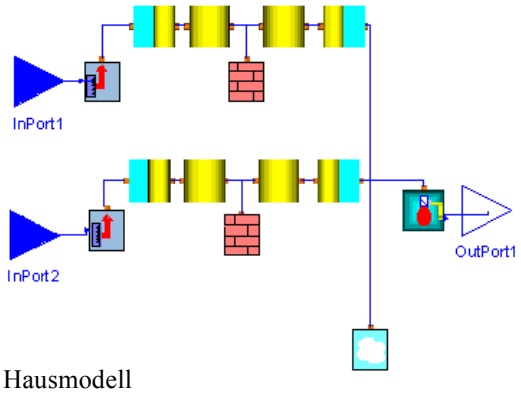


Abbildung 105: Phänomenologische Verknüpfung: Gesamtmodell

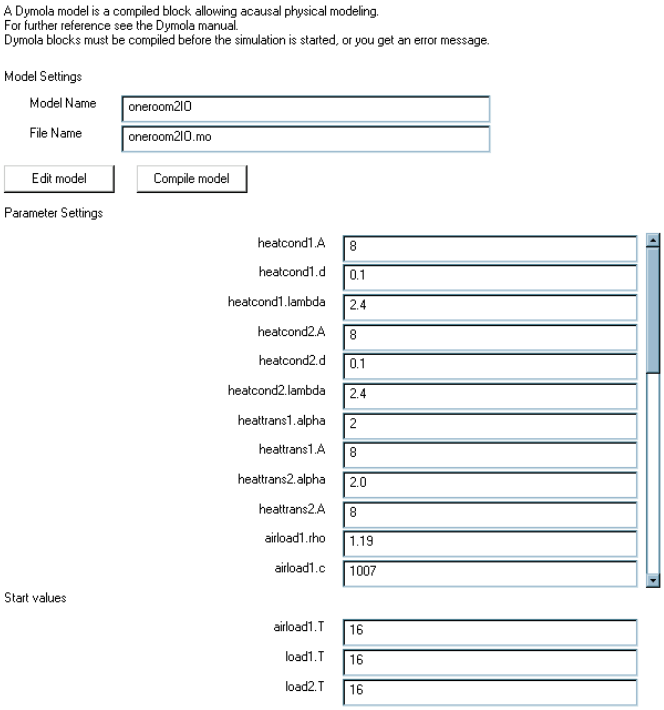


Abbildung 106: Automatisch erzeugtes Parameterfenster zur Parametrierung des *Modelica-Modells* in *Simulink*

Die Simulationsergebnisse zeigt Abb. 107. Die Raumlufttemperatur liegt zwischen den extremen Süd- und Nordaußenlufttemperaturen. Unter Beibehaltung einer gewissen Oszillation nähert sie sich asymptotisch einem Mittelwert an. Wie zu erwarten, ergibt sich in der Rechengenauigkeit keinerlei Differenz zwischen den drei unterschiedlichen Beschreibungsformen, da die mathematischen Modelle auch analytisch ineinander überführt werden können.

Im direkten Vergleich wird der intuitive Charakter der *Modelica*-Implementierung besonders deutlich. Das mathematische Modell widerspiegelt sofort die Struktur des realen Prozesses. Die Implementierung des physikalischen Ersatzmodells beschränkt sich auf die Komponentenebene. Dort spezifiziert sie in mathematischer Notation das bekannte physikalische Verhalten.

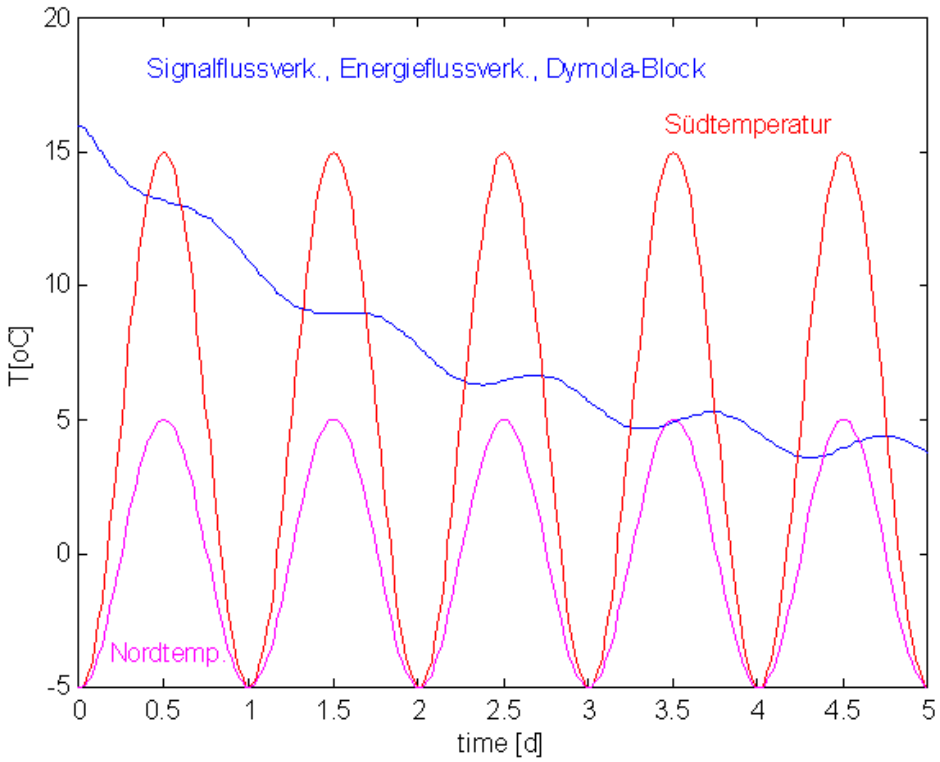


Abbildung 107: Berechnung der resultierenden Raumlufttemperatur bei Vorgabe der Außentemperatur auf der Nordseite und der Außentemperatur auf der Südseite. Die Ergebnisse nach den drei Modellabsätzen aus den Abschnitten 3.2.2, 3.2.3, 3.2.4 sind identisch und fallen in der blauen Kurve zusammen.

6.2 Das Konzept der konzentrierten Speicher

Bei der Modellierung in *Modelica* wurden die Wände häufig nur durch wenige Wandschichten in Form konzentrierter Speicher- und Wärmeleitungsschichten aufgebaut. Um diese Vorgehensweise zu rechtfertigen wird in Abb. 109 das dynamische Verhalten einer Wand mit 2, 4 und 200 elementaren Speichern für verschiedene Wandmaterialien und -stärken berechnet. Simulationen mit mehr als 200 Speicher- und Wärmeleitungsschichten ergaben keine signifikante Änderung. Daher wird im folgenden die *n=200*-Kurve als Referenz genutzt.

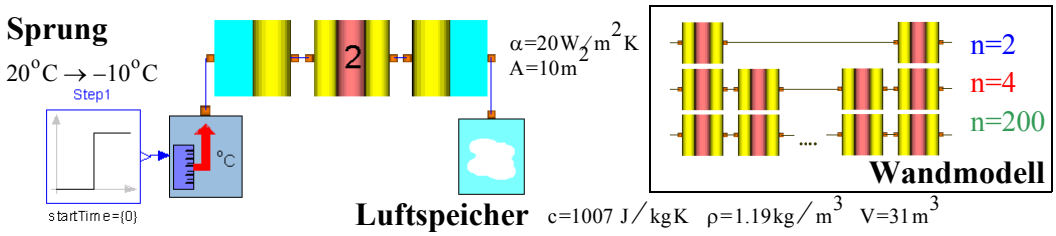


Abbildung 108: Objektdiagramm des Testbeispiels *(n=Anzahl der Wandschichten)*

Hierbei wurde ein extremer Sprung der Außenlufttemperatur von $\Delta T = -30^{\circ}\mathrm{C}$ angenommen und die resultierende Raumlufttemperatur beobachtet. Es zeigte sich, dass schon die Näherung *n=4* bei üblichen Baumaterialien und Wanddimensionierungen zu einer relativ geringen Abweichung des dynamischen Verhaltens gegenüber der *n=200*-Kurve führt (Abb. 109). Die geringere Anzahl von Speicherschichten täuscht eine schnellere Reaktion der Raumlufttemperatur auf den Sprung der Außenlufttemperatur vor. Es kommt zu einer zeitlichen Verschiebung von bis zu ca. 6 min. Die stationären Endwerte werden nicht beeinflusst. Aufgrund des extremen Sprungs und des daher sehr steilen Temperaturverlaufes korrespondiert die zeitliche Verschiebung allerdings mit Temperaturdifferenzen zur *n=200*-Kurve von bis zu $0.3^{\circ}\mathrm{C}$ während des Ausgleichsprozesses.

Anders ist dies bei sehr gut wärmedämmenden Wänden aufgrund einer niedrigen Wärmeleitfähigkeit oder einer sehr großen Wandstärke. Hier kann eine höhere Schichtzahl zur Modellierung erforderlich sein. Betrachtet man jedoch die relativen Abweichungen der Raumlufttemperatur bezogen auf den absoluten Betrag des Außenlufttemperatursprungs bzw. die relative zeitliche Abweichung bezogen auf die Dauer des Ausgleichsprozesses, so bleiben beide relativen Abweichungen unter 1%.

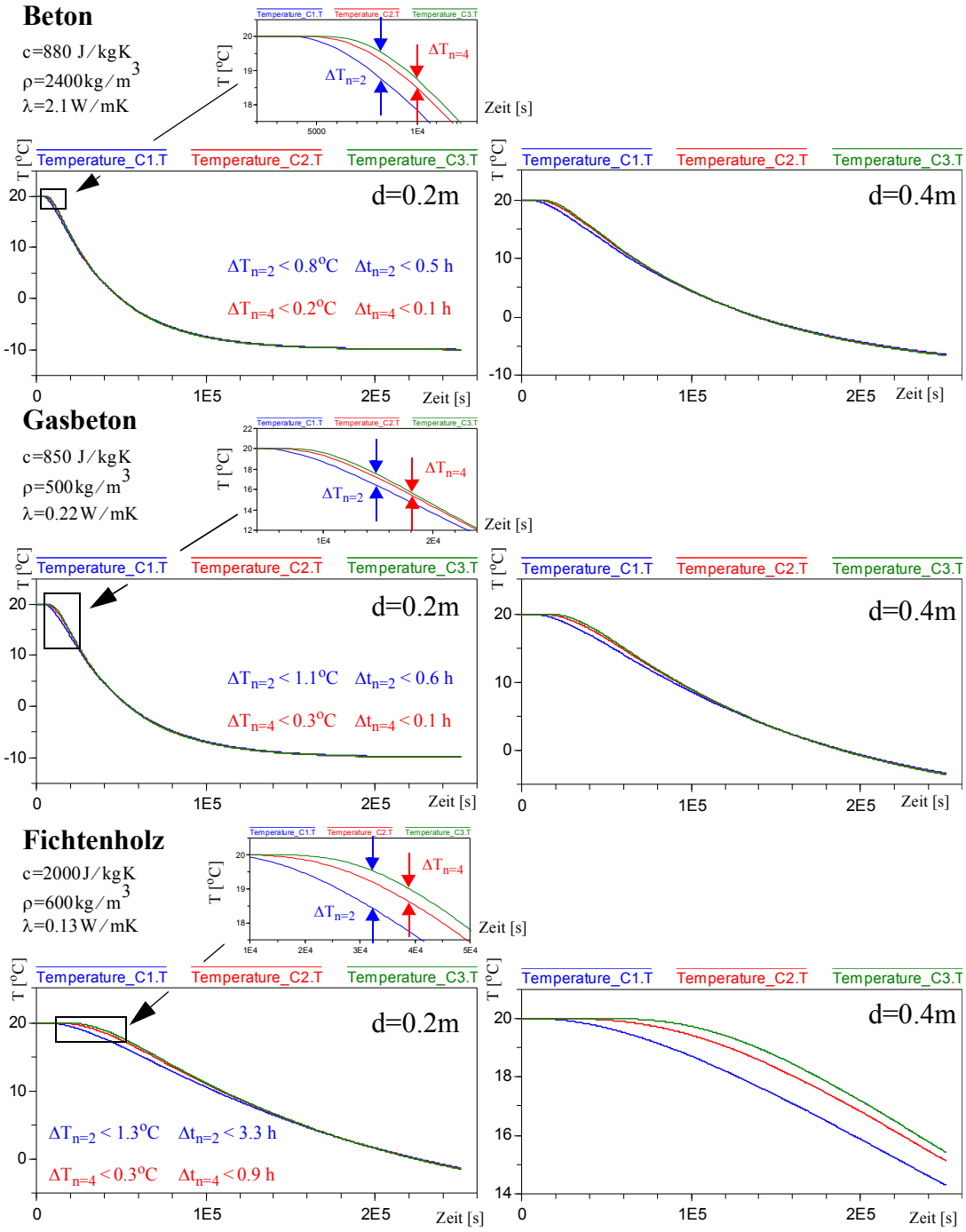


Abbildung 109: Unterschiede des thermischen Verhaltens einfacher Wände modelliert mit unterschiedlicher Anzahl (n) von Wandschichten, verschiedenen Wandmaterialien und -stärken
(blau: 2 Wandschichten, rot: 4 Wandschichten, grün: 200 Wandschichten).

6.3 Die Simulation eines Einfamilienhaus des Bestandes (EFHB)

6.3.1 Die Implementierung in Modelica

Im folgenden soll an einem konkreten, praxisorientierten Beispiel realistischen Umfangs der objektorientierte Ansatz demonstriert und auch validiert werden.

Die Fa. Bosch-Thermotechnik entwickelte einen Prüfstand für Wärmeerzeuger, der es ermöglicht ein reales Gerät innerhalb einer virtuellen, simulierten Umgebung zu betreiben. Man nennt diese Technik Emulation. Die thermischen Eigenschaften eines fiktiven Gebäudes, die herrschenden Klimaeinflüsse, der Aufbau und die Auslegung seines Heiznetzes, die genutzten Regelungsalgorithmen sowie das Nutzerverhalten werden in einem Simulationsmodell nachgebildet. Die reale Gastherme interagiert rückgekoppelt innerhalb ihres Prüfstandes mit diesem Modell [KiSi-01].

Im Rahmen des vom damaligen BMBF (Bundesministerium für Bildung und Forschung) geförderten Projektes Ikarus (Instrumente für Klimagasreduktionsstrategien) entstanden mehrere statistische Studien des Gebäudebestandes der Bundesrepublik Deutschland [z.B. HKS-97, Eic-94, EiSi-97]. Auf dieser Basis wurden in einem Forschungsprojekt der TU Hamburg-Harburg in Zusammenarbeit mit Bosch-Thermotechnik Modelle von Mustergebäuden erstellt, die nach dem Stand von 1997 55% der Wohnfläche des Bestandes und 61% des Neubaues der Bundesrepublik abdecken [Büh-96]. Zu diesen Gebäudemodellen wurden von Mitarbeitern der Fa. Bosch-Thermotechnik Heizungsnetze entworfen [Kri-98]. Als Simulationsumgebung wurde TRNSYS gewählt. Eine logische Validierung der TRNSYS-Gebäudemodelle erfolgte anhand der statistischen Datensätze des Ikarus-Projektes. Die Mustergebäudemodelle werden von Bosch-Thermotechnik zur Emulation genutzt [KiSi-01].

Wir griffen aus diesem Repertoire das ausgewählte Beispiel eines typischen Einfamilienhauses des Bestandes der Baujahre 1949 bis 1978 mit einer Wohnfläche von 178m^2 heraus. Hierbei handelt es sich um ein freistehendes Einfamilienhaus mit einem 45^o-Schrägdach und einem nach Westen zeigenden Giebel (siehe Abb. 110). Die Räume im Obergeschoss besitzen im oberen Teil Dachschrägen. Der Dachboden im Spitzgiebel ist nach außen ungedämmt. Das Obergeschoss ist sowohl zum Dachboden als auch nach außen gedämmt. Das Gebäude besitzt keinen Keller.

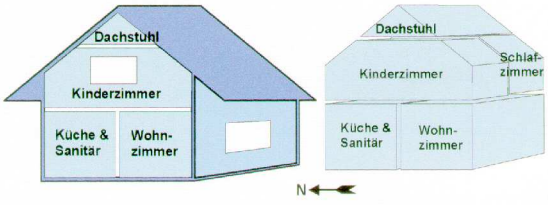


Abbildung 110: Prinzipskizze des ausgewählten Einfamilienhauses als typischen Repräsentanten des Bestandes aus den Baujahren 1949 bis 1978 mit einer Wohnfläche von 178m^2 [KiSi-01].

Es existiert ein vollständiges TRNSYS-Modell des Gebäudes [Kri-98], das im folgenden als Standard genutzt wird. Ziel der Diskussion ist es, zu überprüfen, ob der *Modelica*-Ansatz bei gleichen Randwerten die TRNSYS-Resultate reproduziert.

Bei diesem Vergleich wird das verfügbare TRNSYS-Modell als Referenz betrachtet. Parametervariationen oder größere Veränderungen des Gebäudemodells waren aufgrund der äußeren Projektrandwerte nur in beschränktem Umfang realisierbar.

Von einer korrekten Abbildung des Gebäudeverhaltens durch das TRNSYS-Modell kann man zunächst ausgehen. Neben der Verwendung erprobter Mustergebäudemodelle [KiSi-01] stellt das Simulationssystem TRNSYS ein etabliertes Tool dar, das häufig zur Gebäude- und Anlagenplanung eingesetzt wird [Trn-02, Hil et.al.-01]. So gehört TRNSYS auch zu den ersten 7 im Rahmen des BESTTEST der International Energy Agency überprüften Simulationssysteme [JuNe-95]. Zu den TRNSYS Komponentenmodellen existieren experimentelle und theoretische Studien bezüglich der Validierung auf Komponentenbasis [Kle et al.-96]. Die mathematische Behandlung des Gesamtmodelles ist ebenfalls etabliert. Die Simulationsumgebung TRNSYS verwendet ausschließlich standardisierte numerische Verfahren.

Um zunächst Erfahrungen zu sammeln und zur eigenen Beurteilung, wurde TRNSYS von uns vorab in einer experimentellen Studie zur Überprüfung des thermischen Verhaltens eines mit der notwendigen Sensorik ausgestatteten realen Einfamilienhauses eingesetzt [SpMe-01]. Die Grenzen der numerischen Simulation unter TRNSYS werden bei der Untersuchung von Systemen mit kleinen Zeitkonstanten erreicht. Das Gebäudemodell und die Numerik sind für die schnelle Berechnung ganzer Jahresdurchläufe in der Regel in Stundenschritten ausgelegt. Die Zeitschritte sind nur konstant vorwählbar. Zeitschritte im Sekundenbereich führen zu numerischen Instabilitäten. Im verfügbaren TRNSYS-Referenzmodell wurde als Simulationsschrittweite *3 min* gewählt. Als *timebase* zur Berechnung des Übertragungsverhaltens der Wände wurde 1 h gewählt. Hierauf ist die zu beobachtende treppenförmige Modulation der TRNSYS-Temperaturverläufe im Stundenrhythmus zurückzuführen.

Ein vollständiges Objektdiagramm des Simulationsmodells in *Modelica* findet sich in Abb. 111. Das Diagramm aggregiert sowohl das Gebäudemodell *(rechter Teil)* als auch das Umgebungsmodell *(linker Teil)*. Der wesentliche Teil des Umgebungsmodells wurde in Form eines eigenen Objektdiagrammes schon in Abb. 86 beschrieben.

Die Implementierung des Gebäudes im *Modelica*-Ansatz orientiert sich an dem Beuken-Modell ([Beu-36], *z.B.* in [RoZi-97]). Die Wände werden in Außen- und Innenwandhälften unterteilt, die selbst wieder einzelne wärmeleitende bzw. wärmespeichernde Schichten aggregieren. Die Außenwände werden in mindestens 3 Schichten untergliedert, so dass keine Wand mit weniger als 4 speichernden Schichten modelliert wird. Die Innenwandhälften werden entsprechend ihrer räumlichen Anordnung als Umschließungsflächen der thermischen Zonen betrachtet. Die Außenwandhälften bilden die äußere Hüllfläche des Gebäudes und werden nur von den Fenstern durchbrochen.

Die Grundstruktur des Gebäudes wird entsprechend den fünf Räumen in fünf thermische Zonen überführt. Aufgrund der unterschiedlichen Anzahl von Nachbarzonen sind thermische Zonen mit 7 bzw. 9 möglichen Ankopplungen für Nachbarzonen bzw. auch Außenwandhälften einzusetzen.

Die Raumtemperatur der beheizten thermischen Zonen (Küche, Wohn-, Schlaf- und Kinderzimmer) wird lokal mittels PI-Regler und idealen, aber in der maximalen Heizleistung limitierten Einzelraumheizungen auf dem jeweiligen Sollwert gehalten. Das Verhältnis von konvektiver und Strahlungswärmeabgabe der Heizkörper ist bei der Parametrierung einstellbar.

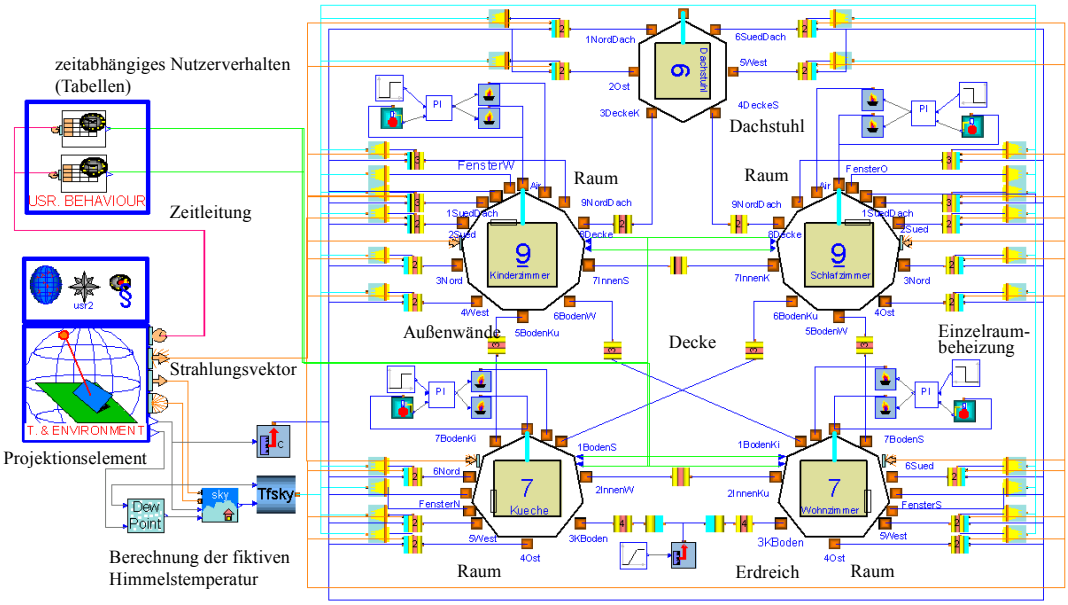


Abbildung 111: Vollständiges Objektdiagramm eines Einfamilienhauses in *Modelica* spezifiziert. Dargestellt wurde die höchste Hierarchieebene des Gesamtmodells. Man erkennt sowohl das Umgebungsmodell *(links)* als auch das Gebäudemodell *(rechts)*.

Die Berechnung der solaren Einstrahlung auf die Vorzugsrichtungen ist Teil des Umgebungsmodells. Die äußeren klimatischen Einflüsse werden aus einem Wetterdatenfile gelesen. Dieses enthält in Abhängigkeit von der Ortszeit die Außenlufttemperatur, die solare Einstrahlung auf eine horizontale Fläche, aufgeteilt in direkten und diffusen Anteil, sowie die relative Feuchte.

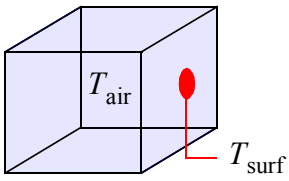
Die Außenlufttemperatur wird über konvektive Wärmeübergänge an alle Außenflächen angeschlossen. Die relative Feuchte wird in obigem Modell nicht bilanziert, sondern fließt in die Berechnung der fiktiven Himmelstemperatur ein. Dies ist eine Rechengröße zur Beschreibung des langwelligen Strahlungsaustausches der Gebäudehülle mit der Umgebungsatmosphäre.

Die solare Einstrahlung wird an die Gebäudeaußenflächen angekoppelt. Hierzu wird die solare Einstrahlung auf vordefinierte Fassadenorientierungen berechnet und über einen Strahlungsvektor dem Gebäudemodell zur Verfügung gestellt. Es wurden vorab vier Orientierungen *SüdVertikal, OstVertikal, WestVertikal* und das *geneigte Süddach* als relevant ausgewählt. Die Einstrahlung berücksichtigt den *Beam*-Anteil, den diffusen Anteil sowie Reflexionen am Boden. Die mit dem Absorptionsfaktor bzw. der absorbierenden Fläche multiplizierten Werte werden den jeweiligen Fassadenflächen zugeschlagen.

Anders verhält es sich bei den Fenstern. Hier wird der transmittierte und auf die Fensterfläche bezogene Anteil mittels eines Strahlungsverteilers auf die Innenflächen der Räume verteilt. Darüber hinaus wurde innerhalb der Fensterkomponenten auch ein Lüftungswärmeverlust durch Undichtigkeiten sowie die bewusste Lüftung nach einem Zeitschema berücksichtigt. Die Luftaustauschraten werden täglichen Tabellen entnommen, in denen das Nutzerverhalten

beschrieben ist. Der Zugriff erfolgt über die momentane Uhrzeit bzw. Kalenderzeit, welche die Solarmodelle zur Verfügung stellen.

Es werden zwei Testfälle angenommen. Hierbei werden jeweils 12 Tage simuliert, wovon nur die Resultate der letzten 5 Tage verglichen werden. Die innerhalb eines Raumes empfundene Temperatur hängt neben der Raumlufttemperatur (T_{air}) stark von den Oberflächentemperaturen der den Raum umschließenden Flächen $(T_{surf\ i})$ ab. Daher wird innerhalb der bewohnten Räume der mit den Flächengrößen $(A_{surf\ i})$ gewichtete Mittelwert der Temperaturen zur Charakterisierung des Raumes herangezogen.



$$T_{sens} = \frac{1}{2} \cdot \left(\frac{\sum_i A_{surf\ i} \cdot T_{surf\ i}}{\sum_i A_{surf\ i}} + T_{air} \right)$$

Abbildung 112: Empfindungstemperatur (T_{sens}), berechnet als flächengewichteter Mittelwert.

6.3.2 Die Validierung des EFHB-Modells

Testfall A schreibt einen Luftwechsel nach vorgegebenem Zeitschema bei konstanten Sollwerttemperaturen vor (Abb. 116). Die relative Luftwechselzahl bezogen auf das Raumvolumen des konstanten Grundluftwechsels beträgt 0.6 / 1 [h]. Hinzu kommt täglich viermaliges Stoßlüften (um 8.00 Uhr, 12.00 Uhr, 18.00 Uhr und 22.00 Uhr) mit einer relativen Luftwechselzahl von 10 / 1 [h] mit einer Dauer von 0.5 h. Die Sollwerttemperatur beträgt für das Schlafzimmer konstant 16°C, für alle anderen Zimmer konstant 21°C.

Testfall B schreibt eine konstante relative Luftwechselzahl von 0.6/h vor (Abb. 120). Die Solltemperaturen von 16°C im Schlafzimmer und 21°C in den anderen Zimmern werden von 6.00 Uhr bis 9.00 Uhr und 19.00 Uhr bis 22.00 Uhr um 4°C angehoben und danach wieder abgesenkt.

Der betrachtete Zeitraum überdeckt recht kalte Wintertage. In Abb. 113 findet sich neben der Außenlufttemperatur die fiktive Himmelstemperatur zur Berechnung der Abstrahlung der Gebäudeoberflächen. Bei der fiktiven Himmelstemperatur ergeben sich in der letzten Nacht des betrachteten Simulationsintervalls Abweichungen, die auf die mangelnde Kenntnis des nächtlichen Bewölkungsgrades zurückzuführen sind. TRNSYS nimmt als nächtlichen Bedeckungsgrad einen Mittelwert des vorangegangenen Nachmittags an. Der *Modelica*-Ansatz hält den letzten Wert konstant. Hier erreicht man die Grenzen simulativer Validierung, keine der beiden Methoden ist absolut korrekt.

Da zur Berechnung der langwelligen Abstrahlung nur ein mit dem geometrischen Sichtfaktor (f_{sky}) und der Außentemperatur korrigierter Wert genutzt wird (siehe Abschnitt 5.3.1.2), ist die Abweichung unerheblich. Diese Aussage wird auch von beiden Testfällen A und B bestätigt, da die Temperatur im unbeheizten Dachstuhl praktisch keine Abweichungen zwischen TRNSYS und *Modelica* aufweist. Dort sollte jedoch die Auswirkung einer falschen fiktiven Himmelstemperatur am größten sein.

Darüberhinaus ist die während des Zeitraums konstant angenommene Erdbodentemperatur unterhalb des nicht unterkellerten Hauses abgebildet. Der Wert ist recht hoch, er musste jedoch als Vorgabe der verfügbaren Wetterdaten akzeptiert werden.

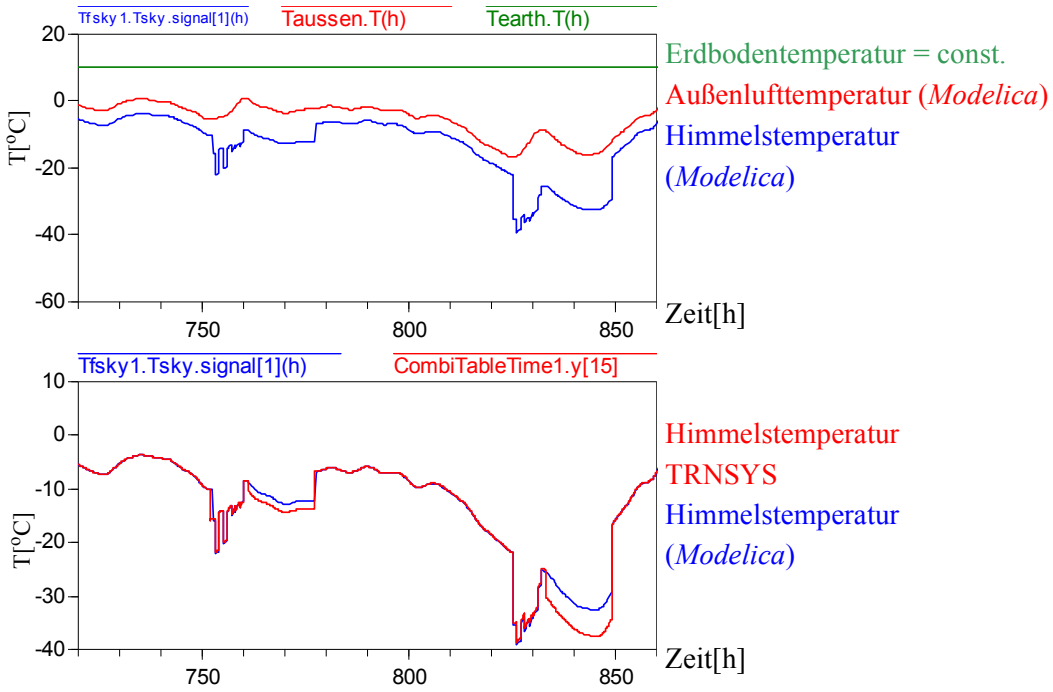


Abbildung 113: Die äußeren klimatischen Einflüsse während des betrachteten Zeitintervalls. Es handelt sich um kalte, jedoch teilweise sonnige Wintertage. Die Werte für Erdbodentemperatur und Außenlufttemperatur entstammen dem Wetterdatenfile. Die Himmelstemperatur wird berechnet. Im unteren Diagramm erfolgt der Vergleich mit TRNSYS.

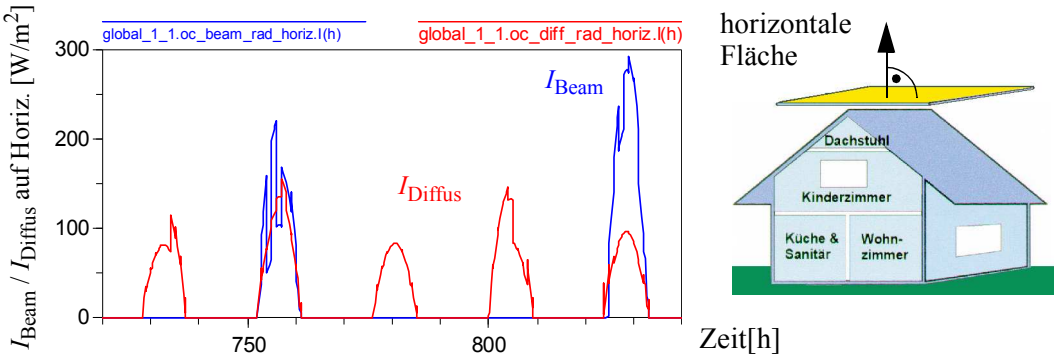


Abbildung 114: Dargestellt ist die *Beam*- und die Diffusstrahlung auf eine horizontale Oberfläche in [W/m^2]. Die Werte entstammen dem Wetterdatenfile.

Die Abbildung Abb. 115 zeigt die solare *Beam*-Strahlung auf die Gebäudefassaden in Nord-, Ost-, Süd- und Westrichtung sowie auf das Süddach längs der jeweiligen Flächennormalen. Die Darstellung beschränkt sich auf die relevanten 5 Tage. Die *Beam*-Strahlung längs der Flächennormalen wird in [W/m^2] aus der *Beam*-Strahlung auf eine horizontale Fläche (Abb. 114) berechnet. Eine solare *Beam*-Strahlung auf den Fassaden ist nur am 2. und 5. Tag vorhanden.

Man erkennt, auf der Ostfassade, in der das Schlafzimmer liegt, steigt die *Beam*-Strahlung zunächst nahezu vertikal an und fällt dann in Form eines Sägezahnes ab.

Auf der Südfassade, in der das Wohnzimmerfenster liegt, ergibt sich ein nahezu gaußförmiges Profil für den Tagesverlauf der solaren *Beam*-Strahlung.

Auf der Westfassade, in der das Kinderzimmerfenster liegt, steigt die Strahlung zunächst langsamer an, um dann bei Sonnenuntergang nahezu vertikal abzufallen. Hier ergibt sich eine Diskrepanz zu TRNSYS.

Das TRNSYS-Ergebnis zeigt am 5. Tag eine um etwa 100 W/m^2 geringere solare Einstrahlung auf die Westfassade und weist nicht das zu erwartende sägezahnförmige Profil auf. Eine plötzliche Bedeckungsänderung kann dies nicht verursachen, da sich diese auch auf den anderen Fassaden auswirken würde.

Tests mit TRNSYS ergaben, dass es sich hierbei um ein Artefakt handeln muss. In TRNSYS wird zusätzlich zur Interpolation eine quadratische Glättungsfunktion *(smooth)* eingesetzt, die zur Vermeidung numerischer Instabilitäten bei Sonnenuntergang den wahren Anstieg der solaren Einstrahlung auf die Westfassade gegen Abend unterschätzt und den tatsächlichen Verlauf stark glättet.

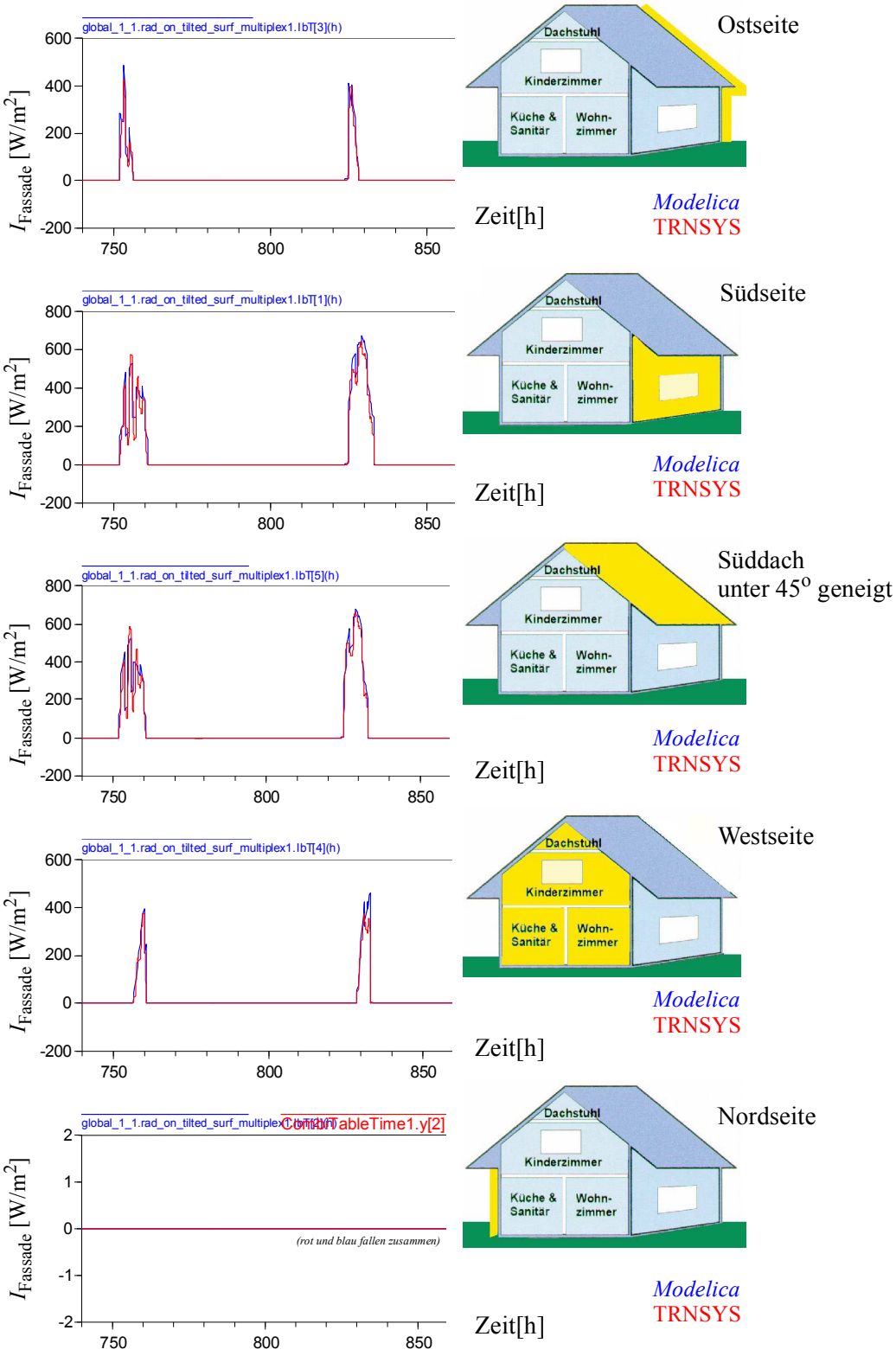


Abbildung 115: Solare *Beam*strahlung auf die Gebäudefassaden in Nord-, Ost-, Süd- und Westrichtung sowie auf das Süddach. Blau dargestellt sind die *Modelica* Ergebnisse, rot dargestellt sind die TRNSYS-Ergebnisse.

6.3.3 Der Testfall A: Stoßlüften

Testfall A beschreibt ein Stoßlüften nach vorgegebenem Zeitschema und konstant zu haltenden Raumtemperatursollwerten. Die Abb. 117 zeigt die Simulationsergebnisse zu den Raumlufttemperaturen des *Modelica*-Modells sowie des TRNSYS-Referenzmodells für Testfall A. Da innerhalb der geheizten Räume die Raumlufttemperaturen geregelt werden und bei beiden Modellansätzen überwiegend auf dem Sollwert bleiben, kann ihre Übereinstimmung nicht als signifikantes Kriterium zur Überprüfung der Modellansätze genutzt werden.

Während des Stoßlüftens kommt es allerdings - aufgrund der limitierten Heizleistung - trotz der Regelung zu Einbrüchen der Raumlufttemperatur. Folgende Kriterien können daher zum Vergleich der beiden Modellansätze herangezogen werden: zum einen die Tiefe der Einbrüche der Raumlufttemperatur im Lüftungsfall, zum anderen die erforderlichen Heizströme, darüber hinaus die Raumlufttemperatur innerhalb des unbeheizten Dachstuhls. Die Raumlufttemperaturen (Abb. 117) zeigen eine nahezu vollständige Übereinstimmung zwischen beiden Modellansätzen.

In Abb. 118 schließen sich die Empfindungstemperaturen an. Diese können, da die Raumlufttemperatur geregelt wird, durchaus Abweichungen zum Sollwert der Raumlufttemperatur aufweisen. Insbesondere bei solarer Einstrahlung wird dies deutlich. Dies erklärt sich durch die Berücksichtigung der Oberflächentemperaturen der Raumwände in der Empfindungstemperatur. Diesen wird jedoch gerade die durch die Fenster in den Raum einfallende solare Einstrahlung zugeschlagen. Das Wohnzimmer mit seinem ausgedehnten Südfenster sowie, entsprechend dem Sonnenlauf von Ost nach West, etwas später das Kinderzimmer mit seinem Westfenster zeigen deutliche Maxima der Empfindungstemperatur in Abb. 118.

An dieser Stelle sei auf die Bedeutung der Interpolation der nur stündlich vorhandenen Wetterdaten hingewiesen. Unter Anwendung einer rein linearen Interpolation der stündlichen Strahlungswerte ergaben sich im *Modelica*-Modell als Artefakt extreme Peaks der solaren Einstrahlung auf die Westfassade bei Sonnenuntergang. Diese schlugen direkt auf die Empfindungstemperatur durch. Erst die Berücksichtigung des Kurvenverlaufes der extraterrestrischen Einstrahlung zur Interpolation vermied diese Artefakte.

Ein Einfluss der thermischen Speicherfähigkeit der Raumwände zeigt sich auch bei der geringeren Tiefe der Temperatureinbrüche während des Stoßlüftens in Abb. 118 im Vergleich zur Raumlufttemperatur in Abb. 117.

In Abb. 119 werden die momentan zugeführten Heizleistungen zur Aufrechterhaltung des Raumlufttemperatursollwertes aufgezeigt. Die Heizungen sind ideal im Zeitverhalten, nur die maximal verfügbare Heizleistung pro Raum ist limitiert. Die volle Heizleistung wird nur während des Stoßlüftens angefordert. Hier könnte man durch Abschalten der Heizkörper während der Lüftungsphase Energie einsparen.

Auch hier findet sich eine nahezu vollständige Übereinstimmung der Ergebnisse beider Simulationsansätze. Eine geringfügige Abweichung, wie die geringe Diskrepanz der solaren *Beam*-Strahlung auf die Westfassade am 5. Tag erwarten lässt (siehe Abb. 115), zeigt sich im Kinderzimmer. Da TRNSYS von einer geringeren solaren Einstrahlung ausgeht, fordert es auch eine geringfügig höhere Heizleistung im Kinderzimmer. Die Abweichungen bleiben jedoch gering.

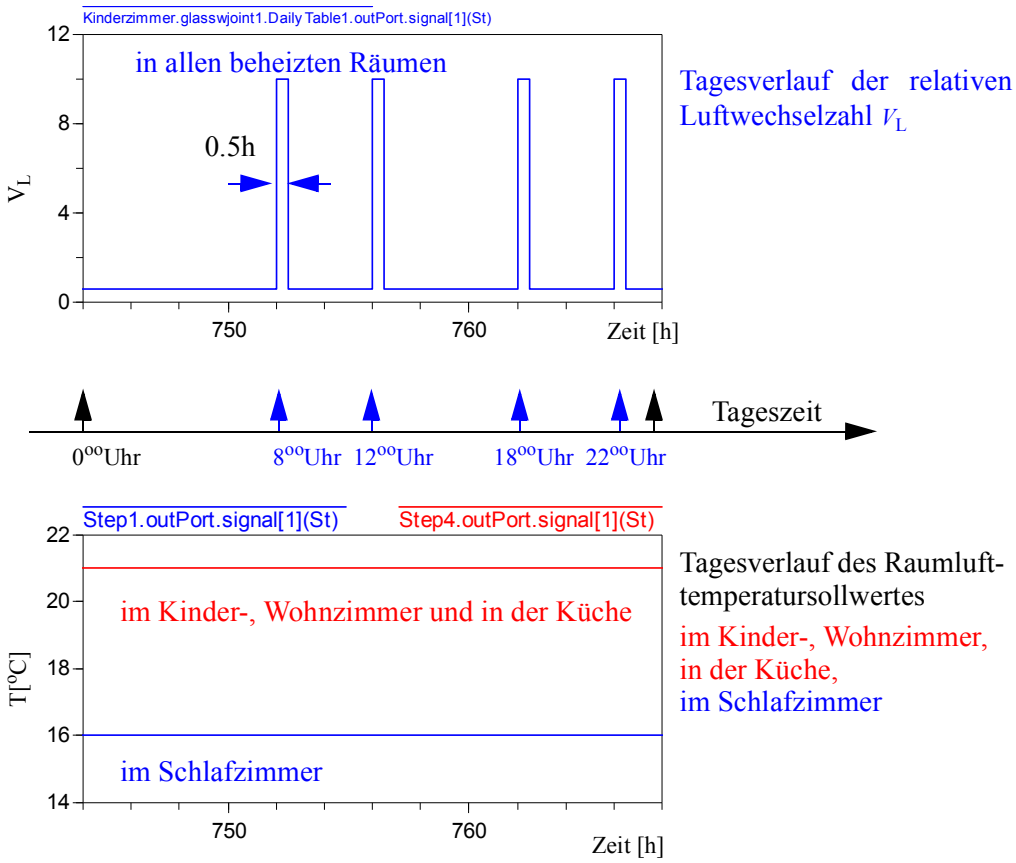
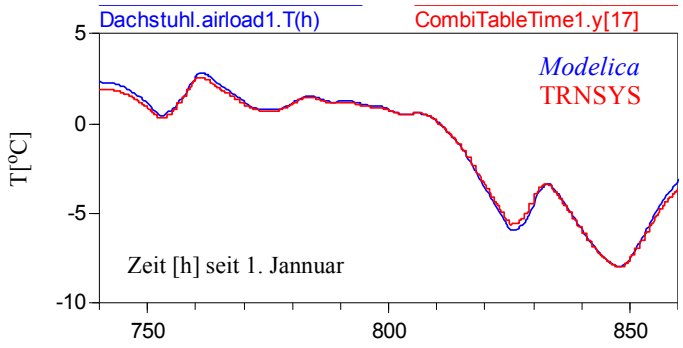
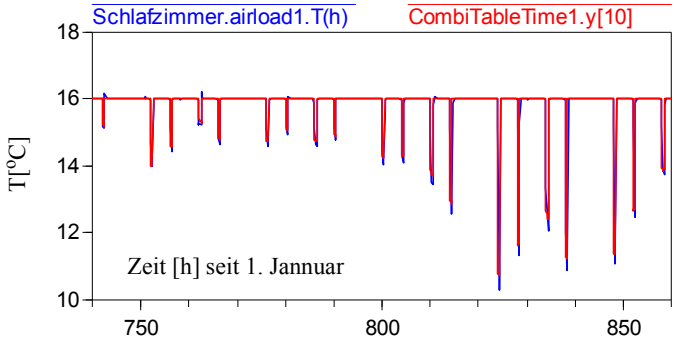


Abbildung 116: Zeitschema des täglichen Sollwertes der Luftwechselzahl und der Raumlufttemperaturen, Fall A

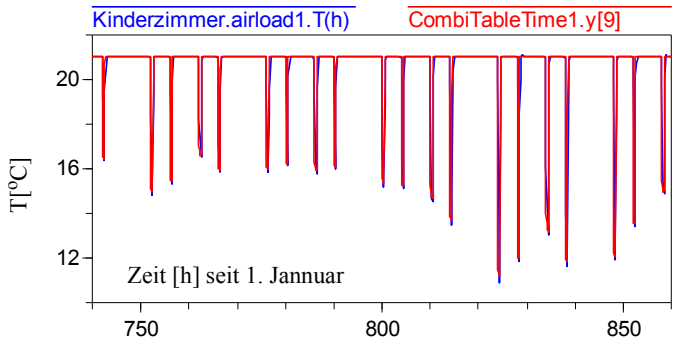


Die Raumlufttemperatur im unbeheizten **Dachstuhl**
Blau dargestellt ist das Ergebnis der *Modelica*-Simulation.
Rot dargestellt ist das Ergebnis des TRNSYS-Referenzmodells.



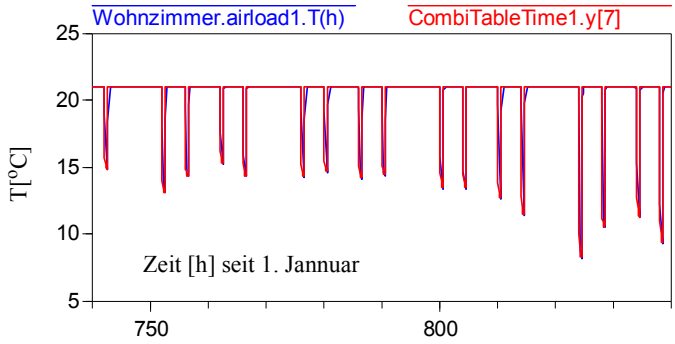
Die Raumlufttemperatur im **Schlafzimmer**
bei einem Sollwert von 16°C
Die Einbrüche ergeben sich durch das stoßartige Lüften.

Modelica
TRNSYS



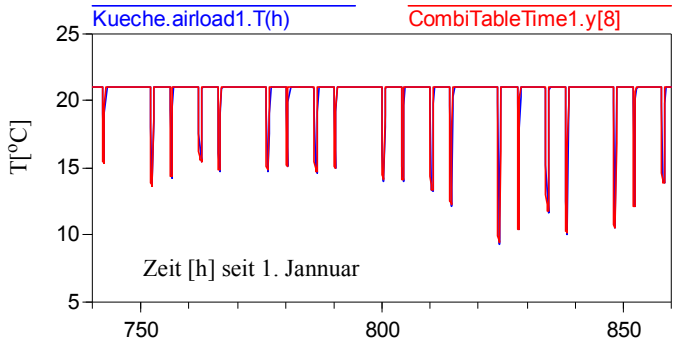
Die Raumlufttemperatur im **Kinderzimmer**
bei einem Sollwert von 21°C
mit stoßartigem Lüften

Modelica
TRNSYS



Die Raumlufttemperatur im **Wohnzimmer**
bei einem Sollwert von 21°C
mit stoßartigem Lüften

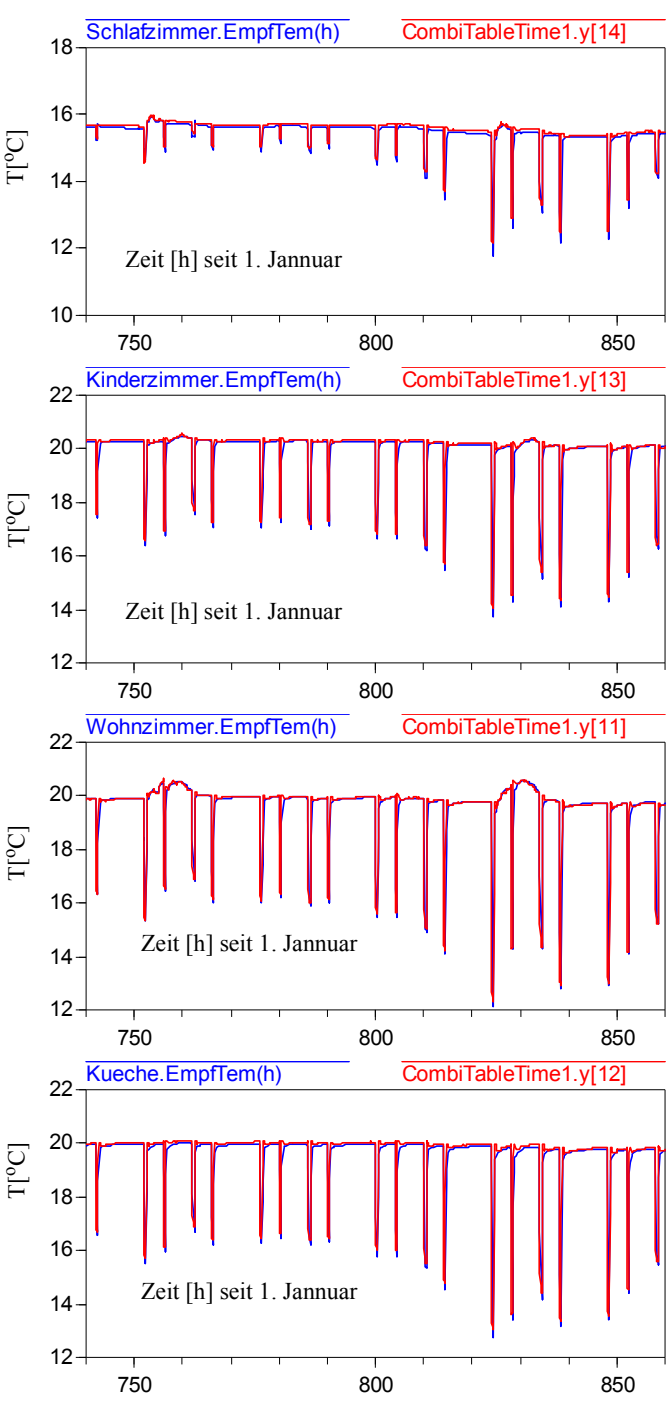
Modelica
TRNSYS



Die Raumlufttemperatur in der **Küche** bzw. im **Bad**
bei einem Sollwert von 21°C
mit stoßartigem Lüften

Modelica
TRNSYS

Abbildung 117: Die Raumlufttemperaturen im Einfamilienhaus, Fall A



Die Empfindungstemperatur im **Schlafzimmer** bei einem Sollwert von 16°C
Die Einbrüche ergeben sich durch das stoßartige Lüften.

Modelica
TRNSYS

Die Empfindungstemperatur im **Kinderzimmer** bei einem Sollwert von 21°C mit stoßartigem Lüften.
Die beiden Maxima ergeben sich durch solare Einstrahlung.

Modelica
TRNSYS

Die Empfindungstemperatur im **Wohnzimmer** bei einem Sollwert von 21°C mit stoßartigem Lüften.
Sichtbar sind auch die Maxima durch solare Einstrahlung.

Modelica
TRNSYS

Die Empfindungstemperatur in der **Küche** bzw. im **Bad** bei einem Sollwert von 21°C mit stoßartigem Lüften

Modelica
TRNSYS

Abbildung 118: Die Empfindungstemperaturen im Einfamilienhaus, Fall A

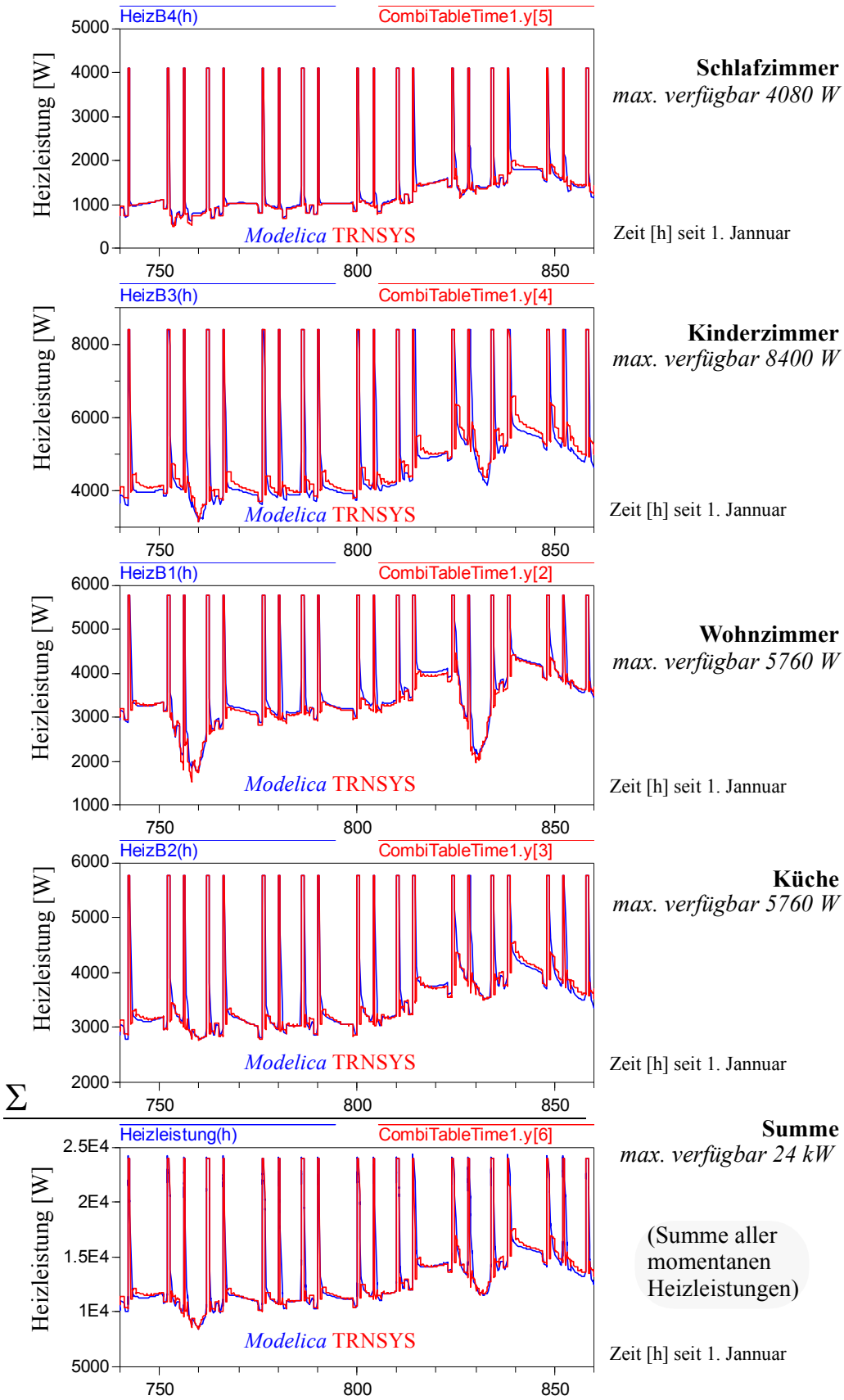


Abbildung 119: Momentan zugeführte Heizleistung der Einzelraumbeheizung, Fall A

6.3.4 Der Testfall B: Zeitabhängige Raumtemperatursollwerte

Testfall B beschreibt einen zeitabhängigen Wechsel der Raumlufttemperatursollwerte. Dies simuliert eine zeitabhängige Benutzeranforderung entsprechend einem üblichen Tagesrhythmus. Auf das Stoßlüften wird verzichtet, stattdessen wird eine konstante Luftwechselrate angenommen.

In Abb. 120 finden sich die konstante Luftwechselrate sowie das Zeitschema der Raumtemperatursollwerte des Testfalls B. Das betrachtete Zeitintervall sowie die Wetterdaten sind identisch zu Testfall A.

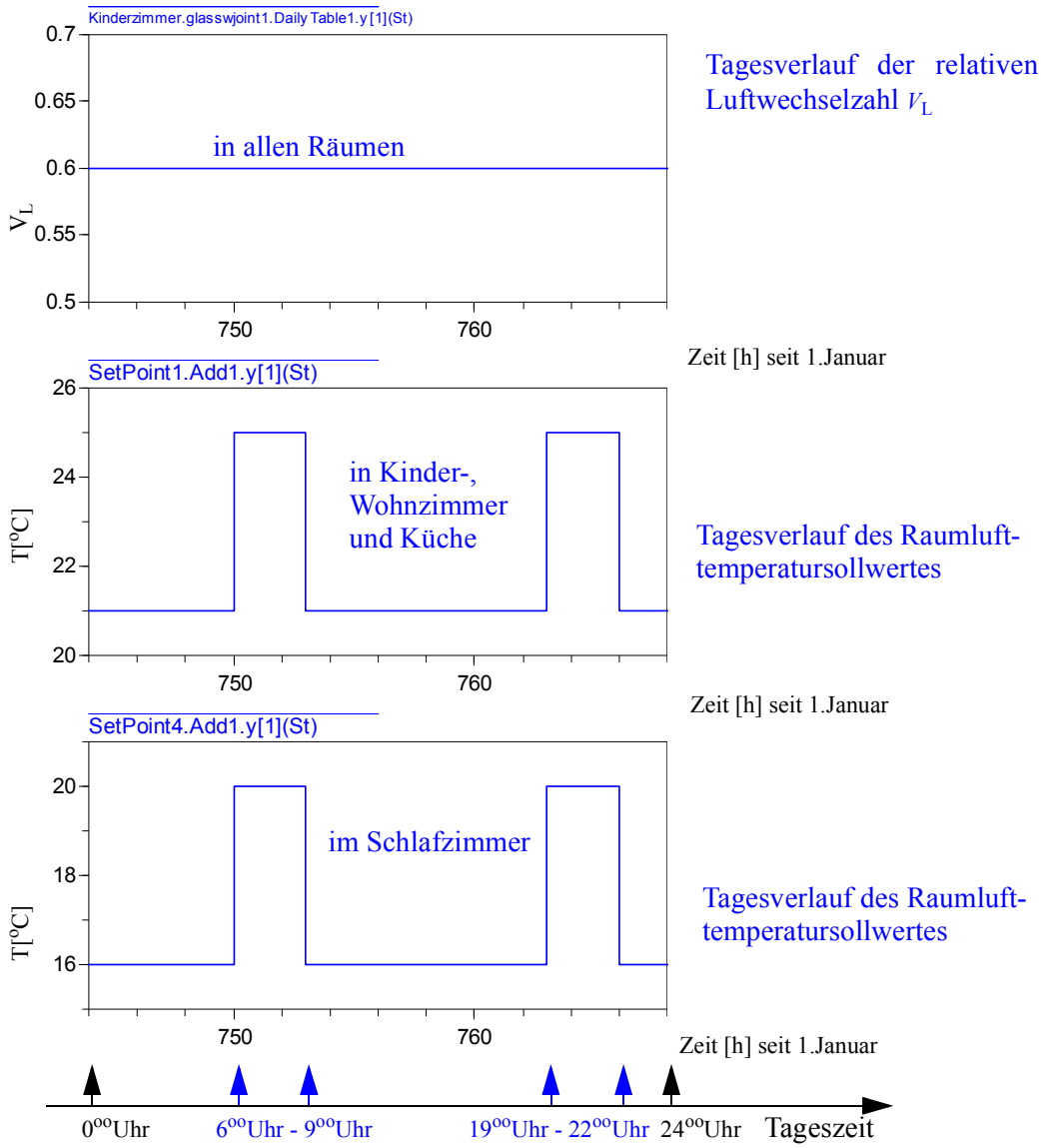
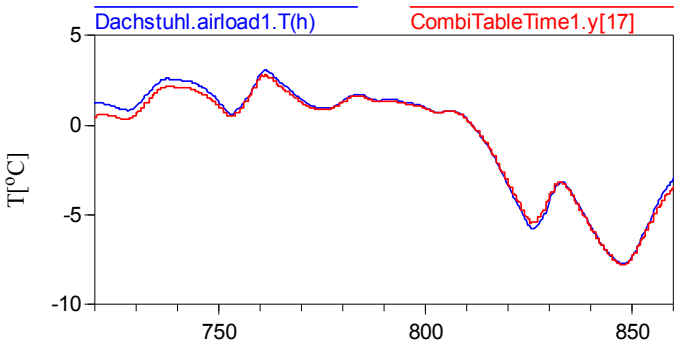


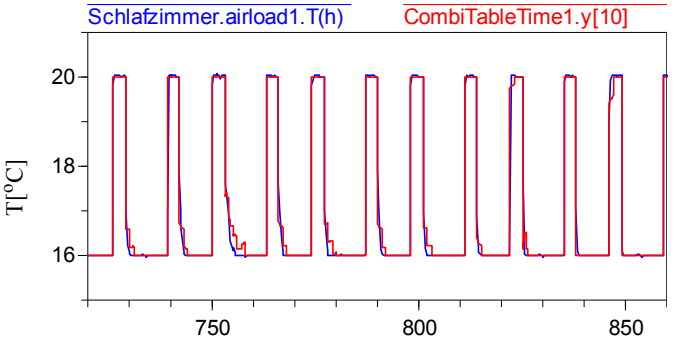
Abbildung 120: Zeitschema des Sollwertes der Luftwechselzahl und der Raumlufttemperaturen, Fall B



Die Raumlufttemperatur im unbeheizten **Dachstuhl** Blau dargestellt ist das Ergebnis der *Modelica*-Simulation. Rot dargestellt ist das Ergebnis des TRNSYS-Referenzmodells.

Zeit [h] seit 1. Jannuar

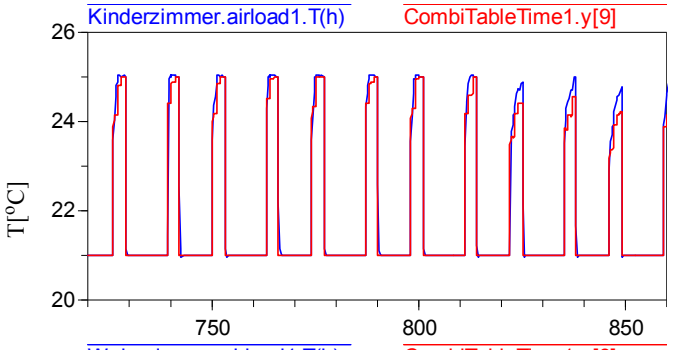
Modelica
TRNSYS



Die Raumlufttemperatur im **Schlafzimmer** bei einem nach Zeitschema wechselnden Sollwert von 16°C bzw.20°C

Zeit [h] seit 1. Jannuar

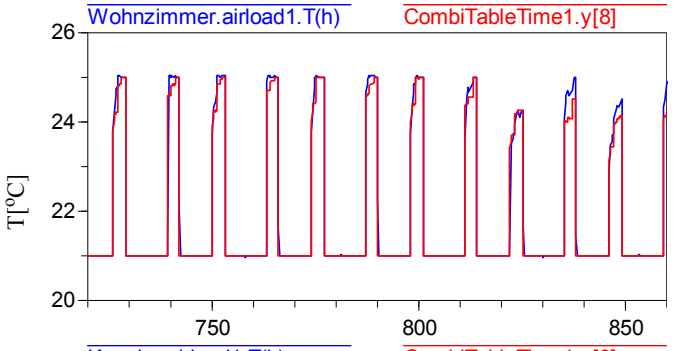
Modelica
TRNSYS



Die Raumlufttemperatur im **Kinderzimmer** bei einem nach Zeitschema wechselnden Sollwert von 21°C bzw.25°C

Zeit [h] seit 1. Jannuar

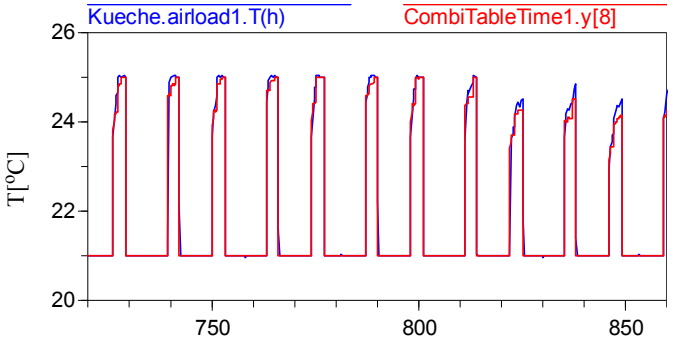
Modelica
TRNSYS



Die Raumlufttemperatur im **Wohnzimmer** bei einem nach Zeitschema wechselnden Sollwert von 21°C bzw.25°C

Zeit [h] seit 1. Jannuar

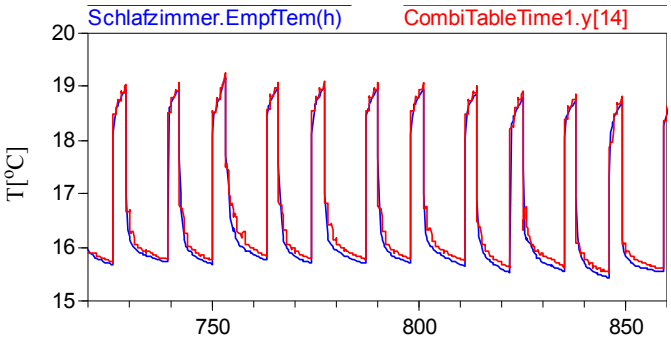
Modelica
TRNSYS



Die Raumlufttemperatur in der **Küche** bzw. im **Bad** bei einem nach Zeitschema wechselnden Sollwert von 21°C bzw.25°C

Zeit [h] seit 1. Jannuar

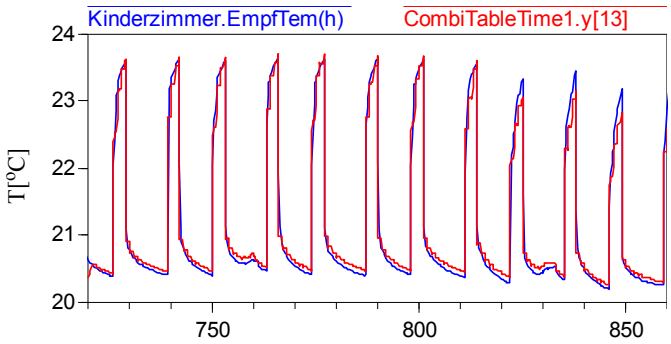
Abbildung 121: Die Raumlufttemperaturen im Einfamilienhaus, Fall B



Die Empfindungstemperatur im **Schlafzimmer**
Blau dargestellt ist das Ergenbis der *Modelica*-Simulation.
Rot dargestellt ist das Ergebnis des TRNSYS-Referenzmodells.

Zeit [h] seit 1. Jannuar

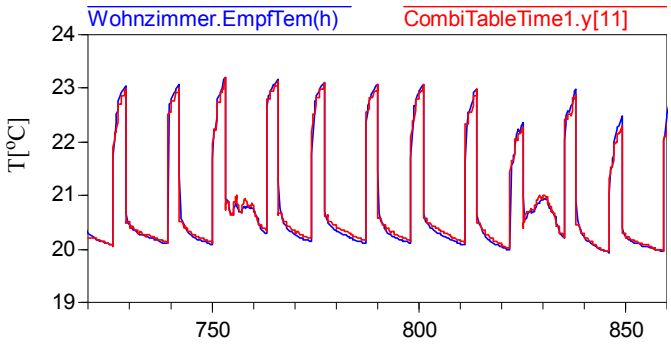
Modelica
TRNSYS



Die Empfindungstemperatur im **Kinderzimmer** bei einem nach Zeitschema wechselnden Sollwert von 16°C bzw.20°C

Zeit [h] seit 1. Jannuar

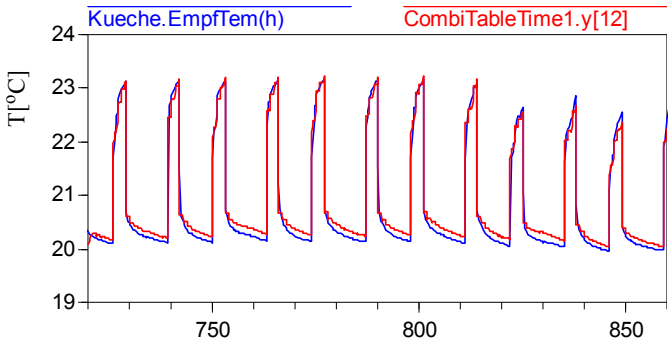
Modelica
TRNSYS



Die Empfindungstemperatur im **Wohnzimmer** bei einem nach Zeitschema wechselnden Sollwert von 21°C bzw.25°C

Zeit [h] seit 1. Jannuar

Modelica
TRNSYS



Die Empfindungstemperatur in der **Küche** bzw. im **Bad** bei einem nach Zeitschema wechselnden Sollwert von 21°C bzw.25°C

Zeit [h] seit 1. Jannuar

Modelica
TRNSYS

Abbildung 122: Die Empfindungstemperaturen im Einfamilienhaus, Fall B

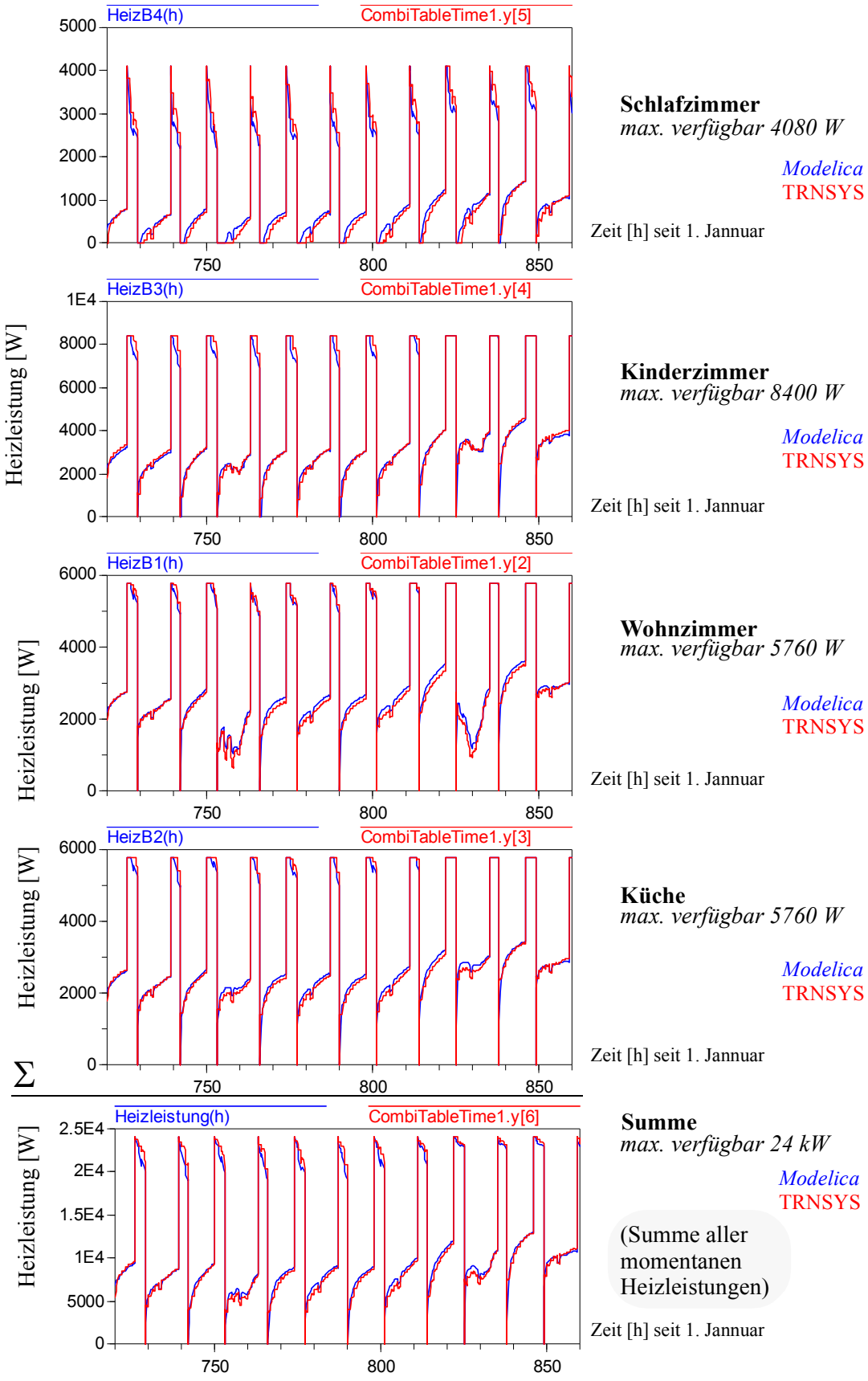


Abbildung 123: Momentan zugeführte Heizleistung der Einzelraumbeheizung, Fall B

Zunächst werden in der Abb. 121 die resultierenden Raumlufttemperaturen dargestellt. Die Raumlufttemperaturen der beheizten Räume des *Modelica*-Modells folgen weitgehend den Ergebnissen des TRNSYS-Modells. Kleinere Abweichungen treten vorwiegend während der letzten beiden sehr kalten und klaren Wintertage auf. Die Abweichungen bleiben auch in Extremfällen kleiner $0.5^{o}C$.

In Abb. 124 wird eine detaillierte Analyse durchgeführt, um mögliche Fehlerursachen zu detektieren bzw. auszuschließen. Die Abweichungen treten ausschließlich während der Phasen höherer Temperaturanforderungen auf, vorwiegend vor und nach Sonnenuntergang. Das Strahlungsmodell der solaren Einstrahlung kann also nicht verantwortlich sein. Eine falsche fiktive Himmelstemperatur (Abb. 113) würde sich vorwiegend im Dachstuhl auswirken und kann daher ebenfalls nicht die beobachteten Abweichungen verursachen. Das TRNSYS-Modell realisiert die ideale Heizung über einen Zweipunktschalter. Im *Modelica*-Ansatz wurde ein PI-Regler mit Anti-Windup-Reset eingesetzt. Das Zeitverhalten des Anstiegs ist zwar unterschiedlich, aber in den kritischen Intervallen fordern beide Modelle maximale Heizleistung an.

Auffällig ist, dass die Abweichungen nur auftreten, wenn der neue Raumtemperatursollwert innerhalb der Phase höherer Temperaturanforderung nicht erreicht werden kann. Der kontinuierliche *Modelica*-Ansatz wurde mit adaptierender Schrittweite (*Dassl*-Algorithmus) gerechnet. Die TRNSYS-Simulation ist aufgrund der 1stündigen *timebase* zur Berechnung des Übertragungsverhaltens der Wände in groben Treppenstufen gerastert. Kleinere Unterschiede in der Wiedergabe des dynamischen Verhaltens machen sich so drastisch bemerkbar, falls sich der Sollwert vor Erreichen des stationären Zustandes erneut ändert. Diese Situation liegt in Abb. 124 vor. Eine Verzerrung des dynamischen Verhaltens aufgrund einer zu geringen Schichtanzahl der Wandmodelle im *Modelica*-Ansatz kann ausgeschlossen werden. Es wurden mindestens 4 Schichten zur Beschreibung des Wandverhaltens eingesetzt. Daher ergibt der *Modelica-A*nsatz im Gegensatz zu *TRNSYS* auch im Zeitbereich $t \ll 1h$ ein realistisches Bild des dynamischen Verhaltens.

Die Übereinstimmung der Ergebnisse der beiden Simulationsansätze ist im Falle der Raumlufttemperaturen des unbeheizten Dachbodens (Abb. 121) nach einem gewissen Vorlauf sehr gut. Die Empfindungstemperaturen (Abb. 122) reagieren aufgrund des Speichervermögens der Wände träger auf die Änderungen der Raumtemperatursollwerte. Die Übereinstimmung ist im allgemeinen nahezu vollständig. Die geringen Abweichungen der Raumtemperatursollwerte am 5. und 6. Tag machen sich geringfügig in den Empfindungstemperaturen bemerkbar, die absolute Abweichung bleibt jedoch gering.

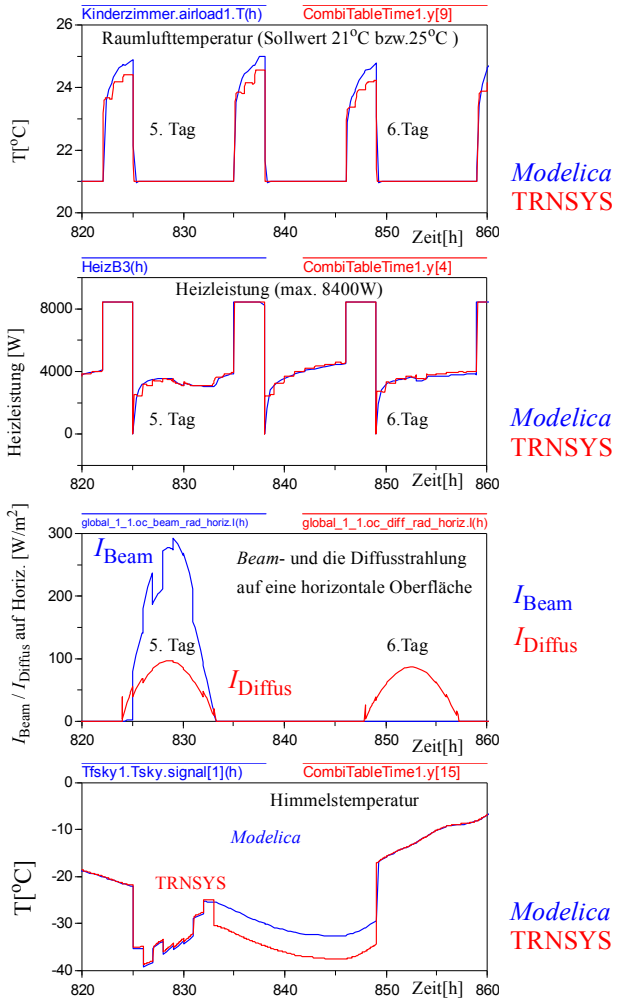


Abbildung 124: Detaillierter Vergleich zur Diskussion der Raumlufttemperaturabweichungen am 5. und 6. Tag im Kinderzimmer

Auch der Vergleich der erforderlichen Heizleistungen (Abb. 123) zeigt kaum Abweichungen. Die geringen Abweichungen am 5. und 6. Tag treten hier nicht auf, weil sowohl TRNSYS als auch *Modelica* in den kritischen Intervallen die maximale Heizleistung anfordern.

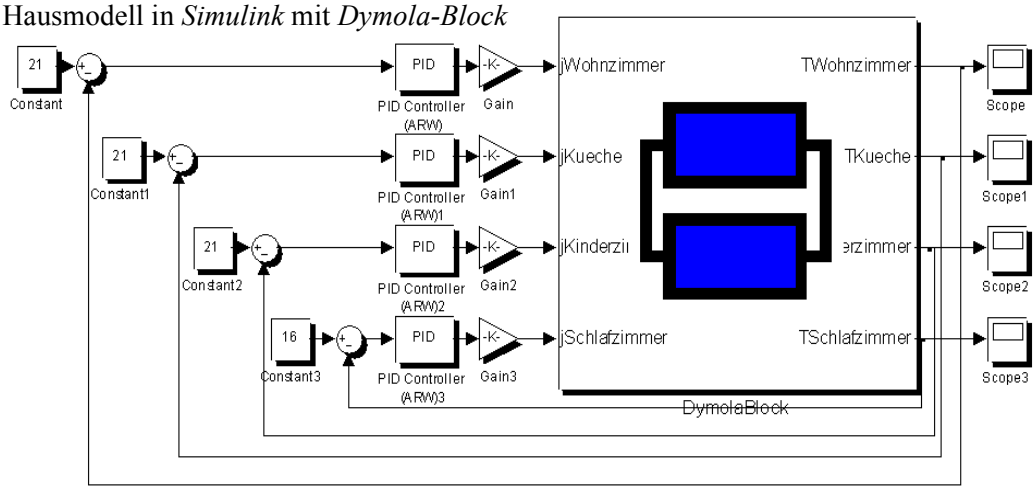
Alle Parameter basieren auf bauphysikalischen Daten. Das Zeitverhalten wie auch die absoluten Werte werden ohne Normierungsfaktoren wiedergegeben. Es erfolgte keine Berechnung von Ersatzwiderständen oder Kapazitäten entsprechend dem Beukenmodell.

6.4 Die Ankopplung Dymola Modelica

Bislang wurde zur Simulation des Beispiels *Einfamilienhaus EFHB1* ausschließlich *Dymola* eingesetzt. In gleicher Art und Weise kann das *Modelica*-Modell auch in *Simulink* eingebunden (Abb. 125) werden. Einzige Voraussetzung ist die Definition äußerer Ein- und Ausgänge.

Als Eingänge in den *Dymola-Block* dienen die Wärmeströme, als Ausgänge die Raumtemperaturen der beheizten und geregelten Räume. Die Einbindung ist weitgehend unproblematisch. Allerdings verursachen interne Tabellen beispielsweise zur Beschreibung des Nutzerverhaltens auf niedrigen Hierarchiestufen des Hausmodells Probleme. Entweder sind die Tabellen in der Hierarchie anzuheben oder über zusätzliche Eingangsschnittstellen in den *Dymola-Block* einzukoppeln. Die Simulationsgeschwindigkeit wird praktisch nicht beeinflusst, da mit *Dymola* ein compiliertes Modul, kein interpretiertes genutzt wird.

Hausmodell in *Simulink* mit *Dymola-Block*



Dymola-Block in *Modelica*

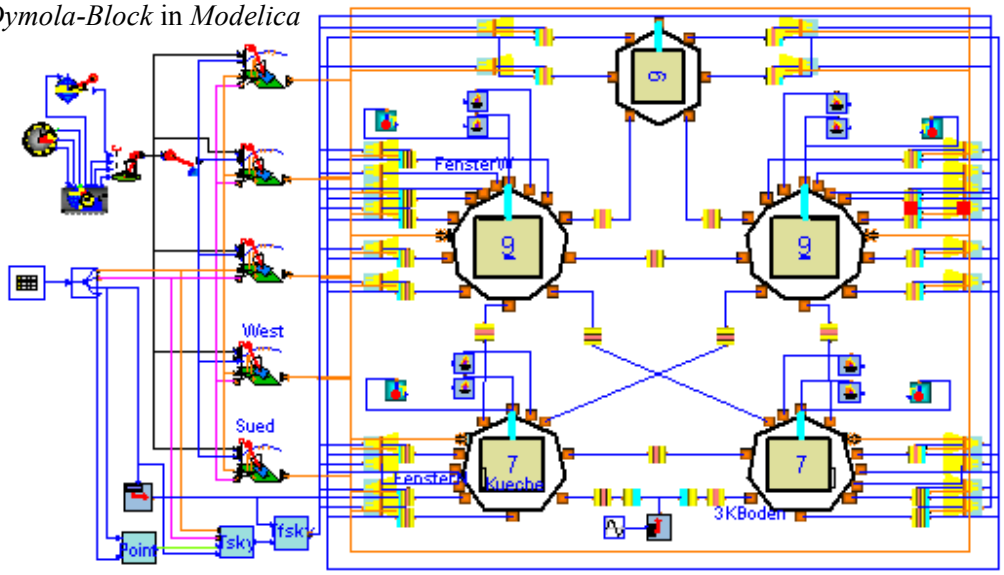


Abbildung 125: Integration des *Modelica* Einfamilienhauses von Seite 114 in *Matlab/ Simulink*

Hier sollte man darauf hinweisen, dass trotz der Nutzung des *Matlab/Simulink*-Simulators auch eine *Dymola*-Lizenz verfügbar sein muss. Stets sind in den *Matlab*-Pfad sowohl der Ordner *dymola\mfiles* als auch *\dymola\mfiles\traj* einzubinden. Ein Entwickeln und Übersetzen des *Dymola*-Modells auf Rechner I, die Übergabe des compilierten Modells auf einen Rechner II, auf dem ausschließlich *Simulink* läuft, ist ohne die entsprechenden Dateien des *Dymola*-Paketes nicht möglich.

7 Zusammenfassung

Zur Entwicklung und Planung energiesparender Gebäude, zum Entwurf geeigneter Regelungsalgorithmen benötigt man detailliertes Wissen über das thermische und energetische Verhalten eines Gebäudes, das in Wechselwirkung mit seiner Umgebung und seinen Bewohnern steht. Dies leistet ein mathematisches Modell.

Die Beschreibung großer, komplexer technischer Systeme führt zu hoch komplexen, umfangreichen mathematischen Modellen, die - zur Simulation implementiert - große Softwaresysteme ergeben. Es liegt daher nahe, Konzepte der Informatik auch in der mathematischen Modellbildung zu nutzen. Neben der Dekomposition in Teilsysteme, den Strukturierungskonzepten zur Beherrschung der Komplexität ist hier ein aktueller Forschungsgegenstand der Informatik von besonderem Interesse. Es handelt sich um die Nutzung der Wiederverwendung als methodisches Element des Softwareentwicklungsprozesses großer Systeme.

Die Wiederverwendung erprobter, d.h. verifizierter und validierter Komponenten erhöht zum einen die Effektivität, aber auch die Qualität des Neuentwurfs. Generische Konzepte stellen den Schlüssel dar, um einen möglichst hohen Wiederverwendungsanteil zu erzielen. Zunächst gilt es, die varianten und invarianten Teile eines Systems zu identifizieren. Dann ist eine geeignete Parametrierung der varianten Teile zu finden, die deren Generierung ermöglicht. Durch Analyse und Spezifikation der Vorgehensweise bei der Modellerstellung lassen sich auch Erfahrungen, Techniken und Wissen aus vorangegangenen Projekten nutzen.

Von großer Bedeutung ist die wiederverwendungsgerechte Darstellung der Modellkomponenten. Diese basiert auf vertikalen Strukturierungskonzepten wie der objektorientierten Beschreibung mit Aggregation, Vererbung und Klassenprinzip. Von zentraler Bedeutung ist jedoch die nicht berechnungskausale Beschreibungstechnik als Folge der Nutzung phänomenologischer Verknüpfungen. Es wird so möglich die Wiederverwendung schon auf der Ebene des physikalischen Ersatzmodells in den Entwicklungsprozess zu integrieren.

Zur Realisierung der theoretischen Konzepte benötigt man ein geeignetes Werkzeug, eine Simulationsumgebung. Sie übernimmt häufig wiederkehrende Aufgaben wie die Numerik, Visualisierung oder die Verwaltung einer Modellbibliothek. Durch ihre Auswahl und Anwendung nutzt man implizit Expertenwissen, Erfahrungen und in der Anwendungsdomäne etablierte Methoden. Diese Anforderungen erfüllt die objektorientierte, nicht berechnungskausale Modellbeschreibungssprache *Modelica*. Sie stellt einen interdisziplinären Standard dar, zu dem eine leistungsfähige Simulationsumgebung *(Dymola)* existiert.

Es wurde eine Modellbibliothek zur Simulation thermischen Gebäudeverhaltens in *Modelica* erstellt. Sie untergliedert sich in die Abschnitte *Gebäude-, Thermohydraulik-, Umgebungs-* und *Algorithmen*bibliothek. Die objektorientiert implementierten, nicht berechnungskausalen Modellkomponenten sind hierarchisch strukturiert. Ihre Implementierung orientiert sich am intuitiven physikalischen Verständnis des zu beschreibenden technischen Prozesses. So aggregiert ein Gebäude einzelne Räume, Fenster, Wände und diese wiederum einzelne Wandschichten.

Die Modellbibliothek wurde in exemplarischen Konfigurationen simulativ validiert. Hierzu wurde ein typisches freistehendes Einfamilienhaus des Bestandes mit 5 Räumen und seiner klimatischen Umgebung aus Bausteinen der Modellbibliothek in *Modelica* implementiert. Als Referenz diente ein TRNSYS-Gebäudemodell. TRNSYS stellt ein etabliertes und auch experimentell validiertes Tool dar, das häufig zur Gebäude- und Anlagenplanung eingesetzt wird. Es steht nun eine praktisch anwendbare Modellbibliothek zur dynamischen Simulation thermischen Gebäudeverhaltens in *Modelica* zur Verfügung.

8 Literatur

A

[Agu-02]
Agustina, S.: Entwicklung eines mathematischen Modells zur Simulation des thermischen Verhaltens eines Einfamilienhauses in Modelica, Masterarbeit, Lehrstuhl für Automatisierungstechnik, FB EIT, Universität Kaiserslautern (2002)

[And-94]
Andersson M.: Object-Oriented Modeling and Simulation of Hybrid Systems. Dissertation, Department of Automatic Control, Lund Institute of Technology, (1994).

[And-95]
Andersson M.: OmSim and Omola, Tutorial and User's guide, Department of Automatic Control, Lund Institute of Technology, (1995)

[Aue-94]
Auer T.: TRNSYS: Modul zur Berechnung der Himmelstemperatur als Funktion des Bedeckungsgrades, der Luftfeuchte, Lufttemperatur, Stuttgart: Transsolar Energietechnik (1994)

[Ave et al.-98]
Avenhaus J., Gotzhein R., Härder T., Litz L., Madlener K., Nehmer J., Richter M., Ritter N., Rombach D., Schürmann B., Zimmermann G.: Entwicklung großer Systeme mit generischen Methoden - Eine Übersicht über den Sonderforschungsbereich 501. Informatik Forsch. Entw. 13, S. 227-234 (1998); *http://www.SFB501.uni-kl.de/*

B

[BaLe-96]
Balters E., Lehmann H.: Planungsinstrument Gebäudeautomation, in Solararchitektur für Europa, Hrsg.: Schneider, A. und focus-film, Birkhäuser Verlag Berlin 1996 S.158-161

[Bas-90]
Basili V.R.: Viewing Maintenance as Reuse-Oriented Software Development, IEEE Software, January (1990) pp.19-25

[BaWi-00]
Bachmann, B. und Wiesman, H.: Objektorientierte Modellierung Physikalischer Systeme, Teil 15: Modellierung von Elektroenergiesystemen. at Automatisierungstechnik, 48, 8, pp. A57 - A60 (2000).

[Bea-99]
Beater P.: Entwurf hydraulischer Maschinen – Modellbildung, Stabilitätsanalyse und Simulation hydrostatischer Antriebe und Steuerungen. Springer, Heidelberg.(1999)

[Bea-00]
Beater P: Objektorientierte Modellierung Physikalischer Systeme, Teil 10: Modellierung hydraulischer Systeme. at Automatisierungstechnik, 48, 1 (2000) pp. A37-A40.

[Beu-36]
Beuken, D. L.: Wärmeverluste bei periodisch betriebenen Öfen, Dissertation Freiburg 1936

[BiBr-97]
Biran A.B., Breiner M.M.G.: *Matlab* für Ingenieure: Systematische und praxisnahe Einführung, Addison-Wesley-Longman, Bonn (1997)

[BiKl-95]
Biersack M. and Klose M.: Issues in Design and Implementation of Simulation Systems. In A. Sydow, editor, Proceedings of the 5th International IMACS-Symposium on Systems Analysis and Simulation (SAS'95),. vol 18-19, p. 497-502 of Systems Analysis Modeling Simulation. Gordon and Breach Publishers (1995).

[BrBr-97)
Breunes A.P.J. and Broenink J.F.: Modeling mechatronic systems using the SIDOPS+ language, Proceedings of ICBGM'97, 3rd International Conference on Bond Graph Modeling and Simulation, Phoenix, Arizona, SCS Publishing, San Diego, California, Simulation Series, Vol.29, No.1, pp 301-306 (1997)

[BrDr-95]
Bröhl, A.-P., Dröschel W.: Das V-Modell. München, Wien: Oldenburg-Verlag, 1995

[Büh-96]
Bühring, A.: Rahmenbedingungenzukünftiger Energiesysteme im Wohnbereich; Entwicklung einer Testumgebung für Heizsysteme unter Verwendung des dynamischen Simulationsprogrammes TRNSYS. Diplomarbeit an der TUHH, (1996), *zitiert in [KiSi-01]*

C

[Car-99]
Solar-Institute Juelich: Carnot-Blockset, Version 1.0, User's Guide,
Scientific Computers GmbH, Aachen 1991

[Car-02]
Solar-Institut Jülich: Carnot-Blockset Produktinformationen,
http://www.internationaltec.com/products/carnot.html,
Scientific Computers GmbH, Aachen 2002

[CeEl-93]
Cellier F.E. und Elmqvist H.: Automated Formula Manipulation Supports Object-Oriented Continous-System Modeling. IEEE Control Systems 13, 28-38 (1993)

[Cel-91]
Cellier F.E.: Continous System Modeling. Heidelberg: Springer Verlag, 1991

[Cla-02]
Cladera Bohigas R.: Entwicklung einer generischen Modelica-Bibliothek zur Simulation solarer Einstrahlung bei terrestrischen Anwendungen, Diplomarbeit, Lehrstuhl für Automatisierungstechnik, FB EIT, Universität Kaiserslautern (2002)

[CSS-00a,b]
Clauß, C. Schneider, A., Schwarz, P.: Objektorientierte Modellierung Physikalischer Systeme, Teil 13 + 14: Modellierung von elektronischen Schaltungen. at Automatisierungstechnik, 48, 5/7 (2000) pp. A49 - A56.

D

[DuBe-74]
Duffie J.A., Beckman W.A.: Solar Energy thermal process, Wiley, New York (1974)

[Dym-02]
Dymola. Lund/Schweden: Fa. Dynasim, 1999; *http://www.dynasim.se*

E

[Eic-94]
Eicke-Hennig W.: Empirische Überprüfung der Möglichkeiten und Kosten, im Gebäudebestand und bei Neubauten Energie einzusparen und die Energieeffiziens zu steigern, Endbericht für die „Deutsch. Bundesstift. Umwelt“, IWU Darmstadt (1994)

[EiSi-97]
Eicke-Hennig W., Siepe B.: Die Heizenergie-Einsparmöglichkeiten durch Verbesserung des Wärmeschutzes typischer hessischer Wohngebäude, IWU Darmstadt (1997)

[Elm-78]
Elmqvist H.: A structures Model Language for Large Continous Systems, Dissertation, Lund Institute of Technology, Sweden, Department of Automatic Control (1978)

[Elm-93]
Elmqvist H.: Object-Oriented Modeling and Automatic Formula Manipulation in *Dymola*. SIMS 93, Scandinavian Simulation Society, Konsberg/Norway, June 9-11, (1993) S.1-10

[ElMa-97]
Elmqvist H., Mattson S.E.: An introduction to the physical Modeling Language. *Modelica*. Proc. 9th Europ. Sim. Symp., Passau 1-5 (1997)

[ElOt-94]
Elmqvist H., Otter M.: Methods for tearing systems of equations in object-oriented modeling, ESM’94, European Simulation Multiconference, Barcelona/Spain June 1-3 (1994)

[Equ-02]
Homepage der Fa. Equa: *http://www.equa.se,* Equa Simulation technology group, Sundbyberg, Sweden (2002)

F

[Fei-94]
Feist W.: Thermische Gebäudesimulation Kritische Prüfung unterschiedlicher Modellansätze, Verlag C.F. Müller, Heidelberg (1994) pp.135

[Fel-02]
Felgner F.: Entwicklung einer Fuzzy-Control-Bibliothek in Modelica, Diplomarbeit, Lehrstuhl für Automatisierungstechnik, FB EIT, Universität Kaiserslautern (2002)

[Fel et al. 02]
Felgner F., Agustina S., Cladera Bohigas R., Merz R., Litz L.: Simulation of thermal building behaviour in modelica, in Proc. of the 2nd. **International Modelica Conference**, March 18-19 DLR e.V. Oberpfaffenhofen (2002) pp. 147-154

[FeMe-01]
Felgner F.; Merz R.: Thermohydraulische Simulationsmodelle für Heizungsanlagen, **ASIM 2001**, Veran.: Gesellschaft für Informatik (GI), SCS Europe, IMACS, EUROSIM, 15.Sym. Sim.tech. Paderborn 2001

[FEV-93]
Fritzson P., Engelson V., Viklund L.: Variant handling, inheritance and composition in the ObjectMath computer algebra environment. In Alfonso Miola, ed.: Design and Implementation of Symbolic Computation Systems, vol. 722 Lec. Notes in Comp. Sci, Springer-Verlag, Heidelberg (1993) pp. 145-160

[Föl-93]
Föllinger O.: Lineare Abtastsysteme, Methoden der Regelungs- und Automatisierungstechnik, R. Oldenbourg Verlag München (1993)

[Föl-94]
Föllinger O.: Regelungstechnik - Einführung in die Methoden und ihre Anwendung, Hüthig Verlag (1994)

[Föl-00]
Föllinger O.: Laplace- Fourier- und Z-Transformation, Hüthig Verlag (2000)

G

[Gey et al.-00]
Geyer L., Kronenburg M., Merz R., Schürmann B.: Technical Report **SFB 501**: Domain Modeling in the SFB 501 Report 07 / Fachbereich Elektro- und Informationstechnik, Universität Kaiserslautern (2000)

[Glü-91]
Glück B.: Zustands- und Stoffwerte (Wasser, Dampf, Luft), Verbrennungsrechnung, 2.erw. Aufl. Berlin: Verlag für Bauwesen (1991)

H

[Hil et.al.-01]
Hiller M., Holst S., Knirsch A., Schuler M.: TRNSYS 15- A Simulation Tool for Innovative Concepts, Proc. of 7th. Int. IBPSA Conf. 13.-15.th Aug., Rio de Janeiro, Brazil (2001), pp. 419-422

[HKS-97]
Hoffmann, H.-J., Katscher W., Stein G.: Energiestrategien für den Klimaschutz in Deutschland. Das IKARUS-Projekt des BMBF. Zusammenfassender Endbericht. Hrsg. v. Forschungszentrum Jülich GmbH, Programmgruppe Technologie-Folgenforschung, Jülich (1997)

[HMS-89]
Hering E., Martin R., Stoher M.: Physik für Ingenieure, 3. Aufl., VDI Verlag, Düsseldorf (1989)

[Hof-98]
Hoffmann, J.: *Matlab* und *Simulink*: Beispielorientierte Einführung in die Simulation dynamischer Systeme, Addison-Wesley-Longman, Bonn (1998)

J

[JeBo-97]
Jeandel, A., Boudaud, F.: Physical System Modelling Languages: from ALLAN to *Modelica*. Building Simulation '97, IBPSA Conference, Prague,(1997.)

[JuNe-95]
Judkoff R., Neymark J.: International Energy Agency, Building Energy Simulation Test „BESTEST“, National Renewable Energy Laboratory, Colorado (1995)

K

[Kel-97]
Keller B., Klimagerechtes Bauen: Grundlagen, Dimensionierungen, Beispiele, B.G. Teubner Stuttgart (1997)

[KFS-95]
Kloas M., Friesen V. Simons M.: *Smile* - a Simulation Environment for Energy Systems. Int. IMACS-Symp. on System Analysis and Simulation (SAS 95). Gordon Beach Publishing House 1995

[KiSi-01]
Kienzlen K. und da Silva P.: Das Haus im Entwicklungslabor, TAB (Technik am Bau), Bertelsmann Fachzeitschriften GmbH, Gütersloh, 10 (2001) S. 35-39

[Kle et al.-96]
Klein, S.A., Duffie, J.A., Mitchell, J.C., Kummer, J.P., Thornton, J.W., Beckman W.A., Duffie, N.A., Braun J.E., Urban, R.R., Blair, N.J., Mitchell, J.W., Freeman T.L., Evans B.L., Fiksel A.: TRNSYS, a transient system simulation program, Solar Energy Laboratory University of Wisconsin, Madison 1996

[Kri-98]
Kriener, K.: Simulation und dynamischer Test von zukünftigen Heizsystemen, Diplomarbeit am Institut für Thermodynamik und Technische Gebäudeausrüstung der TU Dresden (1998), *zitiert in [KiSi-01]*

L

[Lec-92]
Lechner, T.: Berechnung des Wärmestroms durch ebene geschichtete Wände, Mathematische und physikalische Grundlage der Transfer Function Methode, Inst. für Thermodynamik und Wärmetechnik, Universität Stuttgart, Stuttgart (1992) p. 1-29

[LEK-97]
Lechner, T., Ebert, T., Keupp, K.:Thermische Gebäudesimulation: Der Computer als Planungswerkzeug; bai intern 10(1997) S.211-214

[LiFr-99]
Litz, L., Frey G.: Methoden und Werkzeuge zum industriellen Steuerungsentwurf - Historie, Stand, Ausblick, at Automatisierungstechnik, 47, 4 (1999) pp. 145-156

[LiKö-95]
Litz L., König H.: Flexibler Simulator für konventionell und fußbodenbeheizte Häuser. atp 37 (1995), S. 3641

[Lit-83]
Litz, L.: Dezentrale Regelung. Oldenbourg Verlag, München 1983

[Lit-01a]
Litz L.: Vorlesung im SS 01: Grundlagen der Automatisierung, FB EIT, Universität Kaiserslautern (2001).

[Lit-01b]
Litz L.: Vorlesung im WS 01/02: Modellbildung und Identifikation, FB EIT, Universität Kaiserslautern (2001).

[Loh-95]
Lohmeyer G.: Praktische Bauphysik: Eine Einführung mit Berechnungsbeispielen, Teubner-Verlag, Stuttgart (1995)

[Lun-96]
Lunze J.: Regelungstechnik 1, Systemtheoretische Grundlagen, Analyse und Entwurf einschleifiger Regelungen, Springer Heidelberg (1996)

[Lun-97]
Lunze J.: Regelungstechnik 2, Mehrgrößensysteme, Digitale Regelung, Springer Heidelberg (1997)

M

[Mat-02]
Matlab/Simulink - Produktinformation, Mathworks Deutschland GmbH, Aachen 2002
http://www.mathworks.de

[MeLi-99]
Merz R., Litz L.: Technical Report **SFB 501**- Anwendung generischer Methoden bei der objektorientierten Modellierung großer, komplexer dynamischer Systeme 07/99, Fachbereich Elektro- und Informationstechnik, Universität Kaiserslautern, 1999

[MeLi-00a]
Merz R., Litz L.: O.O: math. Mod., generische Methoden bei komplexen dynamischen Systemen. Inf.-Spek. 23 (2000), S. 90-99

[MeLi-00b]
Merz R., Litz L.: Technical Report **SFB 501**- Regelungstechnisches Tutorial 6/00, Fachbereich Elektro- und Informationstechnik, Universität Kaiserslautern, 2000

[MeLi-01]
Merz R., Litz L.: Objektorientierte mathematische Modellbildung zur Simulation thermischen Gebäudeverhaltens, **ASIM 2001**, Veran.: Gesellschaft für Informatik (GI), SCS Europe, IMACS, EUROSIM, 15.Sym. Sim. tech. Paderborn 2001

[Mer-01]
Merz R.: Technical Report **SFB 501**- Messung lokaler Klimadaten X/01, Fachbereich Elektro- und Informationstechnik, Universität Kaiserslautern, 2001

[Mod-02]
Modelica - a unified object-oriented language for physical system modeling.
The *Modelica* Design Group 2002: *http://www.modelica.org,* (siehe auch *http:/ www.dynasim.se/Modelica);*
ausführliche Publikationsliste: *http://www.modelica.org/publications.shtml*

N

[NyBa-01]
Nytsch-Geusen C., Bartsch G.: An object oriented multizone thermal building model based on the simulation environment smile, in Proc. 7th. Int. IBPSA Conf., Rio de Janeiro, Brazil, 13.-14. Aug. (2001) p. 411-418

O

[OEM-99a,b]
Otter M., Elmqvist H., Mattsson S.E.: Objektorientierte Modellierung Physikalischer Systeme, Teil 7 + 8 *Modelica* - Hybride Systeme. at Automatisierungstechnik, 47, 9 (1999) pp. A25-A28 und 11(1999) pp. A29-A32

[OtBa-99a,b]
Otter M., Bachmann B.: Objektorientierte Modellierung Physikalischer Systeme, Teil 5 + 6: Singuläre Systeme. at Automatisierungstechnik, 47, 5 und 6 (1999) pp. A17-A24.

[OtSc-99]
Otter M., Schlegel C.: Objektorientierte Modellierung Physikalischer Systeme, Teil 9: Modellierung von Antriebssträngen. at Automatisierungstechnik, 47, 12 (1999) pp. A33-A36

[Ott-99a]
Otter M.: Objektorientierte Modellierung Physikalischer Systeme, Teil 1: Übersicht. at Automatisierungstechnik, 47, 1 (1999) pp. A1-A4.

[Ott-99b]
Otter M.: Objektorientierte Modellierung Physikalischer Systeme, Teil 2: *Modelica* - Kontinuierliche Systeme. at Automatisierungstechnik, 47, 2 (1999) pp. A5-A8.

[Ott-99c]
Otter M.: Objektorientierte Modellierung Physikalischer Systeme, Teil 3: Komponenten-Schnittstellen. at Automatisierungstechnik, 47, 3 (1999) pp. A9-A12.

[Ott-99d]
Otter M.: Objektorientierte Modellierung Physikalischer Systeme, Teil 4: Transformationsalgorithmen. at Automatisierungstechnik, 47, 4 (1999) pp. A13-A16.

[Ott-00]
Otter M.: Objektorientierte Modellierung Physikalischer Systeme, Teil 17:. Modellierung mechanischer Systeme. at Automatisierungstechnik, 48, 12 (2000) pp. A65-A68.

P

[Pan-88]
Pantelides C.C.: The consistent initialization of differential-algebraic systems. SIAM Journal of Scientific and Statistical Computing, (1988), S.213-231

[Pan-99]
Panreck K.: Systembeschreibung zur Modellierung komplexer Systeme, at 4, S.157-163 (1999)

[PJD-94]
Panreck K., Jahnich M., Dörrscheidt F.: Verwendung differential-algebraischer Gleichungen zur verkoppelungsorientierten Modellierung komplexer Prozesse, at 6, S.239-247 (1994)

R

[RoVe-95]
Rombach D., Verlage M.,: Directions in Software process research, in: Advances in computers, Acad. Press. vol. 41(1995) pp.1-59

[RoZi-97]
Rouvel L. und Zimmermann F.: Ein regelungstechnisches Modell zur Beschreibung des thermisch dynamischen Raumverhaltens. Heizung Lüftung / Klima Haustechnik Bd.48 (1997) Nr. 10 + Bd.48 (1997) Nr. 12, Bd. 49 (1998) Nr. 1

[RSS-95]
Recknagel H., Sprenger E., Schramek E.R. (Hrsg.): Taschenbuch für Heizung und Klimatechnik 94/95, 67. Auflage, München 1995.

S

[Sch-96]
Schneider A.: Solararchitektur für Europa, Birkhäuser Verlag, Berlin (1996)

[See-00]
Seel O.: Objektorientierte mathematische Modellbildung des thermischen Gebäudeverhaltens, Diplomarbeit, Lehrstuhl für Automatisierungstechnik, FB EIT, Universität Kaiserslautern (2000)

[SeGu-00]
Seemann J., v. Gudenberg J.W.: Software- Entwurf mit UML, Springer Verlag Heidelberg (2000)

[SiMe-01]
Sitompul E., Merz R.: Anwendung objektorientierter Entwurfsmethoden zur generischen Entwicklung von Regelungsalgorithmen in der Anwendungsdomäne Gebäudeautomation, **ASIM 2001**, Veran.: Gesellschaft für Informatik (GI), SCS Europe, IMACS, EUROSIM, 15.Sym. Sim.tech. Paderborn 2001

[Sit-00]
Sitompul E.: Anwendung objektorientierter Entwurfsmethoden zur generischen Entwicklung von Regelungsalgorithmen, Masterarbeit, Lehrstuhl für Automatisierungstechnik, FB EIT, Universität Kaiserslautern (2000)

[SpMe-01]
Sprengard C. Merz R.: Simulation des energetischen und thermischen Verhaltens eines Niedrigenergiehauses mit dem Gebäudesimulationsprogramm TRNSYS ASIM 2001, 15.Sym. Sim.tech. Paderborn 2001

[StMi-71]
Stepheson D.G., Mitalas G.P.: Calculation of Heat Conduction Transfer Functions for Multi-layer Slabs, ASHRAE Annual Meeting, Washington D. C. No. 2202 (1971) p. 117-126

T

[Til-01]
Tiller M.: Introduction to physical Modeling with *Modelica*, Kluwer Academic Publishers, London (2001)

[TKE-97]
Tummescheit H., Klose M., Ernst T.: *Modelica* and *Smile* - A Case Study Applying Object-oriented Concepts for Multi-facet Modeling. Passau/ Germany: ESS'97 Europ. Sim. Symp., October 19-22, (1997)

[Trn-02]
TRNSYS, *www.transsolar.de*, Homepage der Fa. Transsolar Energietechnik GmbH. Stuttgart 2001

[TuEb-98]
Tummescheit H., Eborn J.: Design of a Thermo-Hydraulic Model Library in *Modelica*. The 12th European Simulation Multiconference, ESM'98, Manchester, UK (1998), siehe auch: *http://www.control.lth.se/~hubertus/ThermoFluid/*

[Tum-00a,b]
Tummescheit H.: Objektorientierte Modellierung Physikalischer Systeme, Teil 11 + 12: Modellierung von thermohydraulischen Prozessen. at Automatisierungstechnik, 48, 2/4 (2000) pp. A41 - A48.

[Tum-00c]
Tummescheit H.: Development of a *Modelica* Base Library for Modeling of Thermo-hydraulic Systems, *Modelica* Workshop 2000, Lund University, Lund, Sweden (2000)

[TuTi-00]
Tummescheit H., Tiller M.: Objektorientierte Modellierung Physikalischer Systeme, Teil 16: Modellierung von Ottomotoren. at Automatisierungstechnik, 48, 11, pp. A61 - A64 (2000).

V

[VDI-6020]
VDI Richtlinie 6020: Anforderung an Rechenverfahren zur Gebäude- und Anlagensimulation, VDI, Düsseldorf, ersch. bei Beuth Verlag GmbH, Berlin (2001)

[ViFr-92]
Viklund L. and Fritzson P.: An object-oriented language for symbolic computation - applied to machine element analysis. In Wang P.S., ed.: Proc. of the Int. Symp. on Symbolic and Algebraic Computation, ACM Press, New York (1992) pp 397-405

W

[Wei-99]
Weiss D.: A Software Product-Line Engineering: A Family-Based Software Development Process, Addison Wesley, (1999)

[WHS-00a]
Wemhöner C., Hafner B., Schwarzer K: Simulation of solar thermal systems with CARNOT Blockset in the environment *Matlab/Simulink*, Eurosun 2000, Kopenhagen

[WHS-00b]
Wemhöner C., Hafner B., Schwarzer K.: Berechnung von Solaranlagen mit CARNOT unter *Matlab/Simulink*, 10. Symposium Thermische Solarenergie, OTTI Technologiekolleg, 2000, Kloster Banz

[WHS-01]
Wittwer C., Hube H., Schossig P.: ColSim - A New Simulation Environment for Complex system analysis and controllers. Proc. of 7th. Int. IBPSA Conf. 13.-15.th Aug., Rio de Janeiro, Brazil (2001), pp. 237-244

Z

[Zim-01]
Zimmermann G.: A new approach to building simulation based on communicating objects. Proc. of 7th. Int. IBPSA Conf. 13.-15.th Aug., Rio de Janeiro, Brazil (2001), pp. 707-714

9 Anhang

9.1 Die physikalischen Grundlagen des Beuken-Modells

Im folgenden sollen die physikalischen Grundlagen des Beuken-Modells [Beu-36] *(z.B. in [RoZi-97])* zur Behandlung des eindimensionalen dynamischen Wärmeleitungsverhaltens einer Wand vorgestellt werden. Das Verfahren ähnelt der elektrischen Netzwerkanalyse. Die Darstellung folgt Feist [Fei-94].

Der Ausgangspunkt ist die eindimensionale dynamische Wärmeleitungsgleichung (1).

$$\lambda \cdot \frac{\partial^2}{\partial x^2} T(x,t) = c\rho \cdot \frac{\partial}{\partial t} T(x,t) \qquad (1).$$

λ	Wärmeleitfähigkeit
c	spez. Wärmekapazität
ρ	Dichte

Die gesuchte Lösung muss am äußeren Rand $(x = 0)$ und inneren Rand $(x = d_{\text{Wand}})$ der Wand die Randbedingungen (2) erfüllen.

$$\lambda \cdot \frac{\partial T}{\partial x} = \alpha_a (T - T_a)$$
$$-\lambda \cdot \frac{\partial T}{\partial x} = \alpha_i (T - T_i) \qquad (2).$$

α_i	innere Wärmeübergangszahl
α_a	äußere Wärmeübergangszahl

Das Beuken-Modell zerlegt die feste Wand in Schichten. Hierbei wird keine äquidistante Zerlegung vorausgesetzt, auch müssen die Materialkonstanten nicht stückweise konstant sein. Die Zerlegung wird in Abb. 126 gezeigt.

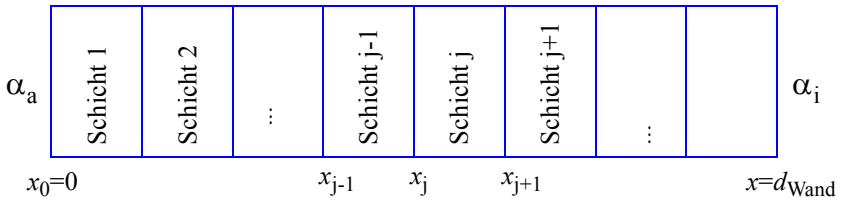


Abbildung 126: Zerlegung der Wand in einzelne Schichten

Dies entspricht einer Ortsdiskretisierung der eindimensionalen Wärmeleitungsgleichung (1) an den Stützstellen x_j. Im folgenden wird die Schreibweise $T_j(t) = T(x_j, t)$ genutzt. Unter Anwendung des zentralen Differenzenquotienten erhält man die Differenzengleichung (3).

$$\frac{2}{\Delta x_{j-1} + \Delta x_j} \left(\frac{\lambda_{j-1}}{\Delta x_{j-1}} (T_{j-1}(t) - T_j(t)) - \frac{\lambda_j}{\Delta x_j} (T_j(t) - T_{j+1}(t)) \right) = c\rho_j \cdot \frac{\partial}{\partial t} T_j(t) \qquad (3).$$

$$\text{mit} \quad \Delta x_j = x_{j+1} - x_j \quad \text{für } 2 \le j \le n\text{-}1$$

Durch geeignete Wahl [(4), (5)] von in den einzelnen Schichten konstanten Ersatzwerten der Materialkonstanten $\lambda_j, c\rho_j$ gelangt man zu einer physikalisch interpretierbaren Darstellung (6).

$$\frac{1}{\lambda_j} = \frac{1}{\Delta x_j}\int_{x_j}^{x_{j+1}} \frac{1}{\lambda(x)}dx \quad \text{für } 1 \le j \le n\text{-}1 \tag{4}$$

$$c\rho_j = \frac{2}{\Delta x_{j-1} + \Delta x_j}\int_{x_{j-1}+\frac{\Delta x_{j-1}}{2}}^{x_j+\frac{\Delta x_j}{2}} c(x)\rho(x)dx \quad \text{für } 2 \le j \le n\text{-}1 \tag{5}$$

$$R_{j-1} = \frac{\Delta x_{j-1}}{\lambda_{j-1}} \qquad R_j = \frac{\Delta x_j}{\lambda_j} \qquad C_j = \frac{\Delta x_{j-1} + \Delta x_j}{2} c\rho_j \tag{6}$$

- R_{j-1}, bzw. R_j können als Wärmedurchgangswiderstände in den Intervallen $[x_{j-1}, x_j]$ bzw. $[x_j, x_{j+1}]$ aufgefasst werden.
- C_j kann als flächenbezogene Wärmekapazität im Intervall $[x_{j-1} - \Delta x_{j-1}/2\ ,\ x_{j-1} + \Delta x_{j-1}/2]$ aufgefasst werden.

Nutzt man die Definitionen von R_{j-1}, R_j und C_j in der Differenzengleichung (3), erhält man die Differentialgleichung (7) eines RC-Netzwerkes (siehe Abb. 127).

$$\frac{1}{R_{j-1}}(T_{j-1}(t) - T_j(t)) - \frac{1}{R_j}(T_j(t) - T_{j+1}(t)) = C_j \cdot \frac{\partial}{\partial t}T_j(t) \tag{7}$$

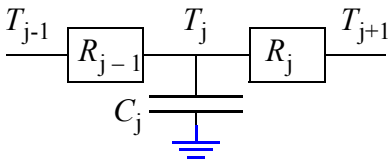


Abbildung 127: RC-Netzwerk

Die Randbedingungen (2) bezüglich der Oberflächentemperaturen T_1 und T_n sind erfüllt, wenn $T_0(t) = T_a(t)$ und $T_{n+1}(t) = T_i(t)$ als Medientemperaturen eingefügt werden und die Widerstände der Randschichten entsprechend gewählt werden (8).

$$R_0 = \frac{1}{\alpha_a} \quad R_1 = \frac{\Delta x_1}{\lambda_1} \quad \ldots\ldots\ldots\ldots \quad R_{n-1} = \frac{\Delta x_{n-1}}{\lambda_{n-1}} \quad R_n = \frac{1}{\alpha_i} \tag{8}$$

$$C_1 = \int_0^{\frac{\Delta x_1}{2}} c(x)\rho(x)dx \quad \ldots\ldots\ldots\ldots \quad C_n = \int_{x_{n-1}+\frac{\Delta x_{n-1}}{2}}^{x_n} c(x)\rho(x)dx$$

Für eine äußere harmonische Störung lässt sich ein System wie (7) mit einem Ansatz der Form $T_j(t) = D_j e^{1\omega t}$ rekursiv exakt lösen (Abb. 128). Die rekursive Lösung wird im allgemeinen als Beukenmodell bezeichnet. Das Konzept wurde vorwiegend auf Analogrechnern genutzt.

1. Start $Z_1 = C_1 i\omega + \alpha_a + 1/R_1$

Rekursion $\rightarrow Z_j = C_j i\omega + 1/R_{j-1} + 1/R_j - 1/(R_{j-1}^2 Z_{j-1})$

2. Start $\zeta_1 = -\alpha_a T_a$

Rekursion $\rightarrow \zeta_j = -\zeta_{j-1}/(R_{j-1} Z_{j-1})$ für $j = 2$ bis n_{Knot}

3. Start $D_{n_{Knot}} = \zeta_{n_{Knot}}/Z_{n_{Knot}}$

Rückwärts-Rekursion $\rightarrow D_j = (-D_{j+1}/R_j + \zeta_j)/Z_j$ für $j = n_{Knot} - 1$ bis 1

Abbildung 128: Rekursionsvorschrift nach [Fei-94]

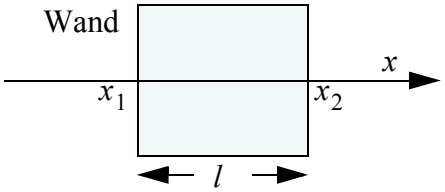
9.2 Die mathematischen Grundlagen der Transferfunktionsmethode

Im folgenden sollen die physikalischen Grundlagen der von TRNSYS genutzten Transferfunktionsmethode zur Berechnung des dynamischen Wärmeleitungsverhaltens von geschichteten Wänden beschrieben werden [Lec-92, StMi-71]. Die Darstellung folgt Lechner [Lec-92].

Die Transferfunktionsmethode besteht aus einer Kombination von Laplace- und z-Transformation [Föl-00]. Die Laplace-Transformation wird zur Lösung kontinuierlicher, partieller Differentialgleichungen eingesetzt. Die z-Transformation erlaubt die zeitliche Diskretisierung der kontinuierlichen Funktionen.

9.2.1 Die Laplace-Transformation

Im folgenden Beispiel (Abb. 129) soll die Ausbreitung eines eindimensionalen Wärmestroms längs der x-Achse durch eine Wandschicht betrachtet werden. Die Lösung der eindimensionalen Wärmeleitungsgleichung im Laplace-Bereich führt zu einer linearen Übertragungsmatrix *H(s)*.



λ	Wärmeleitfähigkeit
c	spez. Wärmekapazität
ρ	Dichte
$l=x_1-x_2$	Dicke

Abbildung 129: Eindimensionale Wärmeleitung durch eine homogene Wandschicht

Die eindimensionale Wärmeleitungsgleichung [Gleichung (1) auf S.145] stellt eine partielle kontinuierliche Differentialgleichung dar. Zu ihrer Lösung wird eine Laplace-Transformation vorgenommen. Die relevanten Größen, die eindimensionale Wärmeleitungsgleichung, die Bestimmungsgleichung der Wärmestromdichten sowie ihre Laplace-Transformierten sind in Abb. 130 aufgelistet. Zur Vereinfachung werden die Temperaturen $\theta(x, t)$ auf einen festen Anfangswert ϑ_0 bezogen, d.h. es gilt $\theta(x, t) = \vartheta(x, t) - \vartheta_0$.

	Zeitbereich	Laplace-Bereich
Temperatur	$\theta(x, t)$	$\Theta(x, s)$
Wärmestrom-dichte	$\dot{q}(x, t)$	$\dot{Q}(x, s)$
eindim. Wärme-leitungsgleichung	$\lambda \frac{\partial^2}{\partial x^2}\theta = c\rho\frac{\partial\theta}{\partial t}$	$\lambda\frac{\partial^2}{\partial x^2}\Theta = c\rho s\Theta$
Wärmestrom-dichte	$\dot{q} = -\lambda\frac{\partial\theta}{\partial x}$	$\dot{Q} = -\lambda\frac{\partial\Theta}{\partial x}$

Abbildung 130: Laplace -Transformation der relevanten Größen und Gleichungen

Abb. 131 zeigt die allgemeine Lösung des Problems im Laplace-Bereich.

$$\Theta(x, s) = C_1(x, \xi)e^{x\xi/l} + C_2(x, \xi)e^{-x\xi/l} \qquad \text{mit } \xi = \sqrt{s \cdot l^2 c\rho/\lambda} \qquad (1).$$

$$\dot{Q}(x, s) = -\frac{\xi}{R}(C_1(x, \xi)e^{x\xi/l} - C_2(x, \xi)e^{-x\xi/l}) \qquad \text{mit } R = \lambda/l \qquad (2).$$

Abbildung 131: Allgemeine Lösung des Problems im Laplace-Bereich

Von Interesse für das Wandverhalten sind vorwiegend die Temperaturen und Wärmeströme an den Wandoberflächen, also

$\Theta_1(s) = \Theta(x_1, s)$, $\Theta_2(s) = \Theta(x_2, s)$,

$\dot{Q}_1(s) = \dot{Q}(x_1, s)$ und $\dot{Q}_2(s) = \dot{Q}(x_2, s)$.

Durch Einsetzen der Ortskoordinaten der Wandoberflächen ergeben sich die in Abb. 132 gezeigten linearen, algebraischen Bestimmungsgleichungen im Laplace-Bereich, welche mit Hilfe einer Übertragungsmatrix $\underline{H(s)}$ vektoriell kompakt geschrieben werden können. Man hat so eine modulare Beschreibungsform geschaffen, die es erlaubt, das Verhalten mehrschichtiger Wände auf eine Matrizenmultiplikation zurückzuführen (Abb. 133).

$$\left.\begin{aligned}\Theta_1 &= \cosh\xi\cdot\Theta_2 + \frac{R}{\xi}\sinh\xi\cdot\dot{Q}_2\\ \dot{Q}_1 &= \frac{\xi}{R}\sinh\xi\cdot\Theta_2 + \cosh\xi\cdot\dot{Q}_2\end{aligned}\right\} \Rightarrow \begin{bmatrix}\Theta_1\\ \dot{Q}_1\end{bmatrix} = \begin{bmatrix}\cosh\xi & \frac{R}{\xi}\sinh\xi\\ \frac{\xi}{R}\sinh\xi & \cosh\xi\end{bmatrix}\cdot\begin{bmatrix}\Theta_2\\ \dot{Q}_2\end{bmatrix} = \underline{H}\cdot\begin{bmatrix}\Theta_2\\ \dot{Q}_2\end{bmatrix}$$

$$\text{mit}\quad \underline{H} = \begin{bmatrix}A & B\\ C & D\end{bmatrix} = \begin{bmatrix}\cosh\xi & \frac{R}{\xi}\sinh\xi\\ \frac{\xi}{R}\sinh\xi & \cosh\xi\end{bmatrix}$$

Abbildung 132: Herleitung einer Matrixdarstellung der linearen Übertragungsfunktion.

$$\begin{bmatrix}\Theta_1\\ \dot{Q}_1\end{bmatrix} = \underline{H}_{\text{Wand}}\cdot\begin{bmatrix}\Theta_2\\ \dot{Q}_2\end{bmatrix} \qquad \text{mit}\quad \underline{H}_{\text{Wand}} = \underline{H}_1\cdot\underline{H}_2\cdot\ldots\cdot\underline{H}_N$$

$$\text{reine Widerstandsschicht}\quad \underline{H}_i = \begin{bmatrix}1 & R_i\\ 0 & 1\end{bmatrix}$$

Abbildung 133: Matrizenmultiplikation im Fall mehrschichtiger Wände

Da die Determinante der Übertragungsmatrix $\underline{H}(s)$ gleich eins ist, lässt sich die Matrixgleichung nach beliebigen Kombinationen der 4 relevanten Größen auflösen (Abb. 134).

$$\begin{bmatrix}\dot{Q}_1\\ \dot{Q}_2\end{bmatrix} = \frac{1}{B}\begin{bmatrix}D & -1\\ 1 & -A\end{bmatrix}\cdot\begin{bmatrix}\Theta_1\\ \Theta_2\end{bmatrix}$$

Abbildung 134: Berechnung der linearen Übertragungsfunktion bei gegebenen Temperaturen im Laplace-Bereich

9.2.2 Die z-Transformation

Gelingt es, das Wandverhalten im z-Bereich durch eine Übertragungsfunktion *G(z)* abzubilden, lässt sich hieraus eine Vorschrift ableiten, wie aus den Systemeingängen und Systemausgängen zu früheren Zeitpunkten die aktuelle Systemantwort zu berechnen ist.

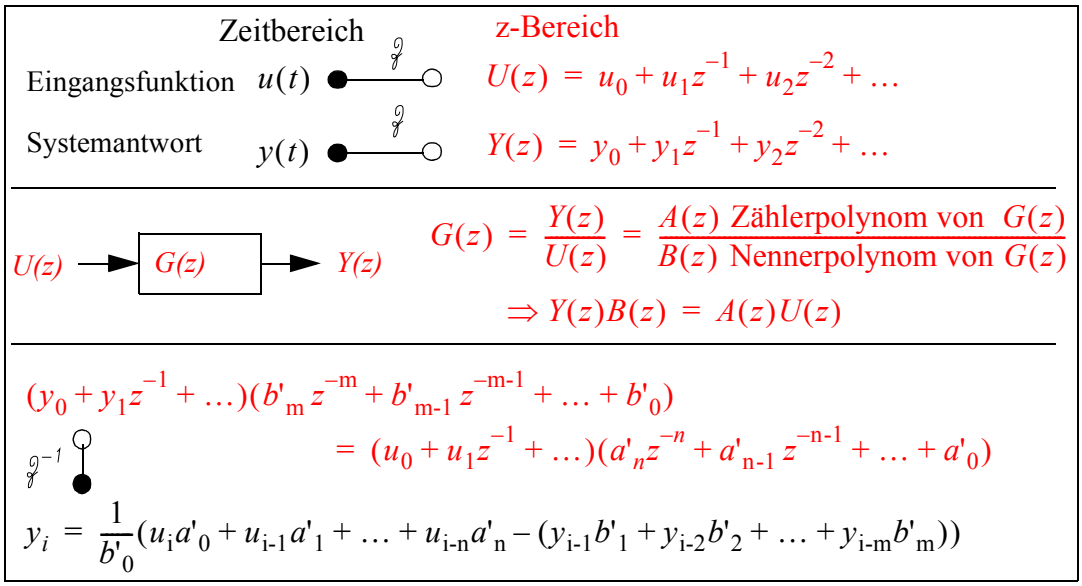


Abbildung 135: Anwendung der z-Transformation. Zur Abkürzung wird die Schreibweise $y_i = y(t=iT)$ genutzt.

So kann in Abb. 135 die Systemantwort y zum Zeitpunkt iT bei bekannten Koeffizienten *a'* und *b'* aus den Systemein- und -ausgängen zu zurückliegenden Zeitpunkten durch eine lineare Berechnungsvorschrift bestimmt werden.

Im folgenden werden zur Veranschaulichung der Methode für den Spezialfall (Abb. 134) die Koeffizienten des Zähler- und Nennerpolynoms (*a'* und *b'*) der Übertragungsfunktion *G(z)* berechnet. Mit der Annahme $\Theta_1 = 0$ reduziert sich das Problem auf die Gleichung $Q_2 = -A/B \cdot \Theta_2$. Beispielsweise führt die Berechnung der Wärmestromdichte auf der Außenwandoberfläche *(Index 2)* bei konstanter Oberflächentemperatur auf der Innenseite *(Index 1)* und einer bekannten, variierenden Oberflächentemperatur auf der Außenseite *(Index 2)* zu dieser Übertragungsfunktion.

Bei der Simulation wird die erregende Funktion als Ergebnis einzelner Simulationsschritte stets in Form einer diskreten Folge vorliegen. Interpoliert man diese beispielsweise in Form einer Treppenfunktion, so ergibt sich als Systemantwort das Produkt $G(s) \cdot 1/s$. Bei einer linearer Interpolation der Folge und damit der Darstellung als Rampenfunktion ergibt sich als Systemantwort das Produkt $G(s) \cdot 1/s^2$. Dies wird im folgenden angenommen.

Nach Durchführung einer Partialbruchzerlegung lassen sich die einzelnen Summanden unter Nutzung der bekannten Korrespondenzen komponentenweise z-transformieren (Abb. 136).

$$Y(s) = G(s)U(s) = \frac{A}{s^2 B} = \frac{c_0}{s^2} + \frac{c_1}{s} + \sum_{n=1}^{\infty} \frac{d_n}{s - \beta_n}$$

$$c_0 = \left.\frac{A}{B}\right|_{s=0} \qquad c_1 = \left.\frac{d}{ds}\frac{A}{B}\right|_{s=0} \qquad d_n = \left.\frac{A}{s^2 \frac{dB}{ds}}\right|_{s=\beta_n}$$

$$Y(z) = \frac{c_0 T}{z(1-z^{-1})^2} + \frac{c_1}{1-z^{-1}} + \sum_{n=1}^{\infty} \frac{d_n}{1 - e^{\beta_n T} z^{-1}}$$

Abbildung 136: Berechnung der z-Transformierten der Systemantwort *Y(z)*. Die Koeffizienten ergeben sich als Ergebnis einer Partialbruchzerlegung. Die Größen β_n sind die Nullstellen von *B*.

Aus der z-Transformierten der Systemantwort $Y(z)$ (Abb. 137) lässt sich durch Division mit der z-Transformierten der erregenden Funktion $U(z)$ (Abb. 137) die z-Transformierte der Übertragungsfunktion ablesen. Ziel ist es, letztere in die Form einer Reihendarstellung zu überführen, deren Koeffizienten vorab berechnet werden können.

$$u(t) = t \quad \overset{\mathcal{L}}{\circ\!\!-\!\!\bullet} \quad U(s) = 1/s^2 \quad \overset{\mathcal{Z}}{\bullet\!\!-\!\!\circ} \quad U(z) = T/z(1-z^{-1})^2$$

Abbildung 137: Berechnung der z-Transformierten der erregenden Funktion

Bringt man die Summanden von *G(z)* auf einen gemeinsamen Nenner, erkennt man, dass dieser die Form des in Abb. 138 gezeigten Produktes hat. Die sich nach Ausmultiplizieren ergebende Summe besitzt schnell kleiner werdende Summanden, so dass es genügt sie nur bis zu einem maximalen *n* auszuwerten und die weiteren Summanden zu vernachlässigen. Das Zählerpolynom berechnet sich analog.

z-Transformierte der Übertragungsfunktion

$$G(z) = \frac{A(z)}{B(z)} = \frac{Y(z)}{U(z)}$$

Berechnung des Nennerpolynoms durch Ausmultiplizieren

$$B(z) = \prod_{n=1}^{\infty} (1 - e^{\beta_n T} z^{-1}) = 1 - \left(\sum_{n=1}^{\infty} e^{\beta_n T} \right) z^{-1} + \ldots$$

Berechnung des Zählerpolynoms

$$A(z) = B(z) \cdot \frac{Y(z)}{U(z)} = \prod_{n=1}^{\infty} (1 - e^{\beta_n T} z^{-1}) \cdot Y(z) \cdot \frac{z(1-z^{-1})^2}{T}$$

Abbildung 138: Die z- Transformierte der Übertragungsfunktion

Nach Kenntnis der Koeffizienten *a'* und *b'* ist es möglich die Systemantwort, also den Wärmestrom auf der Wandoberfläche, zum Zeitpunkt $\mathrm{i}T$ aus den Temperaturen und Wärmeströmen auf der Wandoberfläche zu zurückliegenden Zeitpunkten der Vergangenheit durch eine lineare Berechnungsvorschrift (wie in Abb. 135) gezeigt zu erhalten.

Lebenslauf

Persönliche Daten

Dr. Rolf Mathias Merz

geboren: 24.3.1965 Kaiserslautern
Familienstand: ledig, keine Kinder

Schulausbildung

1971-1973 Grundschule auf dem Betzenberg
1973-1975 Grundschule in der Stiftswaldstraße

1975-1984 Staatl. Gymnasium a. d. Burgstraße
(Abitur 1984)

Hochschulausbildung

1984-1990 Studium der Physik an d. Universität Kaiserslautern
mit der Vertiefungsrichtung Experimentalphysik
(Diplom-Physiker 1990)

1990-1993 Promotionsstudium an d. Universität Kaiserslautern
Promotionsstipendium nach dem Landesgraduiertenförderungsgesetz
(Dr. rer. nat. 1994)

Dissertationsschrift

Merz R.: „Untersuchungen zur Wechselwirkung niederenergetischer Elektronen mit kurzkettigen Kohlenwasserstoffen und deren perfluorierten Derivaten“, Tectum-Verlag, Marburg (1995)

Berufliche Tätigkeit

1993 - 1998 Wissenschaftlicher Mitarbeiter
im Fachbereich Physik der Univ. Kaiserslautern

1998 - 2002 Wissenschaftlicher Mitarbeiter
am Lehrstuhl für Automatisierungstechnik
im Fachbereich Elektro- und Informationstechnik
der Univ. Kaiserslautern

Seit 2002 Wissenschaftlicher Mitarbeiter
im Institut für Oberflächen- und Schichtanalytik (IFOS)
an der Univ. Kaiserslautern