

**Hasso-Plattner-Institut für Softwaresystemtechnik
an der Universität Potsdam**

Konzepte der Beschreibung interaktiver Systeme

Dissertation

zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften
(Dr.-Ing.)
in dem Fachgebiet Software-Engineering

eingereicht an der
Mathematisch-Naturwissenschaftlichen Fakultät
der Universität Potsdam

von
Dipl.-Ing. Andreas Knöpfel

Potsdam, 3. August 2004

Vorwort

An erster Stelle möchte ich mich bei Herrn Prof. Dr.-Ing. Siegfried Wendt bedanken. Sein wissenschaftliches Weltbild, sein analytischer, kritischer Geist und seine menschlichen Überzeugungen haben über viele Jahre hinweg, zunächst im Studium und später auch in vielen persönlichen Gesprächen mein Denken geprägt.

Explizit möchte ich in diesem Zusammenhang ebenfalls Herrn Wolfram Kleis danken, der als wissenschaftlicher Mitarbeiter am Lehrstuhl Wendt in Kaiserslautern während meiner Diplomarbeit entscheidend mein Verständnis für die Beschreibung informativeller Systeme geschärft hat.

Den Wert dieser Sichtweise konnte ich in den folgenden vier Jahren als Mitarbeiter der Firma SAP unmittelbar erleben. Angestoßen durch verschiedene Projekte, begann ich mich während dieser Zeit verstärkt mit der Architektur interaktiver Systeme zu beschäftigen. Im Laufe vieler Diskussionen wuchs mein Interesse an diesem Thema ebenso wie meine Unzufriedenheit über die Darstellung zentraler Konzepte in diesem Bereich.

Diese Unzufriedenheit war der Ausgangspunkt meiner Promotion, als mich mein Wechsel von der SAP an das neu gegründete Hasso-Plattner-Institut Mitte 2000 wieder in die Arbeitsgruppe von Professor Wendt führte. Das menschliche Arbeitsklima sowie das gemeinsame Interesse aller Mitglieder der Arbeitsgruppe, etwas zum Aufbau des Institutes beitragen zu können, hat wesentlich zum Entstehen dieser Arbeit beigetragen.

Für das gegenseitige Vertrauen und die stete Bereitschaft zu offenen und anregenden Diskussionen während dieser Zeit möchte ich mich namentlich weiterhin bei Rémy Apfelbacher, Bernhard Gröne, Frank Keller, Rudolf Kugel, Oliver Schmidt, Peter Tabelling, sowie nicht zuletzt meinen Eltern bedanken.

Den Herren Dr.-Ing. habil. Kenji Hanakata und Prof. Dr.-Ing. habil. Lothar Litz danke ich sehr für die Übernahme des Koreferats.

Andreas Knöpfel

Abstract

Interactive systems are dynamic systems which provide services to one or more users via a user interface. Many of these systems have an information processing core. To effectively use such a system, a user needs to know about the purpose and functional concepts of the system. Only in case of a rather simple functionality, the required knowledge is likely to be obtained by mere exploration of the user interface. For complex systems, a comprehensive description is essential for effective and efficient operation, but also for system development. Especially in the context of this publication, the focus is on the diagrams that are used for communication in the development process of interactive systems.

With regard to its purpose these descriptions vary in form and notation. In any case, it is the underlying terminology which is crucial to the understandability. Established concepts and notations for the description of generic information processing systems promote a strict separation of three categories of system structures: Compositional structures, behavioral structures and value-range structures.

This publication suggests a specialization of this approach for the description of interactive systems. The definition of specific notions and semantic layers is intended to provide a complement to existing description approaches in that field that constitutes a mental framework to enhance the efficiency of communication about interactive systems. In focusing to the description aspect only, design principles, system architectures and development methods are the context but not the subject of this work.

The separation of three semantic layers provides the foundation to distinguish purpose-related, interaction-related and implementation-related models of interactive systems. Each semantic layer is characterized by a specific terminology. Referring to the idea of equivalent models, the compositional system structure varies from layer to layer and provides the framework to ask for the observable behavior and values. The strict assignment of system aspects to system components distinguishes this approach from a simple layering of design decisions in the development of interactive systems as commonly found in the relevant literature. The Fundamental Modeling Concepts (FMC) provide the foundation for the identification and representation of system structures.

A selection of system descriptions taken from the relevant literature provides the starting point to demonstrate the application of the suggested concept. Purpose, content and form of each example are analyzed with regard to the implied system structure. Weaknesses in the original representation are identified. Alternative models complement the examples to illustrate the benefit of the new approach.

Zusammenfassung

Interaktive Systeme sind dynamische Systeme mit einem zumeist informationellen Kern, die über eine Benutzungsschnittstelle von einem oder mehreren Benutzern bedient werden können. Grundlage für die Benutzung interaktiver Systeme ist das Verständnis von Zweck und Funktionsweise. Allein aus Form und Gestalt der Benutzungsschnittstelle ergibt sich ein solches Verständnis nur in einfachen Fällen. Mit steigender Komplexität ist daher eine verständliche Beschreibung solcher Systeme für deren Entwicklung und Benutzung unverzichtbar.

Abhängig von ihrem Zweck variieren die Formen vorgefundener Beschreibungen in der Literatur sehr stark. Ausschlaggebend für die Verständlichkeit einer Beschreibung ist jedoch primär die ihr zugrundeliegende Begriffswelt. Zur Beschreibung allgemeiner komplexer diskreter Systeme — aufbauend auf einer getrennten Betrachtung von Aufbau-, Ablauf- und Wertestrukturen — existiert eine bewährte Begriffswelt. Eine Spezialisierung der Begriffs- und Vorstellungswelt, die den unterschiedlichen Betrachtungsebenen interaktiver Systeme gerecht wird und die als Grundlage beliebiger Beschreibungsansätze interaktiver Systeme dienen kann, gibt es bisher nicht.

Ziel dieser Arbeit ist die Bereitstellung einer solchen Begriffswelt zur effizienten Kommunikation der Strukturen interaktiver Systeme. Dadurch soll die Grundlage für eine sinnvolle Ergänzung bestehender Beschreibungs- und Entwicklungsansätze geschaffen werden. Prinzipien der Gestaltung von Benutzungsschnittstellen, Usability- oder Ergonomiebetrachtungen stehen nicht im Mittelpunkt der Arbeit.

Ausgehend von der informationellen Komponente einer Benutzungsschnittstelle werden drei Modellebenen abgegrenzt, die bei der Betrachtung eines interaktiven Systems zu unterscheiden sind. Jede Modellebene ist durch eine typische Begriffswelt gekennzeichnet, die ihren Ursprung in einer aufbauverwurzelten Vorstellung hat. Der durchgängige Bezug auf eine Systemvorstellung unterscheidet diesen Ansatz von dem bereits bekannten Konzept der Abgrenzung unterschiedlicher Ebenen verschiedenartiger Entwurfsentscheidungen. Die Fundamental Modeling Concepts (FMC) bilden dabei die Grundlage für die Findung und die Darstellung von Systemstrukturen.

Anhand bestehender Systembeschreibungen wird gezeigt, wie die vorgestellte Begriffswelt zur Modellfindung genutzt werden kann. Dazu wird eine repräsentative Auswahl vorgefundener Systembeschreibungen aus der einschlägigen Literatur daraufhin untersucht, in welchem Umfang durch sie die Vorstellungswelt dynamischer Systeme zum Ausdruck kommt. Defizite in der ursprünglichen Darstellung werden identifiziert. Anhand von Alternativmodellen zu den betrachteten Systemen wird der Nutzen der vorgestellten Begriffswelt und Darstellungsweise demonstriert.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Problem und Lösungsansatz	1
1.2	Gliederung der Arbeit	4
2	Grundlegende Begriffe	6
2.1	Begriff des dynamischen Systems	6
2.2	Systemmodelle	7
2.2.1	Modelle als Ergebnis der Wahrnehmung	8
2.2.1.1	Invariantenfindung	8
2.2.1.2	Gegenständliche Objekte	9
2.2.1.3	Prozesse	9
2.2.1.4	Schnittstellen	9
2.2.1.5	Systemmodell und Beobachtungsorte	10
2.2.1.6	Systemkomponenten	10
2.2.2	Modelle als Grundlage der Vorstellung	10
2.2.3	Typ, Klasse und Exemplar	11
2.2.4	Vermittlung von Vorstellungen	12
2.2.5	Klassifikation von Systemmodellen und Beschreibungen	16
2.2.5.1	Analytische versus didaktische Modelle und Beschreibungen	16
2.2.5.2	Zweckbezogenes Modell, Realisierungsmodell und Ersatzmodelle	18
2.3	Informationelle Systeme	21
2.3.1	Charakterisierung informationeller Systeme	21
2.3.2	Realisierung informationeller Systeme	22
2.3.3	Beschreibung informationeller Systeme	23
2.3.4	Fundamental Modelling Concepts	25

3	Interaktive Systeme	28
3.1	Begriffe	28
3.2	Primäre Modellebenen interaktiver Systeme	32
3.3	Zweckbezogene Modelle	39
3.3.1	Zweck und Charakterisierung der Modelle	39
3.3.2	Allgemeine Ansätze zur Modellfindung	40
3.3.3	Abgrenzung des informationellen Systems	44
3.3.4	Konzepte der Modellverfeinerung	45
3.3.5	Abgrenzung von der Ebene interaktionsbezogener Modelle	46
3.4	Interaktionsbezogene Modelle	48
3.4.1	Zweck und Charakterisierung der Modelle	48
3.4.2	Merkmale von Anwendern und Benutzern	50
3.4.3	Interaktionsadapter	52
3.4.3.1	Identifikation informationeller Werte	52
3.4.3.2	Abgrenzung von Interaktionsadaptern	54
3.4.3.3	Formadapter	55
3.4.3.4	Schrittfolgenadapter	57
3.4.3.5	Mehrstufige Wertabbildung	58
3.4.4	Gebräuchliche Gestaltungsmuster	60
3.4.4.1	Abbildung strukturierter Werte	60
3.4.4.2	Abbildung von Werten überschaubaren Repertoires	61
3.4.4.3	Interaktion mit virtuell gegenständlichen Objekten	62
3.4.4.4	Sekundäre Kanäle und Interaktionsassistenten	63
3.4.4.5	Angemessene Schnittstellengestaltung	65
3.4.5	Kanalmultiplex	66
3.5	Realisierungsbezogene Modelle	68
3.5.1	Zweck und Charakterisierung der Modelle	68
3.5.2	Konzepte der Realisierung interaktiver Systeme	70
3.5.2.1	Schnittstellengeräte und informationeller Kern	70
3.5.2.2	Offene Entwurfsentscheidungen	72
3.5.2.3	Trägersystemmodell und Entwicklungsumgebung	74
3.5.3	Einfluss auf die Benutzungsschnittstelle	74

4 Beschreibungsansätze in der Literatur	77
4.1 Beschreibung von Interaktionsprozessen	79
4.1.1 Transitionsdiagramme — Parnas	79
4.1.2 Multiparty-Grammatiken — Shneiderman	81
4.2 Beschreibung und Design interaktiver Systeme	83
4.2.1 SA/SD und HCI-Design — Sutcliffe	83
4.2.2 Task analysis und HCI-Design — Johnson	86
4.2.3 Task analysis, Objekt-Orientierung und HCI-Design — Stry . .	89
4.2.4 Rational Unified Process und HCI-Design — Kruchten	92
4.3 Beschreibung von Architekturmodellen	94
4.3.1 Model-View-Controller (MVC)	94
4.3.2 Presentation-Application-Control (PAC)	100
4.3.3 Seeheim-Modell	103
5 Fazit und Ausblick	107
Literaturverzeichnis	109
Index	115

Kapitel 1

Einleitung

1.1 Problem und Lösungsansatz

Menschen sind ständig mit ihrer Umgebung in Wechselwirkung. Geschieht dies im Rahmen einer bewussten Handlung, so verfolgt der Handelnde eine bestimmte Absicht. Die Wechselwirkung zwischen Handelndem und der Umgebung dient einem bestimmten mehr oder weniger klar definiertem Zweck. Beispiele für solche Wechselwirkungen sind die Interaktion eines Menschen mit anderen Menschen, die Fahrt mit einem Auto oder das Backen eines Brotes. Der Zweck mancher Handlungen ist nur eingebettet in einen größeren Kontext, vor dem Hintergrund eines übergeordneten Zieles, zu verstehen. Vor diesem Hintergrund wird sich das gewählte Verfahren als mehr oder weniger zweckmäßig beurteilen lassen. In manchen Fällen wird sich herausstellen, dass das übergeordnete Ziel klarer oder anders definiert werden muss. In allen diesen Fällen aber ist der Mensch Komponente eines dynamischen Systems, welches aus unterschiedlichen Komponenten besteht, die miteinander in Wechselwirkung stehen.

Für jedes System dieser Art lässt sich angeben, welche Komponenten miteinander in Wechselwirkung stehen und worin die jeweilige Art der Wechselwirkung besteht. Je mehr Wechselwirkungen im Hinblick auf den beabsichtigten Zweck optimiert werden können, umso besser wird das System seinem Zweck gerecht werden können. Im Hinblick auf die zentrale Rolle technischer Systeme im Leben des heutigen Menschen folgen daraus zwei Fragen:

- Wie müssen technische Systeme beschaffen sein, damit Menschen optimal mit ihnen interagieren können bzw. damit sie sich in gegebene menschliche Umgebungen integrieren lassen?
- Wie lassen sich solche Systeme schaffen?

Diese Fragen sind nicht neu. An Relevanz gewinnen sie dadurch, dass in zunehmendem Maße technische Gebilde zu allgegenwärtigen Hilfsmitteln werden, deren effektive und effiziente Benutzung ohne aufwendiges Studium möglich sein soll. Das leichte Erlernen des Umgangs bestimmt heutzutage wesentlich den praktischen und wirtschaftlichen Erfolg dieser Produkte.

Mit der fortschreitenden Verbreitung *programmierter Systeme* in allen Bereichen des menschlichen Lebens werden an die Bedienbarkeit auch solcher Systeme immer höhere Ansprüche gestellt. Ursprünglich dienten solche Systeme rein informationellen Zwecken wie der Erfassung und Auswertung von Daten oder der Lösung mathematischer Probleme. Heute finden sich darunter auch viele Systeme, deren Zweck primär wenig mit Informationsverarbeitung zu tun hat. Darunter befinden sich auch solche, deren Vorläufergenerationen noch ganz ohne Informationstechnologie realisiert wurden. Typische Beispiele hierfür sind Fotoapparate oder Küchengeräte. Vielfach leiden diese Systeme in gleicher Weise unter den Möglichkeiten dieser neuen Technik, wie sie davon profitieren. Der Ergonom Alan Cooper hat es einmal so ausgedrückt: „Egal ob man ein Auto, eine Kamera oder was auch immer mit einem Computer kreuzt, man erhält immer einen Computer“. Es ist Anwendern schwer zu vermitteln, warum durch die Umstellung auf eine neue Technik der Nutzen geschmälert wird.¹

Eine Herausforderung besteht darin, die Kluft zwischen der Welt der Anwender, denen die technischen Gebilde einen Nutzen bringen sollen, und der Welt der Entwickler, deren Verständnis von der Welt der Anwender häufig durch die Fülle technischer Ansätze verstellt ist, zu verkleinern. Die Klasse programmierter Systeme mit Schnittstellen zur Interaktion mit dem Menschen — vielfach als *interaktive Systeme* bezeichnet — bildet daher ein eigenständiges Forschungsgebiet. Je nach Schwerpunkt wird das Themengebiet auch mit dem Begriff *Mensch-Maschine-Schnittstellen* (engl. *Human-Computer Interfaces*) bezeichnet².

Seit mehr als zwei Jahrzehnten beschäftigen sich Fachleute unterschiedlicher Disziplinen mit diesem Themengebiet unter dem Blickwinkel, der die jeweilige Disziplin kennzeichnet. Dazu gehören unter anderen Ergonomen, Software-Designer, Psychologen und Linguisten. Einhergehend mit der schnellen technologischen Entwicklung immer neuer interaktiver Systeme — angefangen von frühen rein Text-basierten Systemen bis hin zu heutigen Multimedia-Systemen — ist eine Fülle von Konzepten zur Beschreibung, Modellierung und Entwicklung interaktiver Systeme entstanden. Trotz der enormen Fortschritte auf diesem Gebiet fallen beim Studium der einschlägigen Literatur zu diesem Themengebiet [Sta96, Shn98, Har01] einige ungelöste Problempunkte auf:

- Mit vielen Konzepten zur Schaffung interaktiver Systeme ist weniger der Aufbau einer Systemvorstellung verknüpft, als vielmehr ein Vorgehensmodell zur Ableitung formaler Beschreibungen. Das dafür erforderliche Anwendungswissen wird als gegeben angenommen. Eine Integration zukünftiger Anwender in den Entwicklungsprozess ist häufig nur auf der Basis von Prototypen möglich. Defizite in der Systemdarstellung führen dazu, dass der Bezug zwischen unterschiedlichen Konzepten nur schwer erkennbar ist.
- Trotz der gängigen Verwendung zentraler Begriffe wie interaktives System, Schnittstelle und Modell, sind die Begriffe vielfach unklar bzw. uneinheitlich de-

¹Einige Beispiele für solch fragwürdige Ergebnisse sind in [GG96] und [Nor01] aufgeführt.

²Die folgenden Kapitel werden zeigen, dass die klare Definition beider Begriffe recht problematisch ist.

finiert. Die Abgrenzung von Modellen, interaktivem System und Komponenten der Umgebung bleibt häufig unklar.

- Viele der Entwurfsansätze implizieren konkrete Implementierungsplattformen, die naturgemäß einem fortlaufendem Wandel unterworfen sind. Hierzu gehört auch die Bindung an bestimmte Modellierungswerkzeuge und Implementierungssprachen.

Daran zeigt sich, dass es auch innerhalb des Kreises der Experten zum Thema interaktiver Systeme kein allgemein akzeptiertes Grundverständnis gibt. In den seit Jahren stattfindenden Integrationsprojekten und Konferenzen [Bar93, RNT95, ICS03] wird dies immer wieder geäußert. So heißt es in [SM91] „Dieser Mangel an Fortschritt mag daraus resultieren, dass die Designer von Benutzungsschnittstellen ihre Aktivitäten nicht mit dem Software Engineering verbinden konnten“ (Übersetzung des Autors).

Als Ursachen werden in [Sta96] „das Fehlen von Konzepten zur Integration von Aufgaben-, Datenmodellen und Interaktionsmodellen“, die mangelnde Berücksichtigung der Umgebung interaktiver Systeme, der „Wissens-, Transparenz- und Konsistenzverlust“ durch „Lücken im Designprozess“, sowie ferner ein „Ungleichgewicht zwischen statischer und dynamischer Spezifikation“ identifiziert.

Aufbauend auf einer Begriffs- und Vorstellungswelt, die den Gedanken eines dynamischen informationellen Systems konsequent in den Mittelpunkt der Vorstellung stellt, kann nach Ansicht des Autors ein praxisrelevanter Beitrag zum Schließen der Kluft zwischen den unterschiedlichen Interessenvertretern im Umgang mit interaktiven Systemen geschaffen werden. Er findet seinen Ausdruck in einer Hierarchie wohl definierter Betrachtungsebenen, die gleichermaßen für alle interaktiven Systeme gültig ist.

Es ist sehr wahrscheinlich, dass sich Verständnisschwierigkeiten zwischen den unterschiedlichen Interessenvertretern im Umgang mit interaktiven Systemen verringern lassen, wenn jede vorkommende Betrachtungsebene durch einer Menge abgeschlossener Systemmodelle definiert ist, die in der Aufbauvorstellung verwurzelt und explizit sichtbar gemacht sind.

Als besonders geeigneter Ausgangspunkt zur Untermauerung dieser These erscheinen dem Autor die Konzepte zur Modellierung allgemeiner informationeller Systeme nach Wendt [Wen01]. Deren primäres Ziel stellt die Verbesserung des Wirkungsgrades der Kommunikation über informationelle Systeme zwischen Menschen dar. Der seit vielen Jahren praktisch erprobte Ansatz integriert ein solides, in der Systemtheorie verwurzeltes Fundament und didaktisch bewährte Darstellungsmittel in Form leicht verständlicher, übersichtlicher, an gängige Normen angelehnte Plantypen.

Das Ziel dieser Arbeit besteht darin, eine Basis für das Verständnis der an Schaffung und Nutzung beteiligten Personen von interaktiven Systemen bereitzustellen, die unmissverständlich in der Vorstellungswelt dynamischer Systeme verhaftet ist. Der allgemein akzeptierte Ansatz der Abgrenzung unterschiedlicher Ebenen von Entwurfsentscheidungen äußert sich in der Abgrenzung von genau drei unterschiedlichen primären Modellebenen. Modelle unterschiedlicher Ebenen entstehen nicht infolge von hierarchischer Vergrößerung oder Verfeinerung, sondern infolge von Betrachtungsebenenwechseln.

Dabei geht es nicht darum, geeignete Strukturen für zukünftige interaktiver Systeme zu identifizieren, sondern darum, Strukturen beliebiger interaktiver Systeme auf einer gemeinsamen begrifflichen Grundlage im Hinblick auf ein optimales Systemverständnis zu beschreiben. Vor dem Hintergrund der Schaffung interaktiver Systeme geht es um die Frage nach Inhalt und Form von "Ingenieursplänen", die auf dem Weg zur Realisierung des Systems erstellt werden sollten.

Ausgehend von diesen Betrachtungen werden am Beispiel einschlägiger bekannter Ansätze der Beschreibung interaktiver Systeme Defizite des Status Quo identifiziert und Verbesserungsmöglichkeiten aufgezeigt.

1.2 Gliederung der Arbeit

Der Ansatz einer in der Aufbauvorstellung verwurzelten Beschreibung interaktiver Systeme kann nicht vorgestellt werden, ohne dass Leser und Autor eine übereinstimmende Vorstellung von fundamentalen Begriffen wie Wahrnehmung, System, Modellfindung und Beschreibung besitzen. Der Aufbau einer solchen begrifflichen Basis ist der Zweck von Kapitel 2. Viele der darin enthaltenen Betrachtungen entstanden gleichermaßen als Ergebnis des „Nach-Denkens“ von Ausführungen in [Wen91]³.

Ausgehend vom Begriff des dynamischen Systems wird die Findung von Modellen als Ergebnis von Wahrnehmung und Invariantenfindung beschrieben. Die Unterscheidung von Aufbau-, Ablauf- und Wertebereichsstrukturen führt zum Begriff des dynamischen Systems. Die Fähigkeit, mit Stellvertreterformen vereinbarte Vorstellungen zu assoziieren, erlaubt sowohl den Aufbau von Sprache und die Erstellung von Modellbeschreibungen, als auch den Umgang mit Modellsystemen. Beides ist Voraussetzung für die Schaffung und Benutzung interaktiver Systeme.

Die speziellen Merkmale informationeller Systeme bilden einen weiteren Schwerpunkt dieses Kapitels. Einerseits ist die Benutzungsschnittstelle interaktiver Systeme im wesentlichen informationeller Natur, andererseits erfolgt deren Realisierung zu einem großen Teil unter Verwendung informationsverarbeitender Systeme.

Kapitel 3 bildet den Kern der Arbeit. Dort werden die Begriffe definiert, auf denen die in dieser Arbeit vorgeschlagene in der Aufbauvorstellung verwurzelte Betrachtung interaktiver Systeme beruht. Ausgangspunkt ist eine Abgrenzung des Begriffs interaktiver Systeme und damit der Klasse der in dieser Arbeit betrachteten Systeme. Die Identifikation von drei Modellebenen und die Festlegung von Abgrenzungskriterien erlaubt die Beschreibung interaktiver Systeme auf unterschiedlichen Betrachtungsebenen. Ausgehend von den Personen, die an der Modellfindung beteiligt sind, lassen sich die zweckbezogene, die interaktionsbezogene und die realisierungsbezogene Modellebene voneinander unterscheiden. Es wird gezeigt, dass Modelle auf unterschiedlichen

³siehe [Wen91], S. 5: „Daran erkennt man, dass es sich um philosophische Erkenntnisse handelt, denn deren Kennzeichen ist es, dass man sie nicht durch Nachschauen, Nachmessen oder Nachrechnen, sondern nur durch *Nach-Denken* überprüfen kann.“

Betrachtungsebenen nicht als das Ergebnis einer einfachen hierarchischen Verfeinerung oder Vergrößerung angesehen werden dürfen.

Das mit der Aufbauvorstellung verbundene Bedürfnis der Findung zuständiger Systemkomponenten führt auf der Ebene interaktionsbezogener Modelle zu Konzepten wie:

- Formadaptern,
- Schrittfolgenadaptern und
- Interaktionsassistenten

Diese sind oberhalb der Ebene elementarer Interaktionselemente angesiedelt, welche Knöpfe oder Menüs beinhaltet.

Während die letztgenannten Begriffe das Denken bei der Gestaltung von Benutzungsschnittstellen im Hinblick auf die Fähigkeiten zukünftiger Benutzer leiten sollen, ist die Ebene realisierungsbezogener Modelle durch Begriffe wie

- Interaktionswerk,
- Schnittstellengerät und
- Trägersystemmodell

geprägt. Die Definition dieser Begriffe bildet die Grundlage für die Beschreibung realisierungsbezogener Modelle.

In Kapitel 4 wird eine repräsentative Auswahl von fertigen Beschreibungen und Konzepten zur Beschreibung interaktiver Systeme aus der einschlägigen Literatur daraufhin untersucht, ob und in welchem Umfang durch sie die in Kapitel 3 beschriebene Vorstellungswelt zum Ausdruck kommt. Dabei werden zwei Ziele verfolgt:

- Defizite in den ausgewählten Beschreibungsansätzen im Hinblick auf die Vermittlung einer aufbaubasierten Systemvorstellung werden identifiziert.
- Der Nutzen der eingeführten Begriffswelt an Hand von Alternativdarstellungen demonstriert.

Kapitel 2

Grundlegende Begriffe

In diesem Kapitel wird das begriffliche Fundament vorgestellt, auf dem die nachfolgenden Betrachtungen interaktiver Systeme aufbauen. Zunächst wird der Zusammenhang zwischen menschlicher Wahrnehmung, Modellbildung und Systemfindung erklärt. Die Ordnung der folgenden Abschnitte ergibt sich aus folgendem Zusammenhang: Die strukturierte menschliche Vorstellung ist die Grundlage der Ausbildung von Sprachen, die sich zunächst aus dem unmittelbaren Bedürfnis nach Kommunikation zwischen Menschen ergibt. Die Verwendung von Sprache erlaubt die Weitergabe von Modellen in Form von Beschreibungen. Diese Beschreibungen lassen sich abhängig von Zweck und Form, sowie der zugrunde liegenden Systemklasse charakterisieren. Den Schwerpunkt bildet hierbei die Beschreibung dynamischer, insbesondere informationeller Systeme, welche die Grundlage interaktiver System darstellen.

2.1 Begriff des dynamischen Systems

Nach dem Herkunftswörterbuch des Dudens [Dro89] geht der Begriff *System* auf das griechisch(-lateinische) *sýstēma*, "das aus mehreren Teilen zusammengesetzte und gegliederte Ganze" zurück. Ein *dynamisches System* ist demnach ein konkretes oder zumindest ein konkret vorstellbares Gebilde, welches ein beobachtbares Verhalten zeigt, wobei dieses Verhalten als Ergebnis des Zusammenwirkens der Systemteile angesehen werden kann [Wen91]. Beispiele für dynamische Systeme sind das Textverarbeitungssystem, das zur Erstellung dieses Dokumentes genutzt wurde, das Bewässerungssystem eines Landwirtschaftsbetriebes oder ein Industrieunternehmen.

Die obigen Beispiele vermitteln aufgrund ihrer Anschaulichkeit möglicherweise den Eindruck, dass sich Systeme im allgemeinen leicht und intuitiv finden lassen. Auch die Definition des Systembegriffes lebt von dieser Vorstellung. Trotzdem stellt sich in Diskussionen über Systeme häufig heraus, dass die Kommunikationspartner keine gemeinsame Vorstellung von dem betrachteten System besitzen. Sogar die unmittelbare Anschauung führt häufig nicht zu derselben intuitiven¹ Vorstellung von einem System. Dies liegt nur teilweise daran, dass es Systeme gibt, deren Struktur sich

¹Der Begriff der Intuition wird im Zusammenhang mit dem Erwerb um Systemwissen von einigen Experten sogar grundsätzlich in Frage gestellt. So ist für den Vater des Macintosh Computer Projektes

aufgrund bestimmter physikalischer Eigenschaften der unmittelbaren Anschaulichkeit entzieht. Unterschiedliche Erfahrungen und Kenntnisse der Kommunikationspartner sind ebenfalls prägend für eine Systemvorstellung. Alternative Sichten auf die Struktur eines Gebildes sind deshalb in vielen Fällen durchaus zweckmäßig. Die zentralen Fragen lauten somit:

- Wie wird ein beobachtbarer Ausschnitt der Wirklichkeit zu einem System, einem strukturierten Gebilde wechselwirkender Teile?
- Wie erkennt man die Grenze eines Systems, wie die seiner Teile?
- Was macht das Verhalten eines Systems aus?

Im folgenden Abschnitt sind aus diesem Grunde zwei Punkte besonders herausgearbeitet:

- bei allen hier betrachteten Systemen handelt es sich — unabhängig von der abgeleiteten Vorstellung — um konkrete materiell-energetische Gebilde in Raum und Zeit und
- die Struktur eines Systems entsteht letztlich erst in der Vorstellung des Menschen.

Da eine einheitliche Vorstellung von einem System aber die Voraussetzung für die gemeinschaftliche Erschaffung oder Nutzung eines Systems durch die Beteiligten darstellt, ist ein Modell davon, wie solche Vorstellungen entstehen, für die Schaffung und Nutzung dynamischer Systeme von großer Bedeutung.

2.2 Systemmodelle

Der Mensch nimmt seine Umgebung aufgrund ständiger Sinnesreize wahr. Durch Erfahrung lernt ein Mensch von frühester Kindheit an innerhalb des Kontinuums seiner materiell-energetischen Umwelt andere Lebewesen, Gegenstände, auffällige Formen in der Umwelt als eigenständige aus der Umgebung heraus gelöst betrachtbare *Individuen* bzw. *Objekte*² zu erkennen und mit einander in Beziehung zu setzen. Diese Wahrnehmungen beeinflussen daher zwangsläufig sein Verhalten im Umgang mit seiner Umgebung.

der Firma Apple Computer, Jef Raskin, jegliches intuitive Verständnis von einem Sachverhalt auf die vorherige Aneignung von Erfahrungen und Vorwissen zurückführbar[Ras00].

²Individuum ist von dem lateinischen Wort *individuus*, unteilbar, abgeleitet und ist im obigen Zusammenhang eigentlich der passende Begriff. So bezeichnet im Kontext der Biologie der Begriff Individuum eine Pflanze oder ein Tier als konkretes Exemplar einer bestimmten Gattung, was sich mit unserer Vorstellung von Lebewesen als eigenständigen, abgeschlossenen Einheiten deckt. Objekt geht auf substantivierte Partizip Perfekt des lateinischen Wortes *obicere*, entgegenstellen, zurück. Erst im Laufe der Zeit erhielt das Wort seine Bedeutung als Gegenstand oder Inhalt der Vorstellung, auf das sich ein Ziel oder Handeln erstreckt. Heute werden beide Begriffe vielfach synonym verwendet, wobei im technischen Umfeld Objekt der geläufigere Begriff ist.

2.2.1 Modelle als Ergebnis der Wahrnehmung

2.2.1.1 Invariantenfindung

Die Fähigkeit zur Abgrenzung von Objekten ist mithilfe des Prinzips der sogenannten Invariantenfindung erklärbar. Hierbei führen übergangslos aufeinander folgende Wahrnehmungen, die in ihrer Charakteristik ähnlich sind, zum Eindruck der Identität des Wahrgenommenen. Umgekehrt sind in ihrer Charakteristik wechselnde Wahrnehmungen, das heißt auffällige Übergänge, zur Wahrnehmung von Grenzen geeignet.

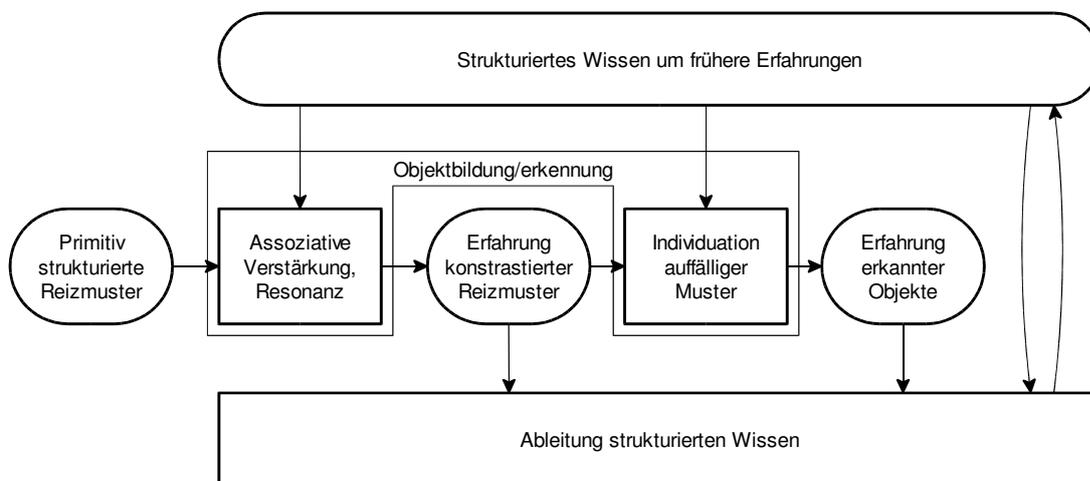


Abbildung 2.1: Invariantenfindung als Grundlage von Erfahrungen

Abbildung 2.1 zeigt ein Modell, anhand dessen sich das Prinzip erklären lässt. Mithilfe seiner Sinnesorgane nimmt ein Mensch in seiner Umgebung materiell-energetische Werte in Form primitiv strukturierte Reize (links im Bild) unterschiedlicher Intensität wahr. Hierbei nimmt ein Mensch bevorzugt solche Reize wahr, die er schon zuvor erfahren hat und an die er sich infolge dessen erinnern kann. Erklären kann man dies durch Assoziation und Resonanz der Erinnerung an frühere Erfahrungen mit den aktuellen Reizen. Ähnliche Muster werden durch die Resonanz verstärkt wahrgenommen. Innerhalb der sich daraus ergebenden kontrastierten Reizmuster lassen sich so Bereiche abgrenzen, die sich geschlossen von ihrer Umgebung abheben. Diese isolierbaren Bereiche auffälliger Reizmuster bewirken die Erfahrung aus der Umgebung abgrenzbarer Objekte. Der Mensch prägt sich solche Objekterfahrungen eng verknüpft mit der damit verbundenen Erfahrung der jeweiligen kontrastierten Reizmuster ein. Über die Zeit sammelt sich strukturiertes Wissen um Objekte mit ihren *Eigenschaften* an. Eigenschaften eines Objektes sind hierbei die beobachtbaren Werte bzw. Werteverläufe, die in ihrer Wahrnehmung an ein Objekt geknüpft sind. Die für die Erkennung eines Objektes ausschlaggebenden Eigenschaften sollen als *charakteristische* Eigenschaften bezeichnet werden.

2.2.1.2 Gegenständliche Objekte

Auf unterster Ebene der Sinneswahrnehmungen treten so aus dem Kontinuum von Farben, Tastempfindungen, Klängen und anderen Wahrnehmungen, die in Raum und Zeit wahrnehmbar sind, gegenständliche Objekte heraus. Diese Objekte sind einerseits durch ihre *räumliche* Grenze zur Umgebung definiert sind und andererseits durch die an dieser Grenze wahrnehmbaren charakteristischen Merkmale. So ist ein Baum inmitten der Natur deutlich als eigenständiges Objekt erkennbar, das sich durch seinen Umriss abgrenzt und an seiner Oberfläche eine charakteristische Färbung aufweist³.

2.2.1.3 Prozesse

In gleicher Weise lässt sich die Abgrenzung charakteristischer Muster über die *Zeit* erklären. So ist ein gesprochenes Wort akustisch durch Anfang und Ende abgrenzbar und besitzt innerhalb dieser Grenzen eine charakteristische Klangform. Aus der unstrukturierten Wahrnehmung kontinuierlicher Größen entsteht so neben der räumlichen Ordnung gegenständlicher Objekte eine zeitliche Ordnung abgrenzbarer Vorgänge, sogenannter *Prozesse*. Jeder Prozess ist dabei durch seinen Anfangs- und seinen Endzeitpunkt begrenzt, die durch das Eintreten eines charakteristischen Sachverhaltes, eines *Ereignisses*, definiert sind.

2.2.1.4 Schnittstellen

In der Regel lässt sich das während eines Prozesses wahrnehmbare Verhalten einem oder mehreren gegenständlichen Objekten zuordnen, die miteinander in *Wechselwirkung* stehen. Mit dem Konzept der Wechselwirkung ist die Vorstellung verbunden, dass bestimmte Objekte auf die Eigenschaften anderer Objekte Einfluss nehmen. Hierbei stellt sich infolge einer räumlichen Kopplung der wechselwirkenden Objekte die Identität bestimmter Eigenschaften der gekoppelten Objekte in den Berührungsorten ein. Die betroffenen Eigenschaften verändern sich infolge der Kopplung in charakteristischer Art und Weise. Die Berührungsorte gegenständlicher Objekte sollen im folgenden *Schnittstellen* genannt werden. Die an einem solchen Ort beobachtbaren und die Schnittstelle charakterisierenden veränderlichen Größen werden *Schnittstellenvariablen* genannt. Ihnen ist zu jedem Zeitpunkt ein konkreter *Wert* zugeordnet ist. Als Beispiele für Schnittstellenvariable können die elektrische Spannung an den Anschlüssen zur Energieversorgung des erwähnten Textverarbeitungssystems oder die Durchflussmenge am Zufluss eines Bewässerungssystems angesehen werden. Beispiele für konkrete Werte wären in diesem Zusammenhang eine Spannung von 230 Volt und eine Strömung von 10 Litern Wasser pro Sekunde. In vielen Fällen sind auch strukturierte Gebilde als Werte anzusehen, wie etwa die Seite, die auf dem Bildschirm des Textverarbeitungssystems aktuell dargestellt wird.

³Im allgemeinen Sprachgebrauch werden Gegenstände und Lebewesen meist insofern unterschieden, als Gegenstände als unbelebte Objekte angesehen werden. Die Bezeichnung "gegenständliches Objekt" wurde daher bewusst als Oberbegriff gewählt.

2.2.1.5 Systemmodell und Beobachtungsorte

Was als Schnittstelle eines Objektes wahrgenommen wird, ist genauso von der subjektiven Wahrnehmung eines Beobachters abhängig, wie die Frage, was als Objekt wahrgenommen wird. Dies ergibt sich aus der natürlichen Beschränkung des Menschen, der nur einen begrenzten *Ausschnitt* seiner Umgebung in begrenzter *Auflösung* wahrnehmen kann. Dabei sind Grad der Auflösung und Umfang des Ausschnitts in der Regel von Mensch zu Mensch unterschiedlich. Obwohl sich das unmittelbare Erleben gegenständlicher Objekte, das ein Mensch aufgrund konkreter Sinnesreize hat, nicht bewusst beeinflussen lässt, ist er dennoch in der Lage, Ausschnitt und Auflösung in gewissen Grenzen frei zu wählen. Der Ausschnitt wird definiert durch die Auswahl einer Menge von *Beobachtungsorten*. Die Auflösung bestimmt den Detailgrad, in dem Objekte und Prozesse erkannt, das heißt voneinander unterschieden werden. Sie wird wesentlich von der Aufmerksamkeit des Beobachters bestimmt und besitzt eine individuell unterschiedliche, natürliche Grenze. Ein Wechsel der Auflösung führt zur Erfahrung von *Bestandteilsbeziehungen* zwischen gegenständlichen Objekten bzw. Prozessen unterschiedlicher Auflösung.

Wichtige Kandidaten für Beobachtungsorte sind die Schnittstellen zwischen erkannten Objekten, wobei beliebige Orte für die Beobachtung wahrnehmbarer Werte ausgewählt werden können. Erlebt ein Mensch dabei sich wiederholende Muster zwischen den beobachteten Werten, führt das zur Erfahrung von Beziehungen zwischen diesen Werten und zum Erkennen von Kausalstrukturen. Wendt definiert hiervon ausgehend den Begriff des *Systemmodells* als „das Netz der angenommenen Kausalbeziehungen, woraus man seine Vorhersagen“ ableitet [Wen91].

2.2.1.6 Systemkomponenten

In der Regel wird bei dynamischen Systemen durch eine Kausalbeziehung eine Menge von Beobachtungsorten identifiziert, die als Menge oder Teilmenge der Schnittstellen eines gegenständlichen Objektes angesehen werden kann. Daraus ergibt sich die Vorstellung der *Zuständigkeit* eines Objektes für die Herstellung einer bestimmten Kausalbeziehung innerhalb der beobachteten Umgebung. Diese zuständigen Objekte bilden gemäß der obigen Definition des dynamischen Systems die besagten Systemteile, die als *Systemkomponenten* bezeichnet werden. Aufgrund seiner klar abgrenzbaren Menge an Beobachtungsorten und Systemkomponenten kann jedes dynamische System ebenfalls als gegenständliches Objekt angesehen werden. Damit ergibt sich die Vorstellung einer Hierarchie aus gegenständlichen Objekten, die ihrerseits aus immer kleineren Objekten zusammengesetzt sein können.

2.2.2 Modelle als Grundlage der Vorstellung

Die Bildung von Systemmodellen ermöglicht es dem Menschen einerseits die Vorgänge in seiner Umgebung vorherzusagen und so sein Leben innerhalb bestehender Systeme zu planen. Andererseits kann er bewusst durch die Schaffung neuer Systeme auf seine Umgebung gestalterisch Einfluss zu nehmen. Aufgrund der Komple-

xität der Summe alles Wissbaren über ein konkretes System, ist der Mensch gezwungen, sich in seiner Vorstellung nacheinander auf unterschiedliche Systemaspekte konzentrieren. Als Aspekt soll hierbei eine Teilmenge aller Sachverhalte verstanden werden, die über ein Objekt bekannt sind und deren gemeinsame Betrachtung einem abgrenzbaren Zweck dient. Auf diesem Wege ergeben sich unterschiedliche Modelle zu ein und demselben System, die sich in Auflösung, Ausschnitt und Betrachtungsebene unterscheiden können [Bun98]. Allgemein kann man in Anlehnung an [Whi87] ein Modell als eine Charakterisierung bezeichnen, die ein Modellbesitzer X von einem bestimmten Objekt Y besitzt.

Ist das betrachtete Gebilde beispielsweise ein Verstärker für elektrische Signale, wird man normalerweise ein Modell bezüglich seines elektrischen Verhaltens und ein anderes Modell der konkreten räumlichen Anordnung seiner Bauteile voneinander unterscheiden. So ist ausgehend vom ersten Modell die Frage beantwortbar, ob der Verstärker klanglich als Komponente einer HiFi-Anlage geeignet ist, während sich ausgehend vom zweiten Modell der erforderliche Platz des Verstärkers ableiten lässt. Trotz ihrer großen Unterschiede implizieren beide Modelle in diesem Beispiel den konkreten Verstärker mit allen seinen Eigenschaften. Lediglich der Zweck der Modelle ist völlig unterschiedlich.

2.2.3 Typ, Klasse und Exemplar

Die Auswahl charakteristischer Sachverhalte, die zu einem Modell führt, bezeichnet man als *Abstraktion*. Das so entstandene Modell ist ein abstraktes Objekt, das keine weiteren Eigenschaften besitzt als die, die durch die ausgewählten Sachverhalte gekennzeichnet werden. Je weniger Sachverhalte in einem Modell berücksichtigt werden, umso wahrscheinlicher ist es, dass dieses Modell auch auf andere Objekte zutrifft.

Ein Modell definiert einen *Typ*, wenn man die Vorstellung hat, dass das Modell nicht die Abstraktion eines bestimmten konkreten Objektes darstellt, sondern dass es ein definierendes Abstraktum einer Menge typgleicher Objekte ist. Eine Menge typgleicher Objekte wird als *Klasse* bezeichnet. Dementsprechend bezeichnet man den Vorgang der Klassenbildung als *Klassifikation*. Die Elemente einer Klasse werden als *Exemplare* des Typs der Klasse bezeichnet. In diesem Zusammenhang muss darauf hingewiesen werden, dass es reine Konstrukte der Vorstellung gibt, mit denen man trotzdem die Vorstellung konkreter Objekte verbindet. So verbindet man mit der Gestalt Frodos aus der Erzählung „Der Herr der Ringe“ von J.R.R. Tolkien durchaus die Vorstellung von einem konkreten Individuum und nicht von einem abstrakten Objekt.

In gleicher Weise, wie man konkrete Objekte klassifizieren kann, lassen sich auch Eigenschaften, Beziehungen und Strukturen klassifizieren. Ebenso lassen sich durch erneute bzw. wiederholte Abstraktion über bereits gefundene Typen weitere abstraktere Typen finden, die auf der Schnittmenge der charakterisierenden Sachverhalte der Ausgangstypen beruhen. Im Ausnahmefall ergibt sich eine Hierarchie von Typen, deren Wurzeltyp im Extremfall durch den einen Sachverhalt des Objektseins gekennzeichnet ist.

Im allgemeinen ist es nicht zweckmäßig, nach einer einzigen Klassenhierarchie zu suchen, auch wenn dies in Ausnahmefällen notwendig sein mag. So stehen die Klasse

der roten Gegenstände und die Klasse der Autos weder in einer hierarchischen Beziehung, noch sind die Klassen disjunkt zueinander. Vielmehr gibt es eine Schnittmenge von Objekten, auf die der Sachverhalt des Rot-Seins und der des Auto-Seins gleichermaßen zutrifft. Sie sind somit Exemplare beider Klassen.

Für den Menschen sind Typen von Bedeutung, da sie ihm als Muster zur Erkennung existenter Objekte und zur Ableitung neuer Objekte dienen. Durch wiederholte Klassifikation lassen sich einzelne Objekte oder eine Menge von Objekten aus einem Repertoire auswählen, das durch reine Aufzählung unüberschaubar wäre. Ebenso lässt sich durch Zuordnung zu unterschiedlichen Klassen auch von komplexen Objekten effizient eine konkrete Vorstellung gewinnen. Typen sind somit ein wichtiges Mittel zur Beherrschung von Komplexität für den Menschen. Basierend auf dem Wissen um konkrete Objekte und Typen lässt sich so das Wissen um die Struktur und das Verhalten dynamischer Systeme effizient strukturieren.

Auch das Finden abstrakter Objekte lässt sich mit dem Modell der Invariantenfindung erklären. Um Missverständnisse zu vermeiden, ist eine Unterscheidung im Nachdenken über konkrete Objekte und dem Nachdenken über Typen zu empfehlen. Ob eine Menge von Sachverhalten explizit auf ein bestimmtes Objekt in Raum und Zeit Bezug nimmt oder ob sie definitorisch eine Klasse von Objekten umschreibt, ist eine bewusste Entscheidung. Deshalb ist zwischen Systemmodellen und Systemmodelltypen zu unterscheiden. Im Zusammenhang mit der Kommunikation über Systeme ist dies insofern problematisch, als aus vielen Beschreibungen nicht hervorgeht, ob sie ein konkretes System oder eine ganze Klasse von Systemen identifizieren.

2.2.4 Vermittlung von Vorstellungen

Zum Zwecke der Arbeitsteilung, der Konfliktlösung oder auch nur zur Unterhaltung tauschen Menschen ihre Vorstellungen, insbesondere Aufforderungen und Antworten, aus. Dieser Prozess wird als *Kommunikation* bezeichnet. Im einfachsten Falle wirken an einem solchen Prozess nur zwei Personen mit. Dabei lenkt einer der Mitwirkenden, der als *Sender* bezeichnet werden soll, die Aufmerksamkeit des zweiten Mitwirkenden, der als *Empfänger* bezeichnet werden soll, auf ein bestimmtes Objekt und vermittelt diesem so die gewünschte Vorstellung. Dieser Prozess wird als *Identifikation* bezeichnet, und kann in dreierlei Form geschehen:

- Zeigen,
- Benennen und
- Umschreiben.

Das *Zeigen* setzt eine gleichermaßen von Sender und Empfänger beobachtbare Umgebung voraus. Abbildung 2.2 zeigt, dass der Sender, also der Zeigende, dabei ein Zeigemittel benutzt, um die Aufmerksamkeit des Empfängers auf den Ort in der Umgebung zu lenken, an dem das identifizierte Objekte beobachtbar ist. Die vom Sender

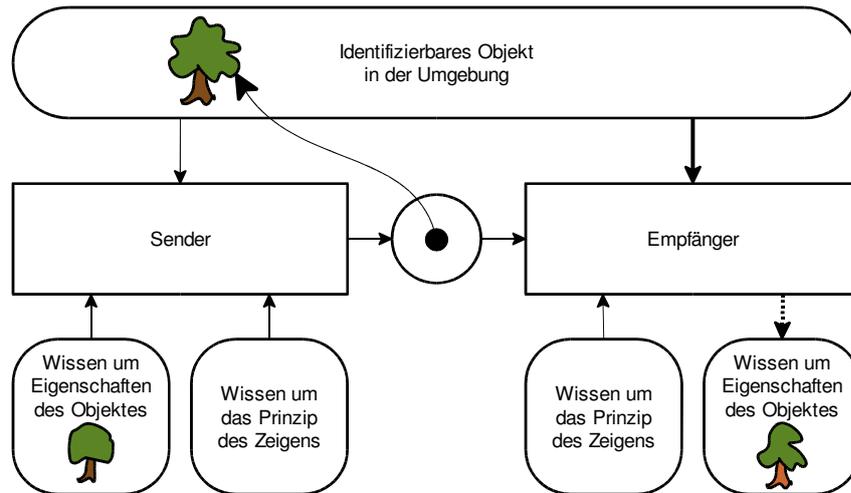


Abbildung 2.2: Zeigen konkreter Objekte

zu vermittelnde Vorstellung ergibt sich beim Empfänger durch unmittelbare Erfahrung des identifizierten Objektes. In der Regel wird die resultierende Vorstellung von dem Objekt zwischen Sender und Empfänger aufgrund ähnlicher Wahrnehmungen recht ähnlich sein. Um ein Objekt durch Zeigen zu identifizieren, benötigen Sender und Empfänger ein bestimmtes Vorwissen. Der Sender muss das Wissen besitzen, das zu zeigende Objekt in der Umgebung zu lokalisieren. Außerdem muss er sich der Aufmerksamkeit des Empfängers sicher sein können. Hinzu kommt das Wissen um unterschiedliche Zeigemittel. So kann man das zu identifizierende Objekt jemandem in die Hand geben, mit dem Finger in dessen Richtung zeigen oder das Objekt mit einem Pfeil markieren. Darüber hinaus ist die Vorgabe des Beobachtungsortes aufgrund von Enthaltenseinsbeziehungen zwischen Objekten häufig mehrdeutig. Die Klärung solcher Mehrdeutigkeiten erfordert zusätzliche Kommunikationsschritte zwischen den Mitwirkenden, die hier nicht gezeigt sind. Wesentlich ist, dass der Empfänger kein Vorwissen über das zu identifizierende Objekt benötigt. Der Zweck des Zeigens besteht in der Vermittlung von Wissen über das identifizierte Objekt. Das Zeigen ermöglicht den Beteiligten den Aufbau einer gemeinsamen Vorstellung von Objekten und den daran beobachtbaren Sachverhalten.

Ein Sonderfall des Zeigens, das *Präsentieren*, liegt vor, wenn ein Sender zum Vermitteln einer Vorstellung ein konkretes künstlich erschaffenes Objekt vorzeigt, dessen Form ausgewählte Sachverhalte, also ein Modell, des zu identifizierenden Objektes widerspiegelt. Handelt es sich bei dem zugrundeliegenden Modell um ein Systemmodell, mit dessen Hilfe das Verhalten des System simuliert werden kann, soll das geschaffene Objekt *Modellsystem* genannt werden. Alle anderen Formen wie Zeichnungen, Fotografien, Filme oder Skulpturen, sollen unter dem Begriff *Abbildungen* zusammengefasst werden.

Das *Benennen* von Objekten erfordert, wie Abbildung 2.3 zeigt, sowohl vom Sender als auch vom Empfänger Vorwissen über das zu identifizierende Objekt. Übermittelt der Sender ein Stellvertreterobjekt, ein sogenanntes *Symbol*, an den Empfänger, wird bei diesem die Vorstellung um das zu identifizierende Objekt ausgelöst. Zu diesem

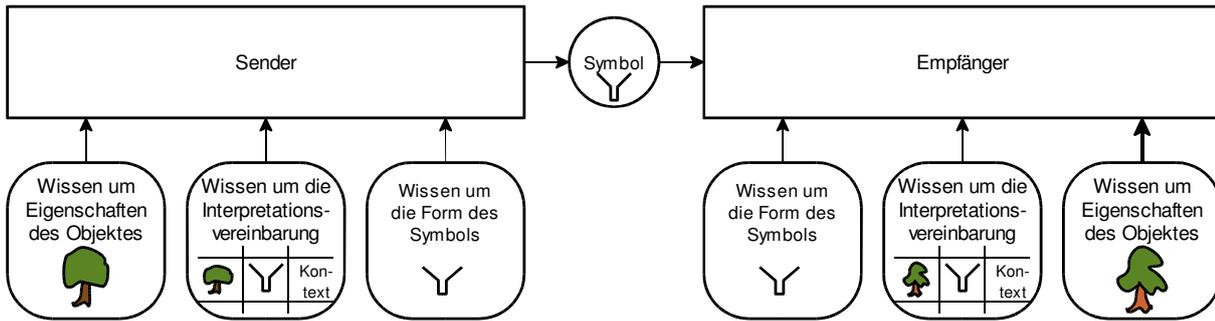


Abbildung 2.3: Benennen von Objekten

Zweck besitzen Sender und Empfänger neben dem Wissen um das zu identifizierende Objekt noch das Wissen um die charakteristische Form des Symbols, sowie das Wissen um die Umstände, den Kontext, in denen das Symbol die gewünschte Vorstellung auslösen soll. Ausgelöst durch die übermittelte Form findet zunächst eine Erkennung des Symbols statt. Im Falle strukturierter Symbole, wie sie aus Buchstaben zusammengesetzte Wörter darstellen, kann diese mehrschrittig ausfallen. Der Prozess, in dem der Empfänger dem erkannten Symbol die Vorstellung um das zu identifizierende Objekt zuordnet, wird als *Interpretation* bezeichnet. Die Vorstellung selbst wird als *Interpretationsergebnis* bezeichnet. Die Form des Symboles kann im Rahmen der *Interpretationsvereinbarung* zwischen Sender und Empfänger frei definiert werden.

Zeigt beispielsweise ein Schiedsrichter (Sender) während eines Fußballspieles (Kontext) einem Spieler (Empfänger) die rote Karte (Symbol), so bedeutet dies den sofortigen Platzverweis (identifiziertes Objekt). Da dem Empfänger das Wissen um das zu identifizierende Objekt definitionsgemäß vorab bekannt sein muss, liegt der Wissensgewinn und damit der Zweck einer solchen Kommunikation darin, dass sie das Objekt, einen Sachverhalt, mit einem bestimmten Zeitpunkt verknüpft. Der Zweck der Kommunikation besteht darin, dem Empfänger mitzuteilen, dass ein bestimmtes Ereignis eingetreten ist. Das Wissen um den Ereignistyp muss der Empfänger bereits besitzen. Aus diesem Grund wird diese Form der Kommunikation auch als *Ereigniskommunikation* bezeichnet.

Das *Umschreiben* von Objekten hat den Zweck Vorstellungen zu vermitteln, die nicht durch Zeigen oder Benennen vermittelbar sind. Dazu werden strukturierte Symbolfolgen, sogenannte *Beschreibungen*, entsprechend den Regeln einer *Sprache* erzeugt. Wie beim Benennen identifizieren die einzelnen Symbole unterschiedliche Objekte. Dabei kann es sich um konkrete oder gedacht konkrete Objekte, sowie um beliebige abstrakte Objekte (Objekt-, Eigenschafts-, Verhaltens-, Beziehungs- und Strukturtypen) handeln. Durch Interpretation, die auf den Regeln der Grammatik und Semantik der Sprache basiert, lässt sich einer Symbolfolge eine *Aussage* zuordnen [Mor38]. Durch eine solche Aussage werden die Objekte, die durch die einzelnen Symbole benannt werden, zueinander in Beziehung gesetzt, wodurch beim Empfänger eine entsprechende Vorstellung geweckt wird. Ein und dieselbe Aussage kann dabei durch unterschiedliche Beschreibungsformen, aufbauend auf unterschiedlichen Begriffen, ausgedrückt werden. Wichtig ist dabei, dass dieselben Objekte identifiziert werden.

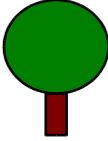
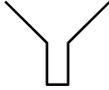
				Baum
Vereinfachte Abbildung	Piktogramm / Symbol	Elementares Symbol		Strukturiertes Symbol
- Strukturähnliche Abbildung charakteristischer Merkmale - Erkennen wahrscheinlich	- Abstrakte Abbildung charakteristischer Merkmale - Erkennen möglich	- Schwache Ähnlichkeit mit abstrakter Darstellung charakteristischer Merkmale - Erkennen unwahrscheinlich	- Willkürlich zugeordnete Form ohne jede Ähnlichkeit mit dem identifizierten Individuum - Erkennen unmöglich	

Abbildung 2.4: Fließender Übergang zwischen Abbildung und Symbol

Eine *Interpretationskette* ergibt sich, wenn man zur Ableitung einer Aussage, die durch eine Symbolfolge identifiziert wird, mehrere aufeinander aufbauende Symbolerkennungs- bzw. Interpretationsschritte durchführen muss. So wird man bei einer Nachricht, die im Morsealphabet codiert ist, eine Folge kurzer und langer Signale wahrnehmen. Rücktransformiert ergibt sich eine Buchstabenfolge, die — zu Wörtern zusammengesetzt — eine Aussage ergibt.

Ohne die Zweckmäßigkeit der obigen Definitionen anzutasten, kann festgestellt werden, dass die Grenze zwischen dem Zeigen einer präsentierten Abbildung und dem Identifizieren durch Benennen nicht in jedem Falle eindeutig ist. Abbildung 2.4 zeigt ein Beispiel dafür. Ein Unterschied ergibt sich aus der Verwendung der identifizierenden Form. Ein Symbol dient, wie zuvor beschrieben, der Ereigniskommunikation oder als Bestandteil einer Beschreibung. Eine Abbildung dient primär zur Vermittlung des abgebildeten Sachverhaltes. Beispielsweise könnte es der Zweck der Abbildung des Baumes sein, etwas über die Form von Krone und Stamm von Laubbäumen zu vermitteln. Abstrakte Abbildungen, sogenannte Piktogramme, werden aber auch zum Identifizieren von Sachverhalten verwendet, in denen die abgebildete Form lediglich eine Komponente oder Eigenschaft einer komplexeren Struktur darstellt. Beispielsweise könnte die Abbildung des Baumes auf einem Schild den Eingang eines Parks seltener Bäume identifizieren. Die Abbildung hätte in diesem Falle eine Doppelfunktion. Sie würde in ihrer Rolle als Symbol den Eingang identifizieren und in ihrer Rolle als Abbildung etwas über das Innere des Parkes vermitteln.

Sender und Empfänger benötigen das gleiche Vorwissen um die Form der verwendeten Symbole, eine ähnliche Vorstellung von den damit identifizierbaren Objekten und das Wissen um die Form und Bedeutung der grammatikalischen Regeln. Somit ergibt sich die Frage: Wie kann eine Gruppe von Menschen sich dieses Wissen erwerben? Von den drei Identifikationsformen kommen offensichtlich nur das Zeigen und das Umschreiben selbst in Frage. Üblicherweise lernt der Mensch eine erste Sprache dadurch, dass er sich durch das Zeigen konkreter Objekte unter gleichzeitiger Benennung oder Umschreibung eine Zuordnung zwischen den konkreten Objekten und Symbolen bzw. Symbolfolgen einprägt. Der erste Umgang mit Sprache wird somit durch „das Zeigen des Gebrauches von Sprache“ vermittelt, indem die Bedeutung konkreter Aussagen mit dem Zeigen der umschreibenden Sachverhalte verknüpft wird. Besitzt ein Mensch

ein gewisses sprachliches Grundrepertoire, kann — darauf aufbauend — durch Umschreibung von Objekten und deren Zuordnung zu neuen bislang unbelegten Formen die Sprache um Symbole erweitert werden. Hierdurch werden Sprache und Vorstellungswelt eines Menschen nahezu untrennbar miteinander verbunden.

Insbesondere bei natürlichen Sprachen ist der Lernprozess bei allen Menschen unterschiedlich. Das heißt, dass trotz der Annahme einer gemeinsamen sprachlichen Basis sich die Vorstellungen von verschiedenen Begriffen um so eher unterscheiden werden, je weniger sie an konkrete Erfahrungen geknüpft sind. Auch aus der Mehrdeutigkeit von Symbolen oder Symbolfolgen können sich Missverständnisse ergeben. Aus diesem Grunde sind bei der Vermittlung abstrakter Sachverhalte anschauliche Abbildungen immer hilfreich, da sich hiermit die Aussagen an konkrete Erfahrungen knüpfen lassen.

Bei formalen Sprachen sollen Missverständnisse durch einen minimalen Sprachkern mit unmissverständlicher Semantik ausgeschlossen werden. Das heißt, dass es im Gegensatz zu natürlichen Sprachen nur eine geringe Anzahl von Begriffen (Terminalsymbolen) und Ableitungsregeln gibt, mit denen man eine unmittelbare Vorstellung verbindet. Um eine praktische Anwendbarkeit zu ermöglichen, lassen sich in den meisten formalen Sprachen neue Begriffe ableiten und so die Sprache erweitern. Hierdurch ist jede Aussage analytisch eindeutig interpretierbar. Eine Anschauung ergibt sich daraus zumeist aber erst mittelbar. Dies hat zur Folge, dass sich in natürlicher Sprache formulierbare Aussagen entweder gar nicht auf Aussagen formaler Sprachen abbilden lassen oder dass die Komplexität vergleichsweise schnell anwächst. Der Zweck formaler Sprachen liegt primär in der rein formalen Ableitung von Konsequenzen, die häufig maschinell geschieht.

Jede Sprache ist durch eine spezifische Begriffswelt gekennzeichnet, auf welche die in dieser Sprache zu formulierenden Aussagen zurückgeführt werden müssen. Die einzelnen Sprachen unterscheiden sich darin, ob eine Aussage formulierbar ist oder nicht bzw. wie aufwendig es ist, diese Aussage zu formulieren. Das Repertoire der durch die Sprache zur Verfügung gestellten Begriffe ist ausschlaggebend für Anschaulichkeit und Komplexität der möglichen Aussagen. Sprachen sind dabei einerseits austauschbares Ausdrucksmittel unserer Vorstellungswelt, andererseits prägen sie infolge stetiger Anwendung durch ihre Begriffswelt unser Denken und damit unsere Modelle. Viele Modelle entstehen folglich mit Bezug auf die Begriffswelt einer vorgegebenen, zuvor erlernten Sprache. Die Abbildung der Begriffswelt auf das Repertoire definierter Symbole und Syntaxregeln, die Notation, ist weitgehend beliebig.

2.2.5 Klassifikation von Systemmodellen und Beschreibungen

2.2.5.1 Analytische versus didaktische Modelle und Beschreibungen

Um ein komplexes System zu beschreiben, müssen in der Regel unterschiedliche Modelle beschrieben werden. Dabei gibt es eine Vielzahl unterschiedlicher Beschreibungsansätze. Zum Teil liegt dies an der Verschiedenheit der Begriffswelt der unterschiedlichen Modelle, zum Teil an der verwendeten Notation. Prinzipiell lassen sich *Be-*

schreibungen didaktischer Modelle und *Beschreibungen analytischer Modelle* unterscheiden [BMW94, Bun98].

Die Vermittlung von Vorstellungen ist das primäre Ziel von Beschreibungen didaktischer Modelle⁴, die deswegen auch *Anschauungsmodelle* genannt werden. Diese Beschreibungen sind gekennzeichnet durch eine hohe Anschaulichkeit und Überschaubarkeit. Ihr Anspruch sollte die optimale Vermittlung von Vorstellungen zwischen Menschen sein. Im Umgang mit komplexen Systemen sind sie ein wichtiges Mittel zur Kommunikation zwischen den Beteiligten. Anschaulich soll eine Beschreibung genannt werden, wenn sie zwei Kriterien erfüllt:

- Das der Beschreibung zugrundeliegende Modell basiert auf einer Objektfindung, die Bezug nimmt auf Systemzweck und Vorwissen der adressierten Empfänger.
- Die Identifizierung der System definierenden Objekte und Beziehungen soll im Hinblick auf bestmögliches Verständnis optimiert sein.

Der erste Punkt setzt ein gemeinsames Grundwissen zwischen Sender und Empfänger voraus, das im Normalfall in einer gemeinsamen natürlichen Sprache zu Ausdruck kommt. Auf dieser Basis können viele — auch komplexe — Objekte nur durch Benennen identifiziert werden. Zu einem Problem wird dies, wenn Sender und Empfänger unterschiedliche Erfahrungen mit einem Begriff verbinden. Auch kann infolge mehrdeutiger Formulierungen oder Aufmerksamkeitsschwankungen irrtümlicherweise ein falsches Objekt identifiziert werden.

Eine Steigerung der Anschaulichkeit erreicht man durch die Verbindung von Beschreibungen mit Mitteln, die eine Identifikation der System definierenden Objekte und Beziehungen durch unmittelbares Zeigen charakteristischer Formen erlauben. So lässt sich in Form einer Grafik der *Aufbau* eines Systems zeigen. Hiermit ist die räumliche Anordnung der Systemkomponenten gemeint, die an den Beobachtungsorten über Schnittstellen verbunden sind. Ebenso lässt sich das *Systemverhalten* bzw. die an den Schnittstellen zulässigen *Wertebereiche* grafisch verdeutlichen. Die getrennte Darstellung von Ablauf-, Aufbau- und Wertebereichsstruktur eines Systems kann sich dabei in vielen Fällen als zweckmäßig erweisen [DeM78, Kel03, Har01]. Allerdings garantiert die Verwendung von Grafik keineswegs das Zustandekommen von Beschreibungen, die zur menschlichen Kommunikation geeignet sind [Pet95]. Ohne das Wissen um die Grundprinzipien der menschlichen Wahrnehmung von grafischen Strukturen kann leicht das Gegenteil erreicht werden.

Unabhängig von der gewählten Form der Darstellung muss die Überschaubarkeit des beschriebenen Modelles gewahrt bleiben. Dies bedeutet, dass die Komplexität dem begrenzten menschlichen Auffassungsvermögen angepasst sein muss und dass demzufolge beispielsweise die Anzahl der in einem Modell erfassten Systemkomponenten nicht beliebig ist.

Systembeschreibungen, die zum Zweck der methodischen Ableitung von Konsequenzen erstellt werden, besitzen eine andere Form und basieren je nach Ableitungszweck

⁴Didaktik ist die Lehre vom Lehren und Lernen.

und Ableitungsart auch auf anderen Modellen. Typische Konsequenzen einer methodischen Ableitung sind Realisierung, Analyse oder Simulation des beschriebenen Systems oder einer seiner Komponenten. Basis hierfür sind die formale Beschreibung und gegebenenfalls weitere bereitzustellende Mittel für eine automatische Bearbeitung. Da eine Methode „ein nach festen Regeln [...] geordnetes Verfahren darstellt“ [Dro89], müssen solche Beschreibungen hinsichtlich der Methode formal korrekt, vollständig und widerspruchsfrei sein, um deren Anwendung sicherzustellen. Aus diesem Grunde basieren solche Beschreibungen in den meisten Fällen auf formalen Sprachen. Form und Begriffswelt müssen primär hinsichtlich der Mittel gewählt sein, die für eine erfolgreiche Ableitung erforderlich sind. Die Verständlichkeit für den Menschen kann unter diesen Anforderungen leiden. Ein häufiger Sonderfall der methodischen Ableitung von Konsequenzen ist die automatisierte Erzeugung bestimmten Verhaltens oder bestimmter Produkte durch Maschinen. Das Einbringen der entsprechenden Beschreibungen in die dafür vorgesehenen Maschinen wird Programmierung genannt. Da Maschinen aber keinerlei Vorstellung besitzen, ist die mit solchen Anschauungen verbundene Vorstellung primär nur eine Hilfe für den Ersteller der Beschreibungen.

Wenn es um die Schaffung komplexer Systeme und damit die Erstellung komplexer Beschreibungen geht, gewinnt die Strukturierung der Beschreibung hinsichtlich einer effizienten Arbeitsteilung eine große Bedeutung. Auch dies geschieht häufig zu Lasten der Anschaulichkeit, da sich Beschreibungsstruktur und Systemstruktur auf komplizierte Art und Weise miteinander vermischen[Kle99, Kel03].

Die Modelle, die solchen Beschreibungen zugrunde liegen, sollen in Abgrenzung zu didaktischen Modellen im folgenden als analytische oder formale Modelle bezeichnet werden. In bestimmten Fällen ist sogar die Beschreibung analytischer Modelle unter Verwendung natürlicher Sprache angemessen. Vorschriften zur Bedienung sicherheitskritischer technischer Anlagen fallen in diese Kategorie.

2.2.5.2 Zweckbezogenes Modell, Realisierungsmodell und Ersatzmodelle

Die Abgrenzung eines dynamischen Systems als Gegenstand der Beobachtung ist immer das Ergebnis einer bewussten Entscheidung. Von Menschen geschaffene oder zu schaffende dynamische Systeme dienen immer einem Zweck. Entweder soll infolge des Systemverhaltens ein beabsichtigter Zustand erreicht werden oder es soll ein beabsichtigter Prozess durchlaufen werden. Vor diesem Hintergrund sind solche Objekte als Bestandteile des Systems anzusehen, die unmittelbaren Einfluss auf das Zustandekommen des beabsichtigten Zustandes bzw. Prozesses haben.

Es gibt kein formales Entscheidungskriterium dafür, was zu einem System hinzu zu zählen ist, und was nicht. Es ist eine willentliche Entscheidung, wobei der Zweck des Systems möglichst klar erkennbar werden muss. Ausgehend vom Zweck des Systems ergibt sich eine Kausalstruktur, die das beabsichtigte Verhalten des System bezüglich der relevanten Beobachtungsorte wiedergibt. Allerdings gibt es im allgemeinen unterschiedliche Mittel, den beabsichtigten Zweck zu erzielen. Abbildung 2.5 zeigt ein Beispiel.

Gezeigt ist ein dynamisches System, in dessen Mittelpunkt ein Gerät zur Warmwasserversorgung steht. Dieses Gerät hat den Zweck aufgrund der Vorgabe von Menge

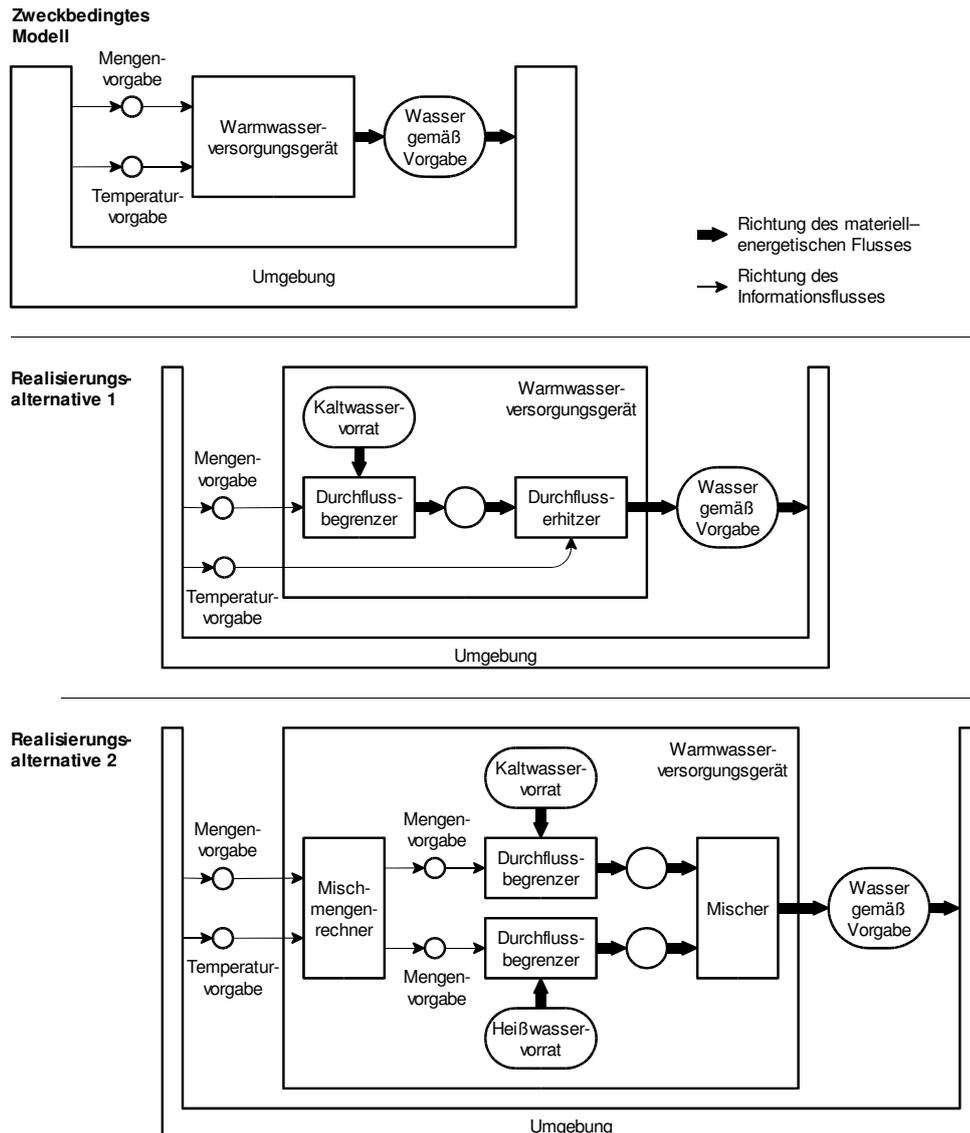


Abbildung 2.5: Beispiel Wasserversorgungssystem

und Temperatur, die es über zwei dafür vorgesehene Schnittstellen aus einer beliebig strukturierten Umgebung erhält, die hierdurch spezifizierte Wassermenge über eine dritte Schnittstelle abzugeben. Der Zweck ist ein klar abgrenzbares gewünschtes Verhalten. Zur Umsetzung dieses Verhaltens ist ein Mittel erforderlich. In diesem Fall ist das Mittel ein Gerät zur Warmwasserversorgung, ein konkret gegenständliches Objekt, welches — eingebettet in seine Umgebung — das gewünschte System ergibt. Die Existenz eines Mittels wird erst durch seine Einbettung in die Umgebung verständlich. Ein Modell, das durch das Bedürfnis begründet ist, den Zweck eines Systems zu verstehen, soll *zweckbezogenes Modell* genannt werden. Da hierbei das Verständnis im Vordergrund steht, sind zweckbezogene Modelle häufig Anschauungsmodelle. Im allgemeinen bestehen dabei nur die zweckbezogenen Modelle sehr einfacher Systeme aus einer unstrukturierten Umgebung und einer einzigen Kernkomponente.

Ein zweckbezogenes Modell legt das Verhalten des Systems bezüglich einer klar definierten Menge von Beobachtungsorten verbindlich fest. Über die interne Realisierung der Mittel, die zur Verwirklichung der zweckbezogenen Kausalbeziehungen erforderlich sind, wird nichts ausgesagt. Die *Rolle des Mittels* kann prinzipiell jedes gegenständliche Objekt übernehmen, das die geforderten Kausalbeziehungen sicherstellt. Was für ein Objekt dies tatsächlich ist und wie die konkrete Form der Schnittstellenwerte im Detail aussieht, ist das Ergebnis einer *Realisierungs- oder Entwurfsentscheidung*, die bei der Schaffung des Systems getroffen wird. So zeigt Abbildung 2.5 zwei unterschiedliche Alternativen, wie das Wasserversorgungsgerät realisiert sein könnte.

Entspricht die Struktur des Systems im Hinblick auf ein Repertoire gegebener Fertigungsbausteine tatsächlich Alternative 1, so heißt das entsprechende Modell *Realisierungsmodell*. In diesem Fall besteht das System tatsächlich aus den Komponenten Durchflussbegrenzer, Durchflusserhitzer und dem Kaltwasservorrat, die entsprechend verbunden sind. Durch ein solches Modell werden gegenständliche Objekte und Beobachtungsorte identifiziert, deren Existenz sich durch eine Analyse des Systems bestätigen würde. Das verbleibende Modell, das eine andere Möglichkeit der Realisierung darstellt, wird *Ersatzmodell* genannt. Ersatzmodelle können außer zur Diskussion von Realisierungsalternativen als Verständnismodelle oder als Grundlage von Simulationen verwendet werden, wenn die tatsächliche Realisierung zu komplex ist. Im obigen Beispiel stellen beide gezeigten Realisierungsalternativen eine *Verfeinerung* des übergeordneten zweckbedingten Modelles dar.

Häufig kommt es jedoch vor, dass ein zweckbezogenes Modell ein Ersatzmodell darstellt. Dies ist immer dann der Fall, wenn eine Komponente des Modells durch ein gegenständliches Objekt realisiert ist, das zu unterschiedlichen Zeiten unterschiedliche Rollen übernimmt. So muss in einem Betrieb mit nur einem Mitarbeiter dieser immer wieder zwischen den Rollen des Verkäufers, Buchhalters, Geschäftsführers usw. wechseln. Zweifelsfrei lässt sich die Funktionsweise dieses Betriebes am einfachsten an einem Modell erklären, das für jede Rolle eine eigene Komponente enthält, anhand derer das Zusammenspiel der unterschiedlichen Funktionen verständlich wird.

Allgemein ist es bei der *Schaffung von Systemen* zweckmäßig, die Realisierung von Komponenten auf ein *vorgegebenes Repertoire* konkreter Objekte zurückzuführen. Hierbei lassen sich grundsätzlich drei Alternativen der Realisierung von Systemkomponenten — ausgehend von einem zweckbezogenen Systemmodell — unterscheiden:

- 1:1-Abbildung zweckbezogener Modellkomponenten auf Realisierungskomponenten
- 1:N-Abbildung zweckbezogener Modellkomponenten auf Realisierungskomponenten: Eine durch das zweckbezogene Modell identifizierte Systemkomponente ist aus mehreren Realisierungskomponenten zusammengesetzt.
- N:1-Zuordnung zweckbezogener Modellkomponenten auf Realisierungskomponenten: Eine Realisierungskomponente wird gleichzeitig oder zeitlich versetzt zu unterschiedlichen Zwecken genutzt. In diesem Fall bezeichnet man die Nutzung der Realisierungskomponente als *Multiplexbetrieb*.

Bungert [Bun98] spricht in diesem Zusammenhang von einem *Betrachtungsebenenwechsel*, wenn das Modell der höheren Betrachtungsebene, das zweckbezogene Modell, als das Ergebnis einer *Interpretation* von Sachverhalten anzusehen ist, die auf einer niedrigeren Betrachtungsebene beobachtet werden. So kann auf einer niedrigen Betrachtungsebene ein System zum Bearbeiten spezieller Linienzugmuster betrachtet werden, das im Bezug auf seinen Zweck als Textverarbeitungssystem anzusehen ist.

2.3 Informationelle Systeme

2.3.1 Charakterisierung informationeller Systeme

Ein informationelles System ist ein dynamisches System, das nur verstanden werden kann, indem man die Wertebereiche oder die Werteverläufe bestimmter Schnittstellen- oder Zustandsvariablen interpretiert [Wen91]. Der Zweck eines informationellen Systems besteht darin, Informationen, die jeweils eindeutig einem der unterschiedlichen Beobachtungsorte des Systems zugeordnet werden können, zu speichern, transportieren und/oder zu verarbeiten. Konsequenterweise dienen die Komponenten eines solchen Systems ebenfalls dem Speichern, Transportieren oder Verarbeiten von Informationen. Aktoren und Sensoren dienen dazu, informationelle Systeme in eine nicht-informationelle Umgebung einzubetten. Im Rahmen dieses Abschnitts soll die Betrachtung jedoch auf den rein informationellen Aspekt solcher Systeme beschränkt bleiben.

Information soll im Rahmen dieser Arbeit ebenfalls in Anlehnung an [Wen91] als gleichbedeutend zum Begriff des Wissbaren verstanden werden. Jede Information stellt somit eine Aussage bezüglich einer Menge von Objekten dar, die entweder in Form einer Benennung oder einer Umschreibung identifiziert wird. Der Informationsträger ist die an einem Beobachtungsort materiell-energetisch wahrnehmbare Form, der sich gemäß einer Interpretationsvereinbarung die Aussage zuordnen lässt (vgl. Abschnitt 2.2.4). Aufgrund des rein informationellen Charakters der Beobachtungsorte ist ein Zeigen von Sachverhalten damit nicht unmittelbar möglich, auch wenn der beschriebene Sachverhalt selbst als Zeiger genutzt werden kann. Beispielsweise könnte eine Information Datum und Ort einer zukünftigen Sonnenfinsternis identifizieren, deren Beobachtung ausgehend von diesem Wissen planbar wäre.

Alle in einem informationellen System beobachtbaren Werte sind als Informationen anzusehen. Da jede Information einen Sachverhalt identifiziert, der entweder wahr oder falsch sein kann, handelt es sich um diskrete Werte. Infolge dessen sind informationelle Systeme angemessenerweise durch *wertdiskrete Systemmodelle* zu erfassen. Aussagen lassen sich nicht fließend durch kontinuierliche Änderung der materiell-energetischen Form des Informationsträgers in immer andere Aussagen überführen. Vielmehr ist die Form, die einer Information zugrundeliegt, ein eindeutig erkennbares Objekt, das sich klar von anderen interpretierbaren Objekten unterscheiden muss. Aus der Diskretheit der beobachtbaren Werte folgt, dass zwei unterschiedliche Werte zeitlich durch ein *Übergangsintervall* getrennt sein müssen. Häufig wird das Ende des Übergangsintervalls als Ereignis angesehen, das eine nachfolgende Aktivität betroffener Komponenten anstößt. Ausgehend von diesen Betrachtungen ergibt sich eine

Vorstellung von wert- und *zeitdiskreten* Systemen, deren Verhalten durch *Ereignisfolgengeflechte* charakterisierbar ist.

Für das unmittelbare Erscheinen einer bestimmten Information an einem Beobachtungsort ist jeweils genau eine Systemkomponente zuständig, die auf diesen Beobachtungsort schreibenden Zugriff hat. Nach einem solchen Zugriff kann diese Information von anderen oder auch derselben Komponente gelesen werden. Ein informationelles System ist daher ein *gerichtetes* System. Am unmittelbaren Zustandekommen eines Wertes sind niemals mehrere Komponenten gleichzeitig beteiligt.

2.3.2 Realisierung informationeller Systeme

Die Begriffswelt informationeller Systeme basiert auf der Vorstellung von Menschen, die Informationen zum Zwecke der Arbeitsteilung, Konfliktlösung oder Unterhaltung untereinander austauschen. Es ist daher zweckmäßig, dass das zweckbezogene Modell eines informationellen Systems ein Netz miteinander kommunizierender Systemkomponenten darstellt. Allerdings ist es eher die Ausnahme, dass alle Komponenten eines solchen Modells tatsächlich durch Menschen realisiert sind. Die Realisierung bestimmter Komponenten durch technische Gebilde erlaubt — im Vergleich zu Menschen — bei vielen Formen der Informationsverarbeitung ein vielfach höheres Maß an Präzision, Speichervolumen und Geschwindigkeit.

Die Unabhängigkeit der Form der Informationsträger von den repräsentierten Sachverhalten hat zur Folge, dass die Werteverläufe an den Schnittstellen informationeller Systeme hinsichtlich ihrer materiell-energetischen Ausprägung unabhängig vom informationellen Zweck des Systems gewählt werden können. Es ist daher möglich, technische Systeme unabhängig von spezifischen informationellen Aufgaben zu konstruieren. Das Schnittstellenverhalten wird erst durch nachträgliche *Programmierung* spezifiziert. Programmierung ist hierbei das Einbringen einer Verhaltenbeschreibung in ein *programmierbares System*, in dessen Folge sich das System gemäß dieser Verhaltensbeschreibung verhält.

Abbildung 2.6 zeigt den Aufbau eines solchen Systems. Gezeigt ist eine Komponente des zweckbezogenen Modells, die Informationen mit ihrer Umgebung austauscht und zu diesem Zweck auf lokale zweckbezogene Operanden zugreifen kann. Realisiert ist diese Komponente durch einen Abwickler, der lesenden Zugriff auf ein Programm hat. Dieses Programm ist eine Beschreibung des gewünschten Verhaltens in einer formalen Sprache, deren Begriffswelt und Syntax durch den Abwicklertyp festgelegt ist. Anhand der Markierungsinformation lässt sich entnehmen, welcher Programmschritt als nächstes auszuführen ist.

In der Regel umfasst das zweckbezogene Modell mehrere Komponenten, die sinnvollerweise mittels eines programmierten Systems zu realisieren sind. Anstelle für jede dieser Komponenten einen separaten Abwickler einzusetzen, ist es möglich, die Verhaltensbeschreibung eines einzigen Abwicklers derart zu formulieren, dass er nacheinander die Rollen unterschiedlicher zweckbezogener Komponenten übernimmt. In Bezug auf das zweckbezogene Modell bezeichnet man diese Nutzung eines Abwicklers als *Rollenmultiplex*. Das zweckbezogene Modell ist in diesem Zusammenhang ein

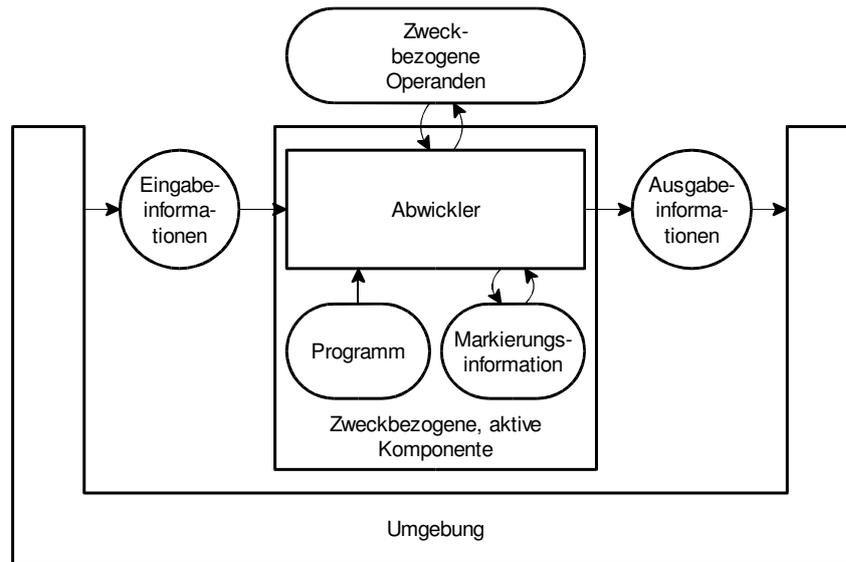


Abbildung 2.6: Abwicklermodell

Ersatzmodell, welches ein *Rollensystem* beschreibt. Das Abwicklermodell ist ein Realisierungsmodell, das in diesem speziellen Fall auch *Trägersystemmodell* genannt wird.

2.3.3 Beschreibung informationeller Systeme

Auch bei der Beschreibung informationeller Systeme ist es zweckmäßig, zwischen Beschreibungen analytischer Modelle zur methodischen Ableitung von Konsequenzen und Beschreibungen zu didaktischen Zwecken zu unterscheiden. Bei der Erstellung analytischer Modelle zur Schaffung informationeller Systeme spielen Programme eine dominierende Rolle. Doch es gibt auch andere Anwendungsgebiete für analytische Modelle. Als Beispiel sei hier die Vorhersage des zu erwartenden Fehlverhaltens genannt. Im folgenden Abschnitt geht es darum, die grundsätzlichen Unterschiede zwischen Programmen und didaktischen Modellen informationeller Systeme zu verdeutlichen.

Besondere Anforderungen an die Beschreibungen informationeller Systeme ergeben sich aus der hohen Komplexität vieler solcher Systeme bzw. der zugrundeliegenden Modelle. Erschwerend für das Verständnis ist, dass sich allein aus der Betrachtung der materiell-energetischen Form eines informationellen Systems der Zusammenhang zu einem zweckbezogenen Modell nur schwer herstellen lässt. Dabei unterscheiden sich die beiden Beschreibungsansätze in ihrem Umgang mit der Komplexität.

Die Komplexität informationeller Systeme äußert sich häufig in der Komplexität der in solchen Systemen zum Einsatz kommenden Programme. Von Programmen ist zu fordern, dass sie das Systemverhalten vollständig und fehlerfrei in der Sprache des Abwicklers beschreiben. Solche Programme umfassen häufig Millionen einzelner Aussagen, so dass die geschickte Strukturierung dieser Programme wesentlich zur Beherrschbarkeit des Entwicklungsprozesses beiträgt. Der Zusammenhang zwischen ei-

ner zweckmäßigen Beschreibungsstruktur und der Struktur des beschriebenen Systems muss nicht immer trivial sein. Umso wichtiger ist es, die Abbildung nachvollziehbar zu gestalten. Zur Schaffung größerer Systeme ist es zweckmäßig, durch die Identifikation von Beschreibungseinheiten, die unabhängig voneinander erstellbar oder austauschbar sind, die arbeitsteilige Entwicklung zu unterstützen. Eine große Rolle spielt dabei die Identifikation von Komponenten-, Verhaltens- und Datentypen, sowie Typ-hierarchien. Dies ermöglicht eine effiziente Beschreibung von Typen unter Bezug auf getrennt beschriebene Typen, die in der Hierarchie übergeordnet sind. In den Klassenhierarchien objektorientierter Programme findet dieses Prinzip seine Anwendung. Der an Details interessierte Leser sei an dieser Stelle auf die einschlägige Fachliteratur verwiesen. Ausgewählte Ansätze der Programmstrukturierung speziell im Zusammenhang mit der Schaffung interaktiver Systeme werden in Kapitel 4.3 behandelt.

Die arbeitsteilige Schaffung eines Systems setzt eine gemeinsame Vorstellung aller Beteiligten voraus. Didaktische Beschreibungen dienen dazu, diese Vorstellungen zu kommunizieren und in der Diskussion mit anderen zu präzisieren. Sie helfen dabei, die Komplexität informationeller Systeme beherrschbar zu machen, indem sie überschaubare anschauliche Modelle des Systems vermitteln. Solche Modelle können auch dazu dienen, Benutzern den Betrieb bzw. die Anwendung von Systemen zu ermöglichen. Im Hinblick auf die tatsächliche Realisierung des Systems wird es sich häufig um Ersatzmodelle handeln.

Ausgewählte Aspekte des Systems werden auf einer relativ hohen Betrachtungsebene beschrieben. Im allgemeinen ist es möglich, durch eine überschaubare Anzahl von Anschauungsmodellen alle Strukturen zu vermitteln, die für ein Verständnis des Systems wesentlich sind. Welche Strukturen dies sind, ist davon abhängig, welchem Zweck die Systemvorstellung dienen soll. Beispielsweise kann es für den Anwender eines CAD-Systems zur Konstruktion von Motoren hilfreich sein, die Einbindung des Systems in das Fertigungssystem des Unternehmens zu verstehen. Der konkrete Aufbau des Firmen-internen Rechnernetzwerkes ist für seine Aufgaben bedeutungslos.

Es existiert eine Vielzahl von Ansätzen zur Modellierung informationeller Systeme, die sich in Zielsetzung, Begriffswelt, Komplexität und Notation voneinander unterscheiden. Neuere Entwicklungen streben ein Schließen der Kluft zwischen der Semantik von Programmiersprachen und der Begriffswelt anwendungsspezifischer Modelle durch sogenannte Modellierungssprachen an. Zweck dieser Modellierungssprachen ist es, sowohl die anschauliche Vermittlung von System- und Beschreibungsstrukturen durch eine grafischen Darstellung als auch die Generierbarkeit abwickelbarer Programme durch Formaltransformationen zu ermöglichen. Es würde den Rahmen dieser Arbeit sprengen, diese Ansätze daraufhin zu untersuchen, inwiefern das primäre Ziel der Programmerstellung Einfluss auf die didaktische Qualität solcher Modelle hat. Allerdings muss bedacht werden, dass ein programmiertes System niemals Selbstzweck ist. Ohne Berücksichtigung der Einbettung in seine Umgebung ist es nicht zu verstehen. Ein Program beschreibt demzufolge hinsichtlich des Systemzweckes nur das Verhalten einer Systemkomponente. Eine didaktische Beschreibung muss aber den Systemzweck und somit das gesamte System verständlich erklären.

Das Spektrum an — im weitesten Sinne — didaktischen Beschreibungen reicht von umgangssprachlichen textuellen Beschreibungen von Anwendungsszenarien über

spontane Skizzen hin zu formalen grafischen Notationen. Zur Visualisierung realisierungsnaher Strukturen auf Programmebene findet die Modellierungssprache UML [BRJ99] zunehmende Verbreitung. Dabei dominiert die Verwendung von Klassendiagrammen. Auf konzeptioneller Ebene gibt es derzeit noch keinen erkennbaren Standard. Defizite in der Modellierungs- und Darstellungspraxis sind keine Ausnahme [Kel03]. Ein tragfähiger Ansatz, der seine Praxistauglichkeit bereits in zahlreichen Projekten mit großen internationalen Softwareunternehmen sind die sogenannten *Fundamental Modelling Concepts*. Unter anderem wurden gemeinsam mit SAP und Siemens mehrere komplexe Softwaresysteme modelliert. Der Ansatz basiert auf den langjährigen Arbeiten von Siegfried Wendt und seinen Mitarbeitern [Wen80, Wen91, Bun98, Kle99, Tab00, Wen01] und stellt die Grundlage der Systemsichtweise und -darstellung dar, die in dieser Arbeit durchgängig zum Ausdruck kommt.

2.3.4 Fundamental Modelling Concepts

Zweck der Fundamental Modelling Concepts (FMC) ist die bestmögliche Unterstützung der Kommunikation zwischen Menschen über jegliche Art informationeller Systeme mittels didaktischer Modelle. Der Ansatz beruht auf einer in sich geschlossenen Begriffswelt, einem Metamodell, das in der Anschaulichkeit dynamischer Systeme verwurzelt ist. Er vereint eine minimale Auswahl von theoretisch abgesicherten und praktisch bewährten Ansätzen zur Beschreibung der komplementären Basisaspekte eines jeden informationellen Systems. Das Metamodell von FMC lenkt das Nachdenken über informationelle Systeme und hilft so den Beteiligten beim Finden und Kommunizieren von Systemstrukturen. FMC bedient sich einer grafischen Notation, die didaktischen Prinzipien folgt und an etablierte Darstellungsformen angelehnt ist. Die folgende Zusammenfassung soll dem Leser die wesentlichen Konzepte des Ansatzes vermitteln.

Jedes informationelle System kann durch ein Systemmodell definiert werden, das im Bezug auf eine Betrachtungsebene vollständig durch die getrennte Betrachtung seiner drei komplementären Aspekte

- Aufbaustruktur,
- Ablaufstruktur und
- Wertebereichsstruktur

beschrieben werden kann. Entsprechend werden zur Darstellung der Systemstrukturen drei korrespondierende Plantypen unterschieden:

- Aufbaudiagramme (Blockdiagramme)
- Petri-Netze und
- Entity-Relationship-Diagramme.

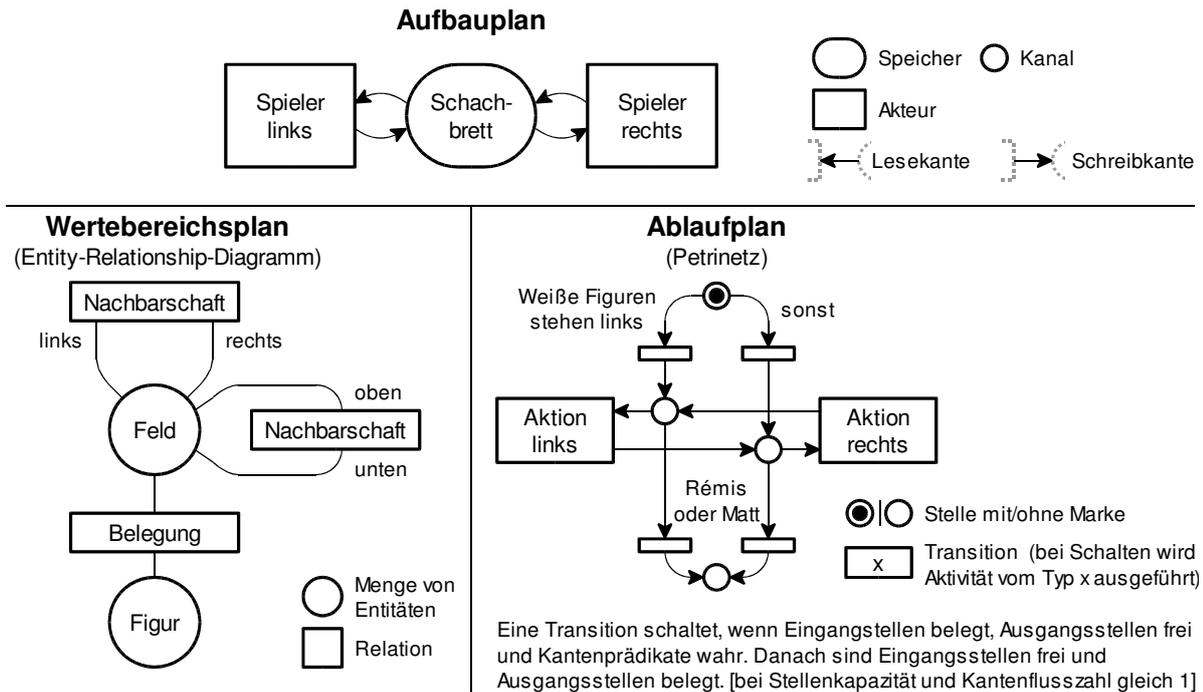


Abbildung 2.7: Beispiel für die drei Strukturplantypen eines Systemmodells (nach [Wen01])

Ein Kennzeichen von FMC ist die konsequente Trennung aktiver und passiver Aufbau-komponenten. Die Aufbaustruktur legt einerseits die Orte eines Systems fest, an denen Informationen beobachtbar sein sollen (Orte sind passive Systemkomponenten, wobei Speicher und Kanäle unterschieden werden). Andererseits legt die Aufbaustruktur fest, welche Akteure im System vorkommen und auf welche Speicher und Kanäle des Systems sie lesend und/oder schreibend Zugriff haben. Akteure sind die aktiven Komponenten des Systems. Eine Besonderheit in der Aufbaumodellierung ist die Möglichkeit, Strukturvarianz, das heißt das Entstehen oder Verschwinden von Systemkomponenten, darstellen zu können.

Das Verhalten des Systems lässt sich im Bezug auf eine Aufbaustruktur als das Ereignis-folgenrechteck der vorkommenden Schreib- und Lesezugriffe definieren. Die Kausalstruktur, die einem solchen Geflecht zugrunde liegt, kann durch ein Petri-Netz beschrieben werden. Die Darstellung nebenläufigen Verhaltens ist möglich. Im allgemeinen ist es zweckmäßig, eine Partitionierung des Gesamtzustandes eines Systems in seinen Steuerzustand und seinen operationellen Zustand (die Unterscheidung zwischen Steuerdaten und operationellen Daten) vorzunehmen. Steuerzustände sind solche Zustände des Systems, aus welchen sich die Reihenfolge der auszuführenden Operationen im System ergibt und deren Bedeutung nur durch die Einbettung in eine Kausalstruktur verständlich wird. Hiervon zu unterscheiden ist der operationelle Zustand des Systems, der Gegenstand aller im Rahmen der Kausalstruktur definierten Operationen ist. Während Speicher für operationelle Daten grundsätzlich im Aufbau gezeigt werden, ist es im allgemeinen üblich, die Speicher für den Steuerzustand als Bestandteil der Akteure zu sehen und daher nicht explizit zu zeigen.

Das Repertoire möglicher Belegungen für Speicher von operationellen Werten wird

— anders als bei Steuerzustandsspeichern — typischerweise nicht explizit aufgezählt. Beispielsweise ist die Anzahl möglicher (Steuer-)Schritte zur Erfassung eines Beleges in einem konkreten Buchhaltungssystem explizit aufzählbar. Das Repertoire denkbarer Belege in einem solchen System ist praktisch unbegrenzt. Deshalb wird das Repertoire operationeller Werte für einen Speicher durch die Beschreibung des zulässigen Wertestrukturbereiches festgelegt. Dies geschieht in Form eines Entity-Relationship-Diagrammes⁵.

FMC erlaubt die hierarchische Verfeinerung von Aufbau-, Ablauf- und Wertestrukturen. Bei einem solchen Verfeinerungsschritt wird eine innere Struktur der Strukturelemente der übergeordneten Ebene sichtbar. Die Grenzen übergeordneter Strukturelemente verlieren im verfeinerten Modell nicht ihre Semantik. Im Hinblick auf die übergeordnete Struktur ist eine eindeutige Rückabbildung möglich. Der Strukturtyp — Aufbau, Ablauf oder Wertestruktur — bleibt bei einer Verfeinerung erhalten, so dass zusätzliche Diagrammtypen nicht erforderlich sind.

Zusätzlich beinhaltet FMC das Konzept des Betrachtungsebenenwechsels. So ist es möglich und häufig zweckmäßig, ein System dessen Aufbau durch eine bestimmte Struktur aus Komponenten, das Rollensystemmodell, definiert ist, auf einer tieferen Betrachtungsebene durch eine andere Struktur, das Trägersystemmodell, realisiert zu betrachten. Durch Verfeinerung der ursprünglichen Struktur ließe sich dieses Ergebnis nicht gewinnen (vgl. Kapitel 2.3.2). Die Semantik des übergeordneten Modells ist auf dieser Ebene nicht mehr direkt erkennbar, sondern muss durch Interpretation der beobachtbaren Sachverhalte gewonnen werden.

⁵Eine formal präzise Definition des zulässigen Repertoires ist in manchen Fällen nur mittels prädikatenlogischer Ausdrücke möglich. Zur Erhöhung der Verständlichkeit tragen solche Ausdrücke jedoch nur selten bei.

Kapitel 3

Interaktive Systeme

3.1 Begriffe

Die Kommunikation über interaktive Systeme setzt ein gemeinsames Verständnis von wenigen Grundbegriffe zwischen den Beteiligten voraus. Im Zentrum stehen dabei die Begriffe

- interaktives System,
- Benutzer und
- Benutzungsschnittstelle.

Leider gibt es für diese Begriffe keine einheitliche Definition. Dies ist umso problematischer, weil die meisten Leser wahrscheinlich kein Problem haben werden, spontan mit jedem dieser Begriffe eine mögliche Bedeutung zu assoziieren, die sich nicht unbedingt mit der Intention des jeweiligen Autors decken muss. Trotzdem wird in vielen Veröffentlichungen völlig auf eine Definition der Begriffe verzichtet.

Der Begriff des interaktiven Systems wird in der einschlägigen Literatur primär in zweierlei Bedeutung verwendet. In seiner ersten Bedeutung ist ein interaktives System ein System, in dem explizit ein Mensch als Komponente vorkommt, der mit dem übrigen System interagiert [Sal95]. Nach dieser Sichtweise wäre in Abbildung 3.1 die mit dem Kürzel IS gekennzeichnete Struktur als interaktives System zu betrachten. In seiner zweiten Bedeutung umfasst ein interaktives System lediglich die mit IAS bezeichnete Systemkomponente [BC92]. Die unterschiedliche Verwendung des Begriffes legt den Schluss nahe, dass es je nach Kontext wünschenswert ist, entweder die gesamte Struktur IS oder nur die Komponente IAS eindeutig zu benennen. In Veröffentlichungen in denen eine begriffliche Unterscheidung getroffen wird, wird in der Regel die gesamte Struktur als interaktives System und die Komponente IAS als das Computersystem bezeichnet [Joh92, Sta96]. So ist in einigen Teilen der einschlägigen Literatur die Realisierung der Komponente IAS durch einen Computer scheinbar ein kennzeichnendes Merkmal eines interaktiven Systems. In anderen

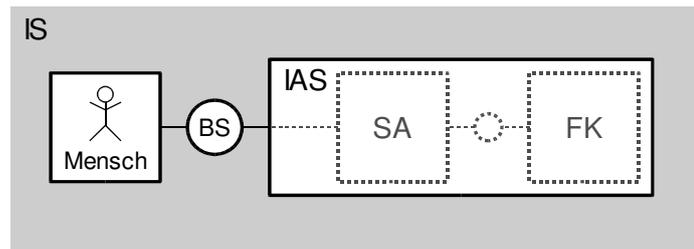


Abbildung 3.1: Interaktives System

Veröffentlichungen wird dies jedoch nicht explizit gefordert oder ganz auf eine Definition verzichtet. Eine Veröffentlichung, in der die Gesamtstruktur IS von der Komponente IAS durch zwei getrennte realisierungsunabhängige Begriffe unterschieden wird, ist dem Autor nicht bekannt.

Als Benutzer eines interaktiven Systems werden üblicherweise die Menschen bezeichnet, die das System unmittelbar benutzen. In Abgrenzung hierzu unterscheidet man die Anwender, die das System indirekt, das heißt, vermittelt durch Interaktionen mit den eigentlichen Benutzer, nutzen. Unter dem System wird dabei in der Regel die durch ein Computersystem realisierte Komponente IAS verstanden. In manchen Arbeiten findet man anstelle des Begriffs Benutzer auch den Begriff des End-Benutzers (engl. end user), wodurch der Begriff des Benutzers dann auf die Ebene des Anwenders gerückt wird (siehe Abschnitt 3.4.2). Manche Veröffentlichungen [BC92, Sta96] gehen von genau einem Benutzer eines interaktiven Systems aus, andere [Joh92, Sal95] lassen dies offen und wieder andere gehen explizit von mehreren gleichzeitig agierenden Benutzern aus [Dew95, CCN97].

Der Begriff der Benutzungsschnittstelle (engl. user interface) wird ebenfalls in unterschiedlicher Bedeutung verwendet. Prinzipiell handelt es sich bei der Schnittstelle um eine räumlich definierte Grenzfläche zwischen einem Benutzer und dem interaktiven System, über welche die Nutzung des Systems durch den Benutzer erfolgt. Diese Schnittstelle ist in Abbildung 3.1 durch den Kanal BS repräsentiert. Eine Benutzungsschnittstelle ist gekennzeichnet durch die Erscheinungsform aller zur Interaktion zwischen Benutzer und System verfügbaren Ein- und Ausgabegeräte des interaktiven Systems. Dennoch heißt es in [Joh92]: „Die Schnittstelle zwischen dem Benutzer und dem Computer ist unklar. Es gibt nicht nur die physikalische Trennung zwischen Mensch und Maschine“ [Übersetzung des Autors]. Nach Johnson muss das Wissen der Benutzer um die Regeln zur Interpretation der Schnittstellenformen und zur Durchführung der Interaktionsfolgen ebenfalls als Teil der Benutzungsschnittstelle angesehen werden.

Im Bereich des Software-Engineering wird der Begriff vielfach in nochmals anderer Bedeutung verwendet. Dort stellt das „User Interface“ eine aktive Systemkomponente dar, die als eine Art Schnittstellenadapter (SA) angesehen werden kann, der zwischen dem Benutzer und dem sogenannten funktionalen Kern (FK) vermittelt. Dieser Kern wird als die Komponente des Systems betrachtet, welche „die Konzepte bzw. das Modell des Anwendungsbereiches des Systems implementiert“ [BC92, Sal95].

Aufgrund der zentralen Rolle dieser Begriffe ist es erforderlich, ihre Bedeutung im Rahmen dieser Arbeit eindeutig festzulegen:

Ein *Interaktionssystem* ist ein dynamisches System, in dem ein oder mehrere Menschen miteinander oder mit technischen Systemkomponenten Informationen austauschen. Der Zweck, der den Interaktionen zwischen den beteiligten Komponenten zugrunde liegt, kann informationeller oder materiell-energetischer Natur sein. Die Möglichkeit der mehrschrittigen bzw. fortschreitenden Willensformulierung zur Erzielung des beabsichtigten Zweckes durch die Benutzer ist kennzeichnendes Merkmal interaktiver Systeme.

Als die Komponenten eines Interaktionssystems sollen genau die Objekte bezeichnet werden, deren Existenz mit dem Zweck des Systems untrennbar in Beziehung steht. Hierbei sind die *Benutzer*¹, von den übrigen Komponenten, die Mittel zum Zweck sind und in ihrer Gesamtheit das *interaktive System* bilden, zu unterscheiden. Ein Interaktionssystem entspricht somit der in Abbildung 3.1 gezeigten Gesamtstruktur IS, deren Komponente IAS das interaktive System repräsentiert.

Ein Flugsimulator ist ein Beispiel für ein interaktives System. Sein Zweck ist informationeller Natur. Der Simulator erweckt in einem Benutzer die Vorstellung ein Flugzeug zu fliegen, während er vor einem Bildschirm sitzt, auf dem die Instrumente des Cockpits eines simulierten oder erdachten Flugzeuges und der Blick aus dem Cockpit in die ebenfalls simulierte oder erdachte Umgebung dargestellt sind. Mittels einer Konsole von Bedienelementen lässt sich das erdachte Flugzeug steuern, so dass der Benutzer unmittelbar erkennen kann, wie das Flugzeug reagiert.

Bei einer nicht-interaktiven Simulation könnte ein Benutzer nach dem Start der Simulation keinen Einfluss mehr auf den Simulationsprozess nehmen. Alle Vorgaben zur Steuerung des Flugzeuges müssten bereits vor dem Start formuliert worden sein. Einen solchen Simulator wird man daher nicht als interaktives System bezeichnen.

In beiden Fällen dient der Simulator jedoch dazu, eine Vorstellung im Kopf seiner Benutzer zu erzeugen und nicht etwa Materie oder Energie zu transportieren, zu verarbeiten oder zu speichern. Der Zweck ist somit informationeller Natur. Würde mithilfe der Konsole ein reales Flugzeug zum Transport von Personen gesteuert werden, so wäre der Zweck des Systems materiell-energetischer Natur.

Die Formulierung des Gewollten wird bei interaktiven Systemen nicht als einmaliges Ereignis angesehen. Stattdessen hat man die Vorstellung eines *Interaktionsprozesses*. Bei einem solchen Prozess tauschen Benutzer und interaktives System solange miteinander Informationen aus, bis das angestrebte Ergebnis oder Verhalten erreicht ist. Es ist dieser Dialogcharakter der Wechselwirkungen zwischen Benutzer und interaktivem System [Joh92], der interaktive Systeme als informationelle Systeme ausweist. Da Informationen prinzipiell beliebig codiert werden können (solange Sender und Empfänger die verwendete Sprache verstehen), ergibt sich daraus eine weitgehende Gestaltungsfreiheit bei der Festlegung der Form zur Formulierung der Benutzerwünsche und Systemnachrichten. Die Verwendung programmierter Komponenten bzw. der Einsatz von Computersystemen bei der Realisierung interaktiver Sys-

¹Prinzipiell sind außer Menschen auch andere Lebewesen als potentielle Interaktionspartner in einem Interaktionssystem denkbar.

teme ist somit kein kennzeichnendes Merkmal. Er erweist sich aber in vielen Fällen aufgrund des informationellen Charakters zumindest der Benutzungsschnittstelle als zweckmäßig (siehe Kapitel 2.3.2).

Eine Mehrschrittigkeit von Benutzungsprozessen ergibt sich im Falle ergebnisorientierter Systeme häufig bereits aus der natürlichen Begrenztheit eines jeden Repertoires an Symbolen mit deren Hilfe das Gewollte formuliert werden muss. Eine komplexe Bestellung kann nicht in einem Wort formuliert werden. Bei einem prozessorientierten System erfordert der Wunsch nach einer fortschreitenden Beeinflussung des Prozesses fortschreitende Anweisungen an das System. Der typische Dialogcharakter interaktiver Systeme entsteht jedoch durch die Art und Weise, in der interaktive Systeme ihre Benutzer bei der mehrschrittigen oder fortschreitenden Formulierung des Gewollten idealerweise unterstützen. Hierzu gehört, dass ein interaktives System seine Benutzer jederzeit mit den im jeweiligen Kontext relevanten Informationen versorgen sollte oder ein Benutzer Aufgaben zurückstellen und dringlichere Aufgaben vorziehen oder Hilfe anfordern kann. Es sollte möglich sein, erst teilweise formulierte Wünsche zu korrigieren und jederzeit auf die Aktivitäten des interaktiven Systems Einfluss zu nehmen.

Der zum Austausch von Informationen zwischen Benutzer und interaktivem System erforderliche Kanal bildet die sogenannte *Benutzungsschnittstelle*, die in Abbildung 3.1 durch den mit BS benannten Kreis symbolisiert ist. Hierbei wird der Begriff in seiner oben vorgestellten Bedeutung als räumlich definierte Grenzfläche zwischen einem Benutzer und dem interaktiven System verwendet. Sie ist durch die Erscheinungsform aller zur Interaktion verwendbaren Ein- und Ausgabegeräte gekennzeichnet. Dies schließt Geräte zur akustischen, haptischen oder sonstwie gearteten Ein- und Ausgabe ein. Aufgrund des informationellen Charakters dieser Schnittstelle sind die dort beobachtbaren Formen prinzipiell interpretationsbedürftig. Deshalb ist es vielfach zweckmäßig, eine Benutzungsschnittstelle auf unterschiedlichen Betrachtungsebenen zu beobachten. Dies ändert jedoch nichts an dem prinzipiellen Schnittstellengedanken der Systemtheorie. Bei manchen interaktiven Systemen mit materiell-energetischem Systemzweck mag es sinnvoll sein, auch bestimmte Kanäle, deren Zweck materiell-energetischer Natur ist, als Bestandteil der Benutzungsschnittstelle anzusehen. Als Beispiel sei hier das Ausgabefach eines Getränkeautomaten genannt. Die Realisierung solcher Komponenten wird im Rahmen dieser Arbeit jedoch nicht betrachtet.

Ob eine Schnittstelle eine dem Zweck des interaktiven Systems angemessene Benutzung zulässt, ist eine Frage der Gestaltung und hängt wesentlich von den Möglichkeiten der Benutzer des Systems ab. Die Frage, ob ein System mit einer Benutzungsschnittstelle interaktive Benutzungsprozesse ermöglicht oder ausschließt, hängt — nach obigen Definitionen — sogar ausschließlich von der Gestaltung der Schnittstelle und nicht vom Zweck des Systems ab.

Wichtiger als die Frage nach einer eindeutigen Klassifikation solcher Systeme in interaktive und nicht-interaktive erscheint dem Autor ein ausgeprägteres Bewusstsein bei den Entwicklern solcher Systeme für die Möglichkeiten einer angemessenen Gestaltung von Benutzungsschnittstellen. Die sich anschließenden Betrachtungen gelten daher ganz allgemein für Systeme mit informationellen Benutzungsschnittstellen. All diese Systeme sollen im folgenden als interaktive Systeme bezeichnet werden. Auf sie beziehen sich die folgenden Betrachtungen.

3.2 Primäre Modellebenen interaktiver Systeme

In der einschlägigen Literatur [Mor81, FD82, Sta96, Shn98, PK03] hat sich die Ansicht verbreitet, dass es zweckmäßig ist, bei der Beschreibung interaktiver Systeme Ebenen aufeinander aufbauender Entwurfsentscheidungen zu unterscheiden. Einigkeit besteht darüber, dass man den Zweck eines interaktiven Systems und das Repertoire der dazu bereitgestellten Funktionen unabhängig von den Interaktionsprozessen und Formen, die mit der Inanspruchnahme der Funktionen an der Benutzungsschnittstelle verbundenen sind, beschreiben soll. Diese Trennung lässt sich an allen gängigen Ansätzen zur Schaffung interaktiver Systeme nachvollziehen. Die Ansätze selbst unterscheiden in ihren Wegen von einer Beschreibung der abstrakten Ebene gewünschter Funktionen hin zur Spezifikation der konkreten Interaktionsprozesse und Formen.

Im allgemeinen entsteht aus einer Menge lose zusammenhängend formulierter Anforderungen, die grob die gewünschte Funktionalität des Systems umschreiben, ein Hierarchiemodell der Benutzeraufgaben. In dem daraus entstehenden Themenplan werden die konzeptionellen Objekte und Operationen benannt, klassifiziert und zueinander in Beziehung gesetzt sind, die zur Durchführung der Aufgaben für einen Benutzer erforderlich sind. Die so umschriebene Struktur wird häufig als *konzeptionelles Modell* oder *Domänenmodell* bezeichnet. Hierauf aufbauend werden die konzeptionellen Objekte und Operationen auf Interaktionsobjekte und Aktionen konkreter Erscheinung, die materiell-energetische Form der Benutzungsschnittstelle, abgebildet. Formal wird dabei üblicherweise das konzeptionelle Modell um die Objekte, die zur Interaktion erforderlich sind, in einem oder mehreren Schritten „angereichert“. Das so entstandene Modell dient als Grundlage für eine Realisierung, die zumeist auf der Basis einer geeigneten als gegeben vorausgesetzten Softwarearchitektur erfolgt. Hierbei erwecken einige Ansätze den Eindruck, dass ein angestrebtes Ziel in der Schaffung einer allgemein anwendbaren Plattform zu sehen ist, die eine Ausführung beliebiger Modelle interaktiver Systeme erlaubt [Sta00].

Vor dem Hintergrund des Verständnisses von dynamischen Systemen im Sinne von Kapitel 2 ergibt sich die Frage, inwieweit Modelle die Vorstellung von einem Interaktionssystem als Struktur aus wechselwirkenden gegenständlichen Aufbaukomponenten reflektieren, die über die in Abbildung 3.1 gezeigte Trivialstruktur hinausgeht. Nach Young lassen sich prinzipiell aufbaubasierte Modelle, sogenannte „Surrogates“, von aufgabenbasierten Modellen, sogenannten „Mappings“ [You83] unterscheiden. Surrogate hat im Englischen die Bedeutung von Ersatz oder Stellvertreter. In diesem Sinne lässt sich Surrogate im Kontext von Modellen am ehesten mit Anschauungsmodell oder Ersatzmodell übersetzen. Surrogates wird eine bessere Eignung zugeschrieben, ein Verständnis des betrachteten Systems zu erzielen. Hingegen lassen sich anhand von Mappings besser Aspekte wie Erlernbarkeit und Benutzbarkeit beurteilen. Eine aufbaubasierte Vorstellung kommt in Mappings nicht zum Ausdruck (vgl. Abschnitt 4.2.3).

Ohne den Nutzen aufgabenbasierter Modelle infrage stellen zu wollen, können demnach in der Aufbauvorstellung verwurzelte Modelle eine verständnisfördernde Grundlage der Betrachtung interaktiver System durchgängig über alle Modellebenen bewirken. Dazu gehören:

- eine Betrachtung des gesamten Interaktionssystems,
- die Klärung von Zuständigkeiten zwischen den Akteuren sowie
- die räumliche Zuordnung von Bearbeitungsgegenständen.

Die klare Unterscheidung zwischen Modellen des gesamten Interaktionssystems und Modellen des interaktiven Systems, deren Fehlen bei manchen Entwicklungsansätzen zu Verwirrung führt, ergibt sich bei Modellen, die in der Aufbauvorstellung verwurzelt sind, zwangsläufig. Die Realisierung interaktiver Systeme auf der Basis vorgegebener Trägersysteme ist in vielen Ansätzen völlig ausgeklammert, so dass der Bezug zu Architekturmodellen nicht klar wird. Ein solcher Bezug lässt sich durch Modelle, die in der Aufbauvorstellung verwurzelt sind, leicht herstellen.

Um weitere Ausführungen dieser Behauptung in Kapitel 4 zu ermöglichen, wird im folgenden ein streng systembasierter Ansatz zur Modellierung interaktiver Systeme vorgestellt. Dabei wird gezeigt, wie eine konsequente Anwendung des Begriffs dynamischer Systeme zur Findung von Modellen interaktiver Systeme mit eindeutig abgrenzbaren Betrachtungsebenen führen kann.

Im Kern dieses systemzentrierten Ansatzes steht, wie Abbildung 3.2 zeigt, die Abgrenzung von drei primären Modellebenen, die zweckmäßigerweise bei der Beschreibung interaktiver Systeme unterschieden werden sollten. Auf jeder dieser Modellebenen wird das System als Gebilde aus aktiven Komponenten betrachtet, die auf informationellen Speichern operieren und miteinander über Kanäle kommunizieren. Jedes dieser Modelle erfasst die Aufbau-, Verhaltens- und Wertestruktur des interaktiven Systems unter Berücksichtigung der Aspekte, die für die jeweilige Ebene charakteristisch sind. Häufig ist es zweckmäßig, mehrere Systemmodelle pro Ebene zu erstellen, die sich in Ausschnitt, Auflösung und Form unterscheiden:

- Auf der *Ebene zweckbezogener Modelle* wird das Interaktionssystem im Hinblick auf genau die Eigenschaften und Strukturen betrachtet, die durch den Zweck des interaktiven Systems bedingt sind.
- Auf der *Ebene interaktionsbezogener Modelle* werden die Eigenschaften und Strukturen des Interaktionssystems betrachtet, die zeigen, *wie* das bezweckte Verhalten des interaktiven Systems durch die Wahl konkreter informationstragender Formen an der Benutzungsschnittstelle erreicht wird.
- Auf der *Ebene realisierungsbezogener Modelle* wird das interaktive System vollständig im Hinblick auf seine Realisierung unter Verwendung eines vorgegebenen Repertoires an Realisierungskomponenten beschrieben.

Jede Ebene ist durch mindestens ein spezifisches Systemmodell gekennzeichnet. Der Übergang zwischen zwei primären Modellebenen ist als Ergebnis eines Betrachtungsebenenwechsels anzusehen. So stehen die identifizierten Komponenten zweier Ebenen zwar miteinander in Beziehung, sind aber nicht durch hierarchische Verfeinerung bzw. Vergrößerung aufeinander abbildbar. Häufig ist es zweckmäßig, innerhalb der Ebene

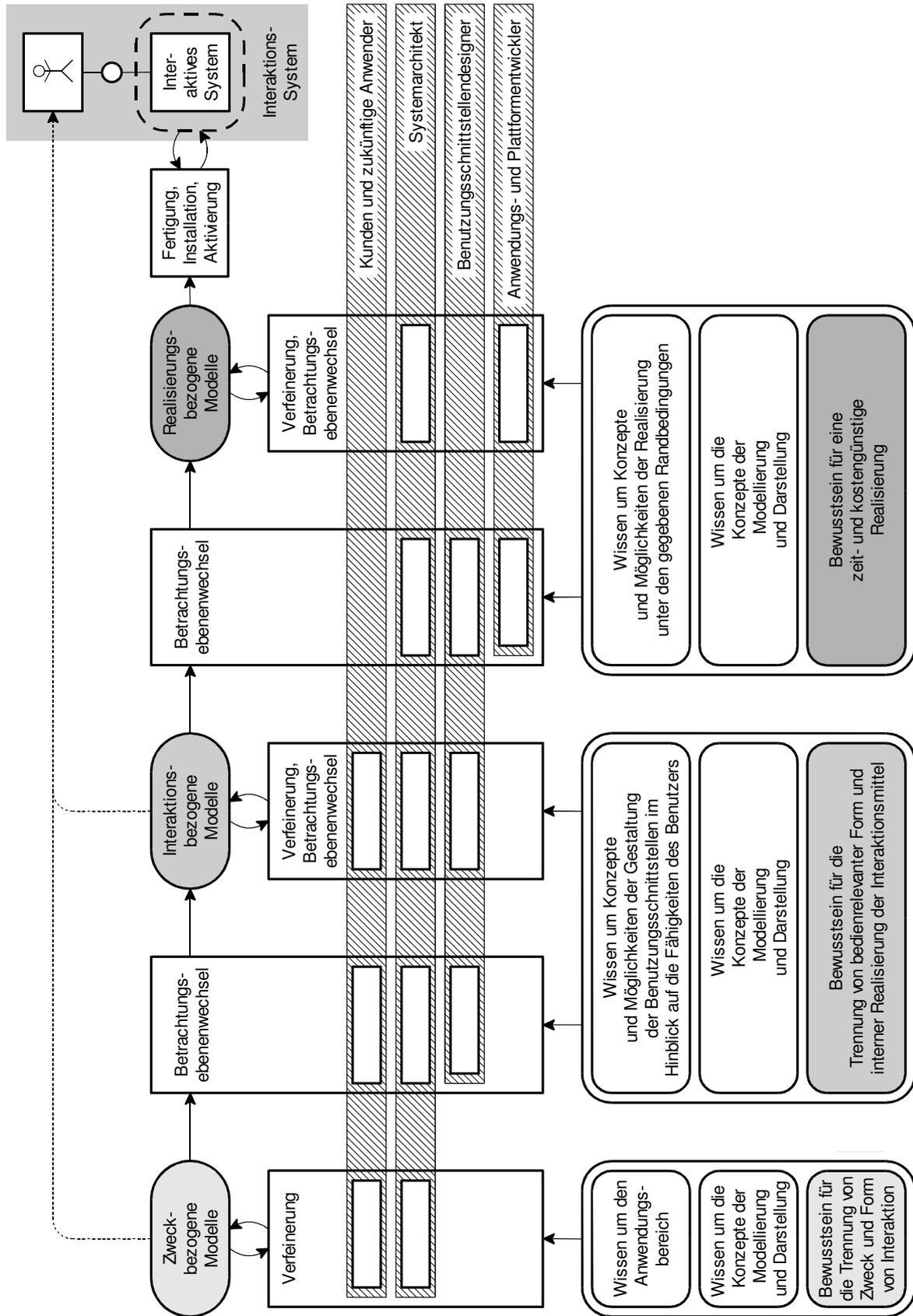


Abbildung 3.2: Ebenenmodell interaktiver Systeme

interaktionsbezogener bzw. realisierungsbezogener Modelle weitere Betrachtungsebenen voneinander zu unterscheiden. Die grundsätzliche Abgrenzung von drei primären Modellebenen berührt dies jedoch nicht.

Die Abgrenzung der drei primären Modellebenen basiert auf der Charakterisierung des Wissens, das zur Modellbildung erforderlich ist, der Definition der Übergänge zwischen den Betrachtungsebenen, sowie der Identifikation der Personengruppen, die an der Entstehung und Nutzung der jeweiligen Modelle beteiligt sind. Das erforderliche Wissen umfasst pro Ebene jeweils die folgenden drei Aspekte:

- Das ebenenspezifische Fach- oder Domainenwissen, das die typischen Aufgabenstellungen und Lösungen der Ebene in Form einer eigenen, abgegrenzten Begriffswelt erfasst.
- Das Modellierungs- und Darstellungswissen, das heißt, das Wissen um die Möglichkeiten der Findung von Systemmodellen, die in der ebenentypischen Begriffswelt verwurzelt sind und um die Möglichkeiten der Darstellung dieser Modelle für Fachleute und Nichtfachleute.
- Das andauernde Bewusstsein, die Grenzen der jeweiligen Betrachtungsebene einzuhalten bzw. auszuloten, um durch eine gezielte Trennung der unterschiedlichen Aspekte die Effektivität und Effizienz in der Entwicklung und die Bedienbarkeit im Betrieb interaktiver Systeme zu steigern.

Am einem Beispiel eines Wegeplanungssystems soll der unterschiedliche Charakter der drei Ebenen verdeutlicht werden. Abbildung 3.3 zeigt zu diesem Zweck drei unterschiedliche Modelle des Systems:

Der Zweck des Systems besteht darin, Benutzeranfragen, wie man von einem bestimmten Ausgangsort zu einem bestimmten Zielort kommt, entgegen zu nehmen und zu beantworten. Auf der Ebene des zweckbezogenen Modells besteht das betrachtete Interaktionssystem aus einem Benutzer und dem Wegeplanungssystem. Letzteres besteht aus einem Auskunftgeber mit einer großen Datenbank, in der Lage und Name bekannter Orte sowie mögliche Verbindungsrouten zwischen den einzelnen Orten hinterlegt sind. Über die Form der Interaktion, wie der Benutzer Ausgangsort und Zielort identifiziert bzw. wie der Auskunftgeber den oder die infrage kommenden Wege mitteilt, darf auf dieser Ebene keine Aussage getroffen werden. Die Benutzungsschnittstelle zwischen Benutzer und Auskunftgeber ist auf dieser Betrachtungsebene ein abstrakter Kanal, für den die Frage nach der konkreten Form des Interaktionsprozesses sinnlos ist.

Auf der Ebene interaktionsbezogener Modelle wird nun festgelegt, dass beide Orte in Form eines mehrschrittigen Dialoges identifiziert werden. Dabei kann ein Benutzer die Orte basierend auf den Vorschlägen eines Assistenten sukzessive einkreisen. Dieser Assistent ist als Komponente des Wegeplanungssystems anzusehen, die erst auf der Ebene interaktionsbezogener Modelle sichtbar wird. Die konkrete Ausprägung des Dialoges in Form gesprochener Worte wird ebenfalls auf dieser Ebene festgelegt.

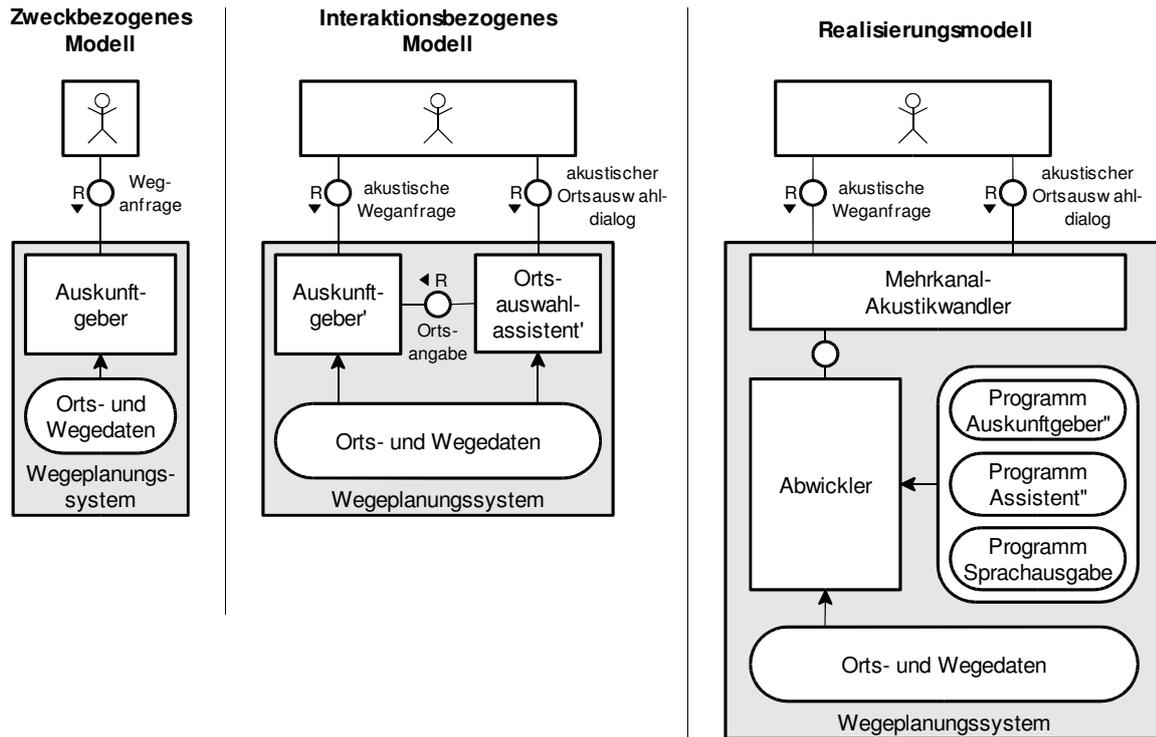


Abbildung 3.3: Betrachtungsebenen eines Wegeplanungssystems

Eine mögliche Realisierung dieses Modells durch Schall abgebende bzw. aufnehmende Schnittstellengeräte und einen informationellen Kern ist Gegenstand des Realisierungsmodells.

Wie Abbildung 3.2 zeigt, können die so entwickelten Modelle sowohl zur Schaffung interaktiver Systeme als auch zum Erwerb von Wissen für den Betrieb oder die Analyse von Interaktionssystemen genutzt werden. In beiden Fällen geht man angemessenerweise von der Ebene zweckbezogener Modelle aus. Bei der Entwicklung eines neuen Systems ist dies offensichtlich, da dies zu einem bestimmten Zweck geschieht. Beim Erwerb von Wissen über ein existentes System ist diese Reihenfolge keineswegs zwingend. Schließlich kann man die Bedienung eines interaktiven Systems erlernen, ohne die Konzepte des zugrunde liegenden Systems zu kennen. Das Wissen um das zweckbezogene Modell ermöglicht es aber, sinnvolle Fragen oder Vermutungen bezüglich der Bedienung und des zu erwartenden Systemverhaltens zu formulieren [Nor01, You83]. Nur wenn ein Mensch das erforderliche Wissen um die Nutzung des interaktiven Systems besitzt, kann er überhaupt in der Rolle eines Benutzer agieren. Die Vermittlung geeigneter Modelle an die potentiellen Benutzer eines interaktiven Systems sollte daher als Bestandteil der Inbetriebnahme des Interaktionssystems angesehen werden.

Das Ebenenmodell ist nicht als streng gerichtetes Prozessmodell zur Schaffung interaktiver Systeme zu verstehen. Eine gewisse Reihenfolge bei der Erstellung der Modelle wird sich von ganz allein ergeben, da die unterschiedlichen Modelle aufeinander aufbauen. Dennoch kann es sein, dass man beispielsweise aufgrund begrenzter Ressourcen

Realisierungsentscheidungen treffen muss, die rückwirkend zu einer Veränderung des Interaktionsmodells führen. Wenn sich die Realisierung der akustischen Schnittstelle des Wegeplanungssystems als unmöglich erweist, könnte man sich beispielsweise für eine textbasierte Benutzungsschnittstelle entscheiden. Unzulässig sind Implementierungsentscheidungen, die Rückwirkungen auf das zweckbezogene Modell besitzen. Sie würden damit den Sinn des interaktiven Systems in Frage stellen. Anhand der unterschiedlichen Zuständigkeiten lassen sich vier wesentliche Rollen identifizieren, in denen Personen agieren können, die während des Schaffungsprozesses interaktiver Systemen an der Modellbildung beteiligt sind:

- Kunden, zukünftige Anwender und Benutzer geben über ihre Bedürfnisse² sowohl den Zweck als auch das Repertoire möglicher Interaktionsformen des interaktiven Systems vor. Sie wirken somit in jedem Fall, direkt oder indirekt, an der Gestaltung der zweckbezogenen und interaktionsbezogenen Modelle mit.
- Die Rolle des Systemarchitekten umfasst alle Aktivitäten zur Festlegung der wesentlichen Strukturen der Systemmodelle, auf deren Grundlage alle weiteren Aktivitäten zur Schaffung des interaktiven Systems arbeitsteilig durchgeführt werden können. In [Kel03] wird die Menge all dieser Strukturen als die Architektur des Systems bezeichnet. Ein erfolgsentscheidender Faktor ist dabei die effektive und effiziente Kommunikation mit den am Schaffungsprozess beteiligten Personen. Eine weitere wichtige Aufgabe des Systemarchitekten besteht darin, die Begriffe und die Vorstellungswelten der unterschiedlichen Beteiligten auf gemeinsame Systemmodelle abzubilden und so die Integration der unterschiedlichen Erfahrungen zu ermöglichen.
- Die Rolle der Benutzungsschnittstellendesigner umfasst die Aktivitäten, die zur Gestaltung angemessener Benutzungsschnittstellen notwendig sind. Ausgehend von dem Wissen um die zweckbezogenen Modelle, die Fähigkeiten der zukünftigen Benutzer und die technischen oder finanziellen Randbedingungen wird die Schnittstelle entworfen. Im Rahmen dieser Aktivitäten wird das interaktionsbezogene Modell des Systems vollständig definiert. Da es vielfach zweckmäßig ist, die Festlegung der Interaktionsformen bereits im Hinblick auf ein verfügbares Repertoire an Realisierungskomponenten zu treffen, werden in diesem Schritt häufig bereits implizit Realisierungsentscheidungen getroffen.
- Die Rolle der Anwendungs- und Plattformentwickler umfasst die Aktivitäten zur Festlegung eines realisierungsvollständigen Modells. Mit dem Wissen um das vollständige interaktionsbezogene Modell und die Möglichkeiten einer Realisierung unter den gegebenen zeitlichen, finanziellen und technologischen Randbedingungen wird dieses Modell erstellt. Aufgabe der Anwendungsentwickler ist

²In der Regel gibt es zusätzliche Anforderungen, die bereits zukünftige Realisierungsentscheidungen vorwegnehmen, wie etwa die Wahl eines bestimmten Computersystems. Im Hinblick auf den unmittelbaren Zweck des Systems ist die Berücksichtigung dieser Anforderungen prinzipiell nicht erforderlich. Im Gesamtkontext der Kunden- bzw. Anwendersituation haben sie in der Regel durchaus ihre Berechtigung und sind daher natürlich zu berücksichtigen.

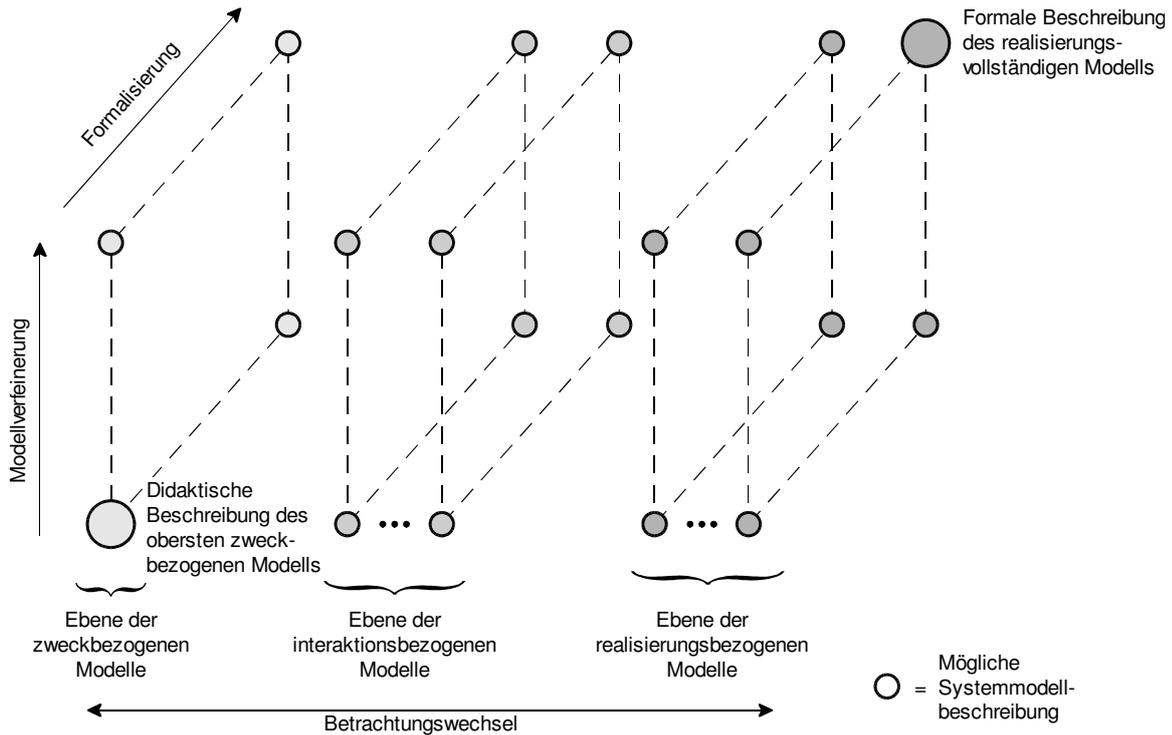


Abbildung 3.4: Ordnungskriterien für Systemmodellbeschreibungen

die Beschreibung des interaktiven Systems mittels formaler Sprachen im Hinblick auf seine Realisierung unter Bezug auf ein vorgegebenes Repertoire an Realisierungskomponenten. Aufgabe der Plattformentwickler ist die Bereitstellung eines solchen Repertoires für eine bestimmte Klasse von Systemen.

Eine ähnliche Trennung in Benutzer, Dialogdesigner und Anwendungsentwickler wird bereits in [BK82] vorgeschlagen. Die Notwendigkeit einer zentralen Vermittler- und Koordinatorrolle, hier in Gestalt des Systemarchitekten, wird erst in späteren Ansätzen erkannt (vgl. [KAB01]). Die Identifikation weiterer Rollen ist vor dem Hintergrund des hier betrachteten Modellfindungsprozesses nicht erforderlich, obwohl sie sicherlich für den Entwicklungsprozess in seiner Gesamtheit zweckmäßig ist.

Liegt ein realisierungsvollständiges Modell vor, ist die Fertigung des Systems möglich. Diese besteht bei einem System, das auf der Basis eines programmierbaren Trägersystems zu realisieren ist, aus der Bereitstellung des Trägersystems, die mit der Fertigung der Hardware verbunden ist, und dem Einbringen des Programms in den Speicher des Trägersystems. Das zur Fertigung der Hardware erforderliche Modell bildet zusammen mit dem Programm das realisierungsvollständige Modell.

Die im Laufe des Entwicklungsprozesses geschaffenen Modellbeschreibungen lassen sich nicht nur entsprechend ihrer Betrachtungsebenen einordnen. Sie unterscheiden sich — abgesehen vom dargestellten Systemausschnitt — hinsichtlich Auflösungsgrad und Zweck. Abbildung 3.4 veranschaulicht diesen Zusammenhang.

Wie in Kapitel 2.2.5 beschrieben, lassen sich Beschreibungen hierbei in solche mit

primär didaktischer Bedeutung und solche zur methodischen Ableitung von Konsequenzen unterscheiden. Letztlich muss mindestens das einzubringende Programm auf einer formalen Sprache basieren, da es durch das Trägersystem maschinell abgearbeitet wird. In der Hoffnung im Prozess nachgelagerte Modelle weitgehend maschinell abzuleiten, wird vielfach versucht, möglichst früh zu formalen Modellen zu gelangen. Eine Voraussetzung dafür ist, dass das Wissen, das zur Ableitung nachgelagerter Modelle erforderlich ist, wie beispielsweise ein Ergonomieleitfaden, ebenfalls formalisierbar ist. Üblicherweise ist dies nur für eng abgegrenzte Klassen interaktiver Systeme, wie etwa für Belegfassungssysteme[Sta00, Bal00], möglich.

Unabhängig davon sollte es daher auf jeder Ebene Modelle geben, die vorrangig der Förderung des Systemverständnisses und als Grundlage der Kommunikation zwischen den mitwirkenden Personen dienen. Für diese Modelle sind aufgrund der unterschiedlichen Hintergründe der am Prozess mitwirkenden Personen allgemein verständliche Darstellungen solchen vorzuziehen, die nur von einem kleinen Kreis eingeweihter Spezialisten verstanden werden können [MA01].

Jede der drei primären Modellebenen wird nachfolgend im Detail betrachtet. Es werden Konzepte vorgestellt, mittels derer auf jeder Ebene eine systematische Modellfindung möglich ist und die von der Vorstellung dynamischer Systeme geprägt sind. In Anlehnung an Abbildung 3.2 geht es darum, zu präzisieren, auf welchen Konzepten die Modellierung und Darstellung auf den unterschiedlichen Ebenen beruht und welche Rolle dabei die stete und bewusste Suche nach den Grenzen zwischen den verschiedenen Modellebenen spielt. Das Wissen um konkrete Anwendungsbereiche, ergonomische Gestaltungsprinzipien und konkrete Realisierungskonzepte steht nicht im Mittelpunkt, auch wenn es in den Modellen konkreter Systeme sichtbar wird. Es geht nicht um die formale Beschreibung von Modellen, sondern um die Fragen, die den Betroffenen beim Finden verständlicher Modelle leiten. Der in Abschnitt 2.3.4 vorgestellte Ansatz der Fundamental Modelling Concepts spielt dabei eine tragende Rolle.

3.3 Zweckbezogene Modelle

3.3.1 Zweck und Charakterisierung der Modelle

Auf der Ebene der zweckbezogenen Modelle wird ein Interaktionssystem im Hinblick auf genau die Eigenschaften und Strukturen des Systems beschrieben, die durch den Zweck des interaktiven Systems bedingt sind. Das Interaktionssystem wird dabei durch die Struktur aus dem interaktiven System und genau den Komponenten in seiner Umgebung gebildet, anhand derer sein Zweck verständlich wird. Es kann daher im Gegensatz zu einem isolierten interaktiven System abgeschlossen betrachtet und verstanden werden. Jeglicher Bezug auf Formen der Benutzung und Erscheinung oder auf interne Strukturen des interaktiven Systems, die als Ergebnis von Realisierungsentscheidungen anzusehen sind, ist auf der Ebene zweckbezogener Modelle unzulässig. Für die Modellfindung erforderlich ist das Wissen um den Anwendungsbereich, die Konzepte der Modellierung und deren Darstellung, sowie das Bewusstsein für die Trennung von Zweck und Form.

Der Zweck dieser Modelle besteht darin, die Aufgaben des interaktiven Systems verbindlich zu beschreiben. Dabei werden explizit all die Formen der Interaktion zwischen Benutzern und System, die prinzipiell kommunikativ lösbar sind, offen gehalten. Durch die Ausblendung der Form- und Realisierungsaspekte sind zweckbezogene Modelle vergleichsweise überschaubar und verständlich. Didaktische Beschreibungen zweckbezogener Modelle sind daher sowohl als Kommunikationsmittel während der Anforderungsfindungsphase von interaktiven Systemen als auch als Einstiegshilfe für potentielle Benutzer geeignet. Streng formale Beschreibungen zweckbezogener Modelle erlauben die Anwendung analytischer Methoden. Diese können beispielsweise zur frühzeitigen Aufdeckung verborgener konzeptioneller Schwachstellen oder Fehler genutzt werden. Durch die Abgrenzung von zweckbezogenen Modellen und interaktionsbezogenen Modellen wird es vergleichsweise einfach, unterschiedliche Interaktionsformen gegeneinander abwägen oder alternativ zu realisieren. Dies begünstigt die Entwicklung angemessener Benutzungsschnittstellen. In vielen Fällen wird die abstrakte Betrachtungsebene zweckbezogener Modelle einen Schritt zurück erfordern. Dadurch werden gewohnte Formen von Benutzungsschnittstellen zunächst bewusst in Frage gestellt.

3.3.2 Allgemeine Ansätze zur Modellfindung

Im allgemeinen ist es bei der Beschreibung komplexerer interaktiver Systeme zweckmäßig, ausgehend von einem Übersichtsmodell schrittweise Verfeinerungen einzuführen, so dass sich eine Halbordnung zweckbezogener Modelle unterschiedlicher Auflösung ergibt. Dies beantwortet jedoch nicht, welche Fragen einen bei der Findung von Systemmodellen leiten und was durch die Modelle festgelegt wird.

Bei der Systemmodellfindung gibt es grundsätzlich zwei verschiedene Ausgangssituationen. Bei der Entwicklung eines interaktiven Systems, der *Systemsynthese*, sind zumeist die Umgebung des zukünftigen Systems, sowie eine Menge von Anforderungen mehr oder weniger präzise vorgegeben. Letztere lassen sich bereits klassifizieren in zweckbezogene, interaktionsbezogene und realisierungsbezogene Anforderungen. Eine *Systemanalyse* setzt ein bestehendes Interaktionssystem voraus. Das Ziel ist es, Strukturen innerhalb des Systems zu erkennen, wobei häufig auch erst die Grenze des Systems bestimmt werden muss.

In beiden Fällen ist der nachstehend formulierte Kanon leitender Fragen hilfreich. Ihre Beantwortung, die auf dem vorgegeben oder zu ermittelnden Domainwissen basiert³, führt im Rahmen der Modellsynthese zur Findung von Systemmodellen.

- Was ist der Zweck des interaktiven Systems? — Vor dem Hintergrund einer gesuchten Systemvorstellung kann ein solcher Zweck darin bestehen, einen gewünschten Systemzustand herzustellen, der mit dem Belegen von bestimmten Systemorten mit bestimmten Werten gleichzusetzen ist. Man spricht in einem

³Eine Vielzahl unterschiedlicher Techniken zur Gewinnung des erforderlichen Wissens ist bekannt. Im Kern geht es darum, die bestehenden Erfahrungen mit dem Anwendungsbereich vertrauter Personen in den Prozess der Modellbildung einfließen zu lassen. Eine Übersicht bekannter Techniken findet sich in [KA92].

solchen Fall von einem ergebnisorientierten System. Andernfalls handelt es sich um ein prozessorientiertes System, wenn der darin Zweck besteht, bestimmten Orten im System einen Werteverlauf aufzuprägen.

Bei Systemen mit informationellem Systemzweck stellen diese Werte Informationen dar. Bei Systemen mit materiell–energetischem Systemzweck sind nicht interpretierte, physikalische Größen bzw. Gebilde zu beobachten. Da sich die Ausklammerung des Formaspektes auf der Ebene zweckbezogener Modelle nur auf die Art und Weise bezieht, in der Benutzer mit dem System kommunizieren, stellt die Modellierung von Systemen mit materiell–energetischem Zweck nicht notwendigerweise einen Vorgriff auf die Ebene interaktionsbezogener Modelle dar.

- Welche Werte sind innerhalb des System beobachtbar und welche Operationen müssen darauf in welcher Reihenfolge ausgeführt werden, damit sich ein bestimmungsgemäßes Systemverhalten ergibt? — Als Werte sind alle innerhalb des Systems vorkommenden Informationen bzw. uninterpretierten, physikalischen, passiven Gebilde und Größen anzusehen, die für ein Verständnis des Systemzweckes erforderlich sind. Die Beobachtbarkeit eines jeden Wertes ist eindeutig an einen bestimmten Ort und eine bestimmte Zeitspanne gebunden. Strukturierte Werte besitzen im Hinblick auf den Systemzweck eine relevante innere Struktur.

Alle Operationen sind in eine Kausalstruktur eingebunden, aus der das beobachtbare Verhalten des System ableitbar ist. Dadurch ist festgelegt, in welcher zeitlichen Abfolge welche Schreib– bzw. Leseoperationen zu erfolgen haben, welche Operationen voneinander abhängig sind und welche voneinander unabhängig erfolgen können.

- Welche Zuständigkeitsbereiche aktiver Systemkomponenten können identifiziert werden und definieren so den zweckbezogenen Aufbau des Interaktionssystems? — Vor dem Hintergrund der gesuchten Systemvorstellung lässt sich jede zweckbezogene Operation einer aktiven Komponente, einem Akteur, zuordnen. Eine solche Komponente muss dazu auf die Systemorte, das heißt Speicher und Kanäle, zugreifen können, von denen zur Durchführung der Operation gelesen oder auf die geschrieben werden muss. Die so identifizierte Aufbaustruktur bildet dann ein angemessenes zweckbezogenes Aufbaumodell des Interaktionssystems, wenn das Wesen aller Systemkomponenten ausschließlich durch den Systemzweck definiert ist und wenn jede Komponente durch eine in ihrer Bedeutung geschlossen verständliche Zuständigkeit besitzt.

Im allgemeinen ist jeder Akteur für die Durchführung von Operationen unterschiedlichen Typs zuständig. Als Merkmal einer verständlichen Abgrenzung einer zweckbezogenen Komponente gilt, wenn diese mit einem einzigen im Anwendungsbereich verwurzelten Begriff benannt werden kann. Dieser Begriff muss das Wesen der Komponente treffend charakterisieren. Per Definition sind die Benutzer des interaktiven Systems identifizierbare Aufbaukomponenten von Interaktionssystemen. Das interaktive System selbst wird durch die übrigen Komponenten des Systems gebildet, mit denen die Benutzer zum Erreichen des definierten Zwecks interagieren.

Der Wert einer Aufbaustruktur besteht darin, dass es möglich ist, ein Verständnis vom Zusammenspiel der Systemfunktionen zu erhalten, ohne im Detail jede einzelne Operation oder jeden Wert kennen zu müssen. Geleitet von einem intuitiven Verständnis der Struktur ist es möglich, Fragen hinsichtlich des Systemverhaltens und nach Zuständigkeiten zu formulieren. Eine mögliche Verhaltensstruktur wird sichtbar. Dabei spielt es keine Rolle, ob die Menge identifizierter Operationen und Werte zunächst vollständig ist oder nicht. Das so gewonnene Systemverständnis erlaubt es Schwachstellen zu identifizieren und schrittweise das Modell und damit das System zu vervollkommen. Die Abgrenzung zweckbezogener Aufbaukomponenten dient dazu, durch Abstraktion von jeglicher Realisierungsform und von funktionalen Details eine Grundlage für die Vorstellung zu gewinnen, wie durch ein Zusammenspiel dieser Komponenten das bezweckte Verhalten des Systems erreicht werden kann.

Von besonderer Bedeutung für die Gestaltung der Benutzungsschnittstelle ist der Verlauf zulässiger informationeller Werte, der sich aus der Abfolge zweckbezogener Ein- und Ausgabeoperationen von interaktivem System und Benutzern ergibt. Im allgemeinen ist es zweckmäßig, Kanäle für Ein- und Ausgaben zu unterscheiden. Bidirektionale Kanäle sollten als Ergebnis einer Realisierungsentscheidung angesehen werden.

Unzweckmäßig ist zumeist die Suche nach einer Isomorphie zwischen einem zweckbezogenen Aufbaumodell und der Struktur gegenständlich erlebbarer Objekte im Anwendungsbereich. So ist ein Teil der eigenständig erlebbaren Objekte formgebender Bestandteil der Benutzungsschnittstelle und daher auf der Ebene zweckbezogener Modelle gewollt unsichtbar. Wird ein gegenständliches Objekt im Hinblick auf den Systemzweck im Rollenmultiplex genutzt, realisiert das Objekt auf der Ebene zweckbezogener Modelle unterschiedliche Komponenten (vgl. Kapitel 2.2.5.2).

Am Beispiel eines Autorenunterstützungsystems soll der Ansatz verdeutlicht werden. Zunächst ist die Frage nach dem Zweck des interaktiven System zu beantworten. Dabei ist nur abstrakt das Gewollte zu identifizieren, nicht das Vorgehen, das zum Erreichen des Gewollten erforderlich ist, nicht das Repertoire an Funktionalität des Systems. Im allgemeinen ist es möglich, auch den Zweck komplexer Systeme in ein oder zwei Sätzen zu beschreiben. Im Beispiel ist das Gewollte ein interaktives System, das einen Autor bei seiner Arbeit, dem Verfassen von Büchern und Artikeln, also von Dokumenten, unterstützt. Jegliche Operation des Autorenunterstützungsystems wird durch eine Eingabe vom Autor angestoßen. Das Ergebnis sind die fertigen Dokumente bzw. letztendlich die gedruckten Exemplare. Es handelt sich daher um ein eingabegetriebenes, ergebnisorientiertes System.

Zur Beantwortung der zweiten Frage benötigt man Informationen über die Arbeitsweise eines Autors. Geht der Verfasser dieses Textes von sich selbst aus, so lassen sich als zweckbezogene Werte das in Bearbeitung befindliche Dokument — gegebenenfalls darauf basierende Druckexemplare — sowie ein Dokumentenarchiv identifizieren. Im Überblick lassen sich vier Aufgabenfelder identifizieren: Das Erstellen, Bearbeiten, Verwalten und Drucken von Dokumenten.

Durch Analyse der Anforderungen lassen sich konkrete Teilaufgaben mit abgeschlossener Bedeutung abgrenzen, aus denen sich das Repertoire zweckbezogener Operationen ergibt. Im Beispiel könnte dies im Hinblick auf die Erstellung und Bearbeitung

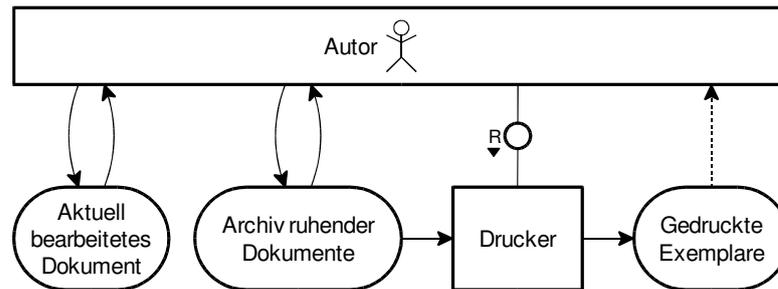


Abbildung 3.5: Autorensystem — Abstrakte Sicht

folgende Operationstypen sein: Identifikation eines neuen oder bestehenden Dokumentes zur Bearbeitung, Identifikation der Bearbeitungsposition im Dokument, Hinzufügen oder Löschen eines Wortes an der Bearbeitungsposition.

Die prinzipiell erforderliche Festlegung konkreter Abläufe, sowie die innere Struktur des Archivs bzw. der Dokumente soll an dieser Stelle ausgespart werden.

Bei der Beantwortung der dritte Frage lässt sich in jedem Falle der Autor als aktive Systemkomponente abgrenzen. Bei einer abstrakten Betrachtung, wie in Abbildung 3.5 gezeigt, besitzt der Autor direkten Zugriff auf das Dokumentenarchiv, das Dokument, das aktuell bearbeitet wird, sowie gegebenenfalls auf die gedruckten Exemplare. Diese werden durch einen Drucker erstellt, mit dem der Autor kommuniziert. Die Verwendung eines interaktiven Systems, das den Autor bei seiner Arbeit unterstützt, geht aus diesem Modell nicht hervor. Im Bezug auf den Zweck des gesuchten Systems ist die Sichtweise deshalb zu abstrakt gewählt.

Bei einer typischen Aufgabenteilung zwischen interaktivem System und Benutzer nimmt der Benutzer die Rolle eines Auftraggebers, des willensäußernden Akteurs, ein. Das interaktive System, der Auftragnehmer, ist der willensumsetzende Akteur. Durch seine Ausgaben vermittelt es dem Benutzer eine Vorstellung von den durchgeführten Aktivitäten. Im Beispiel tritt der Autor gegenüber einem Autorenunterstützungssystem als Auftraggeber auf, das aufgrund der Willensäußerungen des Auftraggebers auf Archiv und Dokument Operationen ausführt. Die entstehende Aufbaustruktur ist in Abbildung 3.6 gezeigt. Der erforderliche Kanal zwischen Autor und Autorenunterstützungssystem wird durch die Benutzungsschnittstelle gebildet, über die auf dieser Betrachtungsebene nur informationelle Werte ausgetauscht werden. Das Repertoire beobachtbarer Werte umfasst die Menge denkbarer Verwaltungs- und Bearbeitungsaufträge des Autors und die möglichen Rückmeldungen des Autorensystems. Trotz der nun indirekten Bearbeitung der Dokumente muss es einem Autor jederzeit möglich sein, den aktuellen Stand eines in Bearbeitung befindlichen Dokumentes zu erfahren. Aus abstrakter Sicht ist das Dokument hierdurch an der Benutzungsschnittstelle sichtbar. Wie dies realisiert ist, ist nicht Gegenstand des zweckbezogenen Modells.

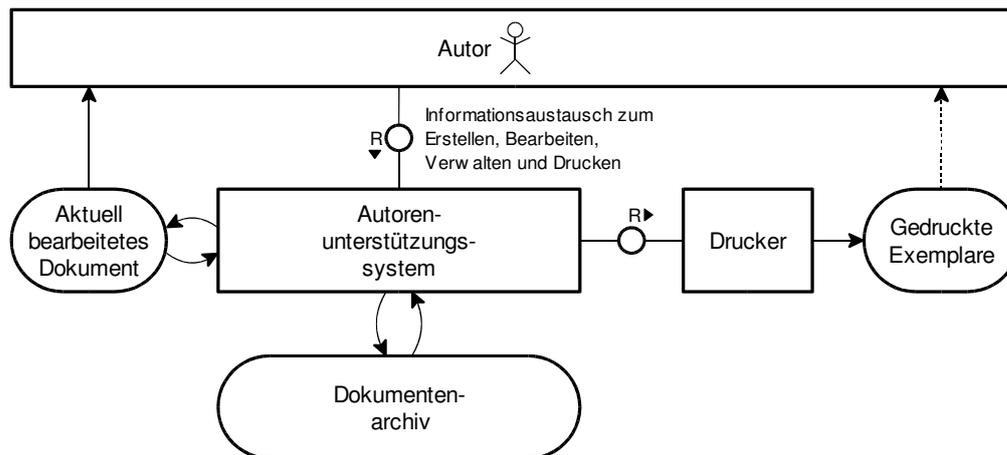


Abbildung 3.6: Autorensystem — Grobes zweckbezogenes Aufbaumodell

3.3.3 Abgrenzung des informationellen Systems

Bei der Modellierung von Interaktionssystemen mit materiell-energetischem Zweck ist es sinnvoll, zwischen informationell zu betrachtenden Komponenten und kontinuierlich zu betrachtenden Komponenten zu unterscheiden. Informationell zu betrachtende Komponenten dienen der Speicherung, der Bearbeitung oder dem Transport interpretierbarer Formen, deren Relevanz durch die darin codierte Information begründet ist. Das Zusammenspiel solcher Komponenten kann durch Ereignisfolgengeflechte oder — angemessenerweise — durch generative Schemata zur Ableitung solcher Geflechte beschrieben werden. Eingangs- und Ausgangsgrößen sind eindeutig definiert. Kontinuierlich zu betrachtende Komponenten dienen der Speicherung, der Beeinflussung oder dem Transport von Gebilden, deren Relevanz im Kontext des betrachteten Systems in ihren messbaren Eigenschaften aus dem materiell-energetischen Kontinuum liegt. Die Beschreibung des Zusammenwirkens solcher Komponenten basiert im allgemeinen auf anderen Ansätzen. Als Beispiel mögen die etwa die Formeln der Differentialrechnung gelten.

Die prinzipielle Einbindung eines interaktiven Systems mit materiell-energetischem Zweck, wie sie beispielsweise der Leitstand eines Kraftwerkes darstellt, ist in Abbildung 3.7 gezeigt. Die Benutzungsschnittstelle dient Benutzern zur Äußerung ihres Willens gegenüber dem interaktiven System und dem Erhalt gewünschter Informationen. Ihr Zweck ist daher informationeller Natur. Benutzer und interaktives System bilden ein informationelles System. Die Kopplung zwischen diskret und kontinuierlich arbeitenden Systemkomponenten, erfolgt mithilfe von Wandlern. Durch diese erfolgt eine Abbildung zwischen den relevanten kontinuierlichen Größen und Informationen über diese Größen im interaktiven System. Charakteristische Merkmale dieser Wandler sind

- die zeitliche Auflösung und
- die Wertebereichsauflösung

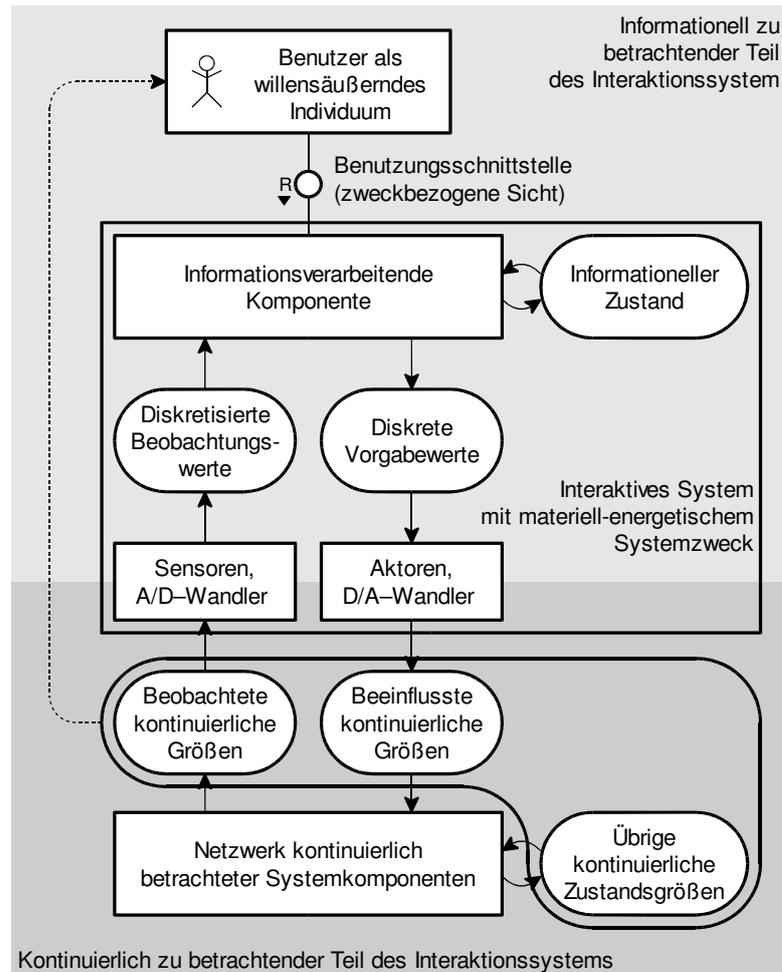


Abbildung 3.7: Interaktives System mit materiell-energetischem Systemzweck

mit der physikalische Größen erfasst bzw. vorgegeben werden können. Unter Berücksichtigung zeitlicher Vorgaben, die durch den Systemzweck bedingt sind, kann so eine gewünschte Beeinflussung kontinuierlicher Größen durch diskret arbeitende Komponenten erfolgen.

Durch die Abgrenzung des informationellen Systems kann sich die Modellbildung für die folgenden Betrachtungen auf den informationell betrachtbaren Teil von Interaktionssystemen beschränken.

3.3.4 Konzepte der Modellverfeinerung

Für das Verständnis komplexer Systeme ist es zweckmäßig, von einem groben Übersichtsmodell auszugehen und Details erst im Rahmen von Verfeinerungen sichtbar werden zu lassen. Die Vorstellung von einem monolithischem interaktivem System, das einem Benutzer als einziger Interaktionspartner das gesamte Repertoire möglicher Operationen bereitstellt, ist nur bei Systemen geringer funktionaler Komplexität sinnvoll. Zweckmäßiger ist eine Vorstellung von mehreren Interaktionspartnern

mit unterschiedlichen Zuständigkeiten, die jeweils nur eine Teilmenge semantisch verwandter Operationen ermöglichen. Eine Menge von Operationen zur Manipulation eines bestimmten Typs von Informationen auf einem gemeinsamen Ort könnte einen solchen Zusammenhang darstellen.

Betrachtet man das Autorensystem, so kann ein Autor über die Benutzungsschnittstelle zum einen Operationen anstoßen, die eine Veränderung des in Bearbeitung befindlichen Dokumentes zur Folge haben und zum anderen solche, die mit dem Management bestehender Dokumente zu tun haben. Eine angemessene Vorstellung ist daher, dass der Autor mit zwei unterschiedlichen Kommunikationspartnern im Wechsel interagiert. Es ergibt sich die in Abbildung 3.8 gezeigte Aufbaustruktur, in der das Autorenunterstützungssystem aus einem Dokumentenbearbeitungsakteur und dem Dokumentenmanager besteht.

Die Aufteilung von Zuständigkeiten auf unterschiedliche Akteure für verschiedene Operationen und die damit verbundene Benennung der Akteure basiert auf einer Abstraktion über den zusammengefassten Operationstypen, die das Wesen der unterschiedlichen Akteure definiert. Das Wissen um Aufbaustruktur und Wesen der unterschiedlichen Aufbaukomponenten erlaubt es, Erwartungen hinsichtlich des operationellen Repertoires zu formulieren. Während der Entwurfsphase eines System sollten diese Erwartungen genutzt werden, das geplante Repertoire auf Vollständigkeit zu überprüfen. Während der Betriebsphase erlauben diese Erwartungen einem unerfahrenen Benutzer Fragen hinsichtlich der Funktionalität und Bedienung zu formulieren. Die zweckbezogene Verfeinerung der Aufbaustruktur ermöglicht somit die Reduktion des zur Beherrschung des Systems erforderlichen Präsenzwissens.

Ist die Auflösung des Modells zu hoch gewählt, resultiert dies in einer unübersichtlichen Anzahl von Kommunikationspartnern. In der Folge leidet das Verständnis. Als Richtgröße für eine beherrschbare Menge unterscheidbarer Objekte wird häufig in Anlehnung an [Mil56] eine Mächtigkeit von etwa „7, plus oder minus 2“ angegeben. Allerdings sind die zugrundeliegenden Studien aufgrund ihres Inhaltes (z.B. das Erkennen unterschiedlicher Töne) wenig geeignet, um verbindliche Rückschlüsse auf eine vertretbare Anzahl unterscheidbarer Kommunikationspartner zu treffen. Dennoch wird es bei komplexeren Systemen zweckmäßig sein, nacheinander unterschiedliche Komponenten zu verfeinern. So könnte es im Beispiel des Autorensystems zweckmäßig sein, eine Aufbaustruktur zu betrachten, in der der Dokumentenbearbeitungsakteur weiter verfeinert wäre. Die Archivverwaltung hingegen könnte in unveränderter Auflösung betrachtet werden.

Theoretisch lässt sich ein Verband zweckbezogener Modelle finden. Dieser ist durch das gröbste und das feinste zweckbezogene Modell beschränkt. Auf ihm sind die beiden Operationen der Verfeinerung und der Vergrößerung definiert. Praktisch wird eine Auswahl von Modellen stattfinden, wobei man sich an didaktischen und an formalen Kriterien, wie Vollständigkeit und Korrektheit, orientieren wird.

3.3.5 Abgrenzung von der Ebene interaktionsbezogener Modelle

Zu Beginn dieses Kapitels wurde als Merkmal zweckbezogener Modelle definiert, dass jeglicher Bezug auf die Form der Benutzung und Erscheinung des interaktiven Sys-

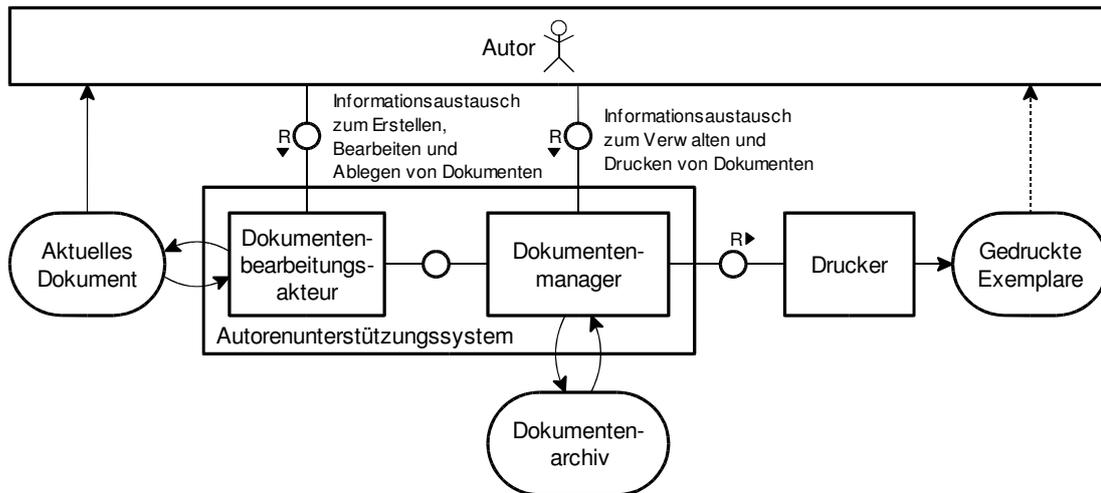


Abbildung 3.8: Autorensystem — Verfeinertes zweckbezogenes Aufbaumodell

tems und dessen interne tatsächliche Realisierung unzulässig ist. Was als zweckbezogen und was interaktionsbezogen anzusehen ist, wird in jedem Fall nur im Bezug auf einen zuvor festgelegten Zweck des Interaktionssystems zu beantworten sein. Wenn der Zweck eines Systems ganz allgemein darin besteht, Texte zu erfassen, wäre es ein Vorgriff auf die interaktionsbezogenen Ebene, wenn festgelegt würde, dass die Beschreibung der Erfassung handschriftlicher Texte mithilfe eines Lesegerätes und einer nachgeschalteten Texterkennungskomponente erfolgen solle. Wenn die Erfassung handschriftlicher Texte jedoch der Zweck des gewünschten interaktiven System ist, muss genau dies durch ein zweckbezogenes Modell erfasst sein. Über den Zweck ist definiert, welche Freiheitsgrade bei der Gestaltung von Form und Verhalten an der Benutzungsschnittstelle vorhanden sind. Die präzise Festlegung des Systemzwecks sollte daher als Ergebnis eines sorgfältigen Entscheidungsprozesses angesehen werden. Die zukünftigen Nutzer sollten in jedem Falle an diesem Entscheidungsprozess beteiligt sein.

Die Festlegung konkreter Darstellungsformen von Informationen gehört zweifelsfrei in die Ebene zweckbezogener Modelle. Daneben gibt es jedoch auch weniger offensichtliche Vorgriffe auf die Ebene interaktionsbezogener Modelle, wie die nachfolgenden Anmerkungen zeigen:

- Für zulässigerweise nebenläufige Operationen darf in zweckbezogenen Modellen keine Ausführungsreihenfolge festgelegt werden.
- Ist durch den Zweck eine Reihenfolge von Interaktionen zwischen Benutzer und interaktivem System vorgegeben, bedingt dies nicht in jedem Fall eine Steuerung des Interaktionsprozesses durch das interaktive System. Abhängig vom Zweck des Systems wird in manchen Fällen der Interaktionsprozess durch den Benutzer vorangetrieben, in anderen Fällen vom interaktiven System.
- Die Identifikation zweckbezogener Werte ist in zweckbezogenen Modellen als einschrittig anzusehen. Eine Identifikation zweckbezogener Werte in Form eines

mehrschrittigen Interaktionsprozesse zwischen Benutzer und interaktivem System muss als Ergebnis einer vorweggenommenen interaktionsbezogenen Entwurfsentscheidung angesehen werden. Hierzu folgendes Beispiel: Ist ein durch den Zweck definiertes Dokument, das in einem Archiv abgelegt ist, zum Abruf über seinen Namen zu identifizieren, so ist von den Interaktionen eines Benutzers mit einer Auswahldialogkomponente zur Bestimmung des Namens auf Ebene des zweckbezogenen Modells zu abstrahieren.

- Im allgemeinen wird beim Entwurf von Benutzungsschnittstellen angestrebt, Zustandskomponenten des interaktiven Systems, die im Rahmen mehrschrittiger Interaktionsprozesse hilfreich sein könnten, unmittelbar beobachtbar zu machen. Im Falle ergebnisorientierter Systeme und Steuerungen ist dies jedoch Konsequenz ergonomischer Überlegung und nicht zweckbezogene Notwendigkeit. Lediglich im Falle benutzergeführter Regelungen muss der Benutzer ständig über den Zustand der Regelgrößen informiert sein. Im Falle des vorgestellten Autorensystems auf Ebene des zweckbezogenen Modells besteht ein Kanal zwischen Benutzer und interaktivem System zur Dokumentenbearbeitung. Eine Notwendigkeit, den aktuellen Dialogzustand zu beobachten, von dem es abhängt, ob Text eingefügt oder ersetzt werden soll, gibt es jedoch nicht.

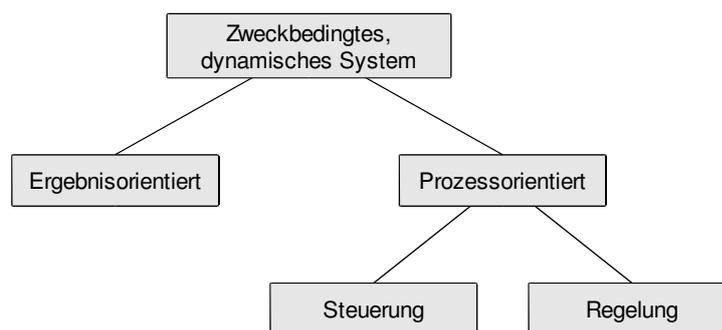


Abbildung 3.9: Klassifikation dynamischer System nach ihrem Zweck

3.4 Interaktionsbezogene Modelle

3.4.1 Zweck und Charakterisierung der Modelle

Interaktionsbezogene Modelle zeigen im Unterschied zu zweckbezogenen Modellen, *wie* das bezweckte Verhalten des interaktiven Systems durch die Verwendung konkreter informationstragender Formen an der Benutzungsschnittstelle erreicht wird. Um dabei den Bezug zwischen Form und Zweck zu erhalten, erweist sich eine Vorstellung von formgebenden bzw. –erfassenden Interaktionsadaptern als zweckmäßig, die zwischen einem Benutzer und dem zweckbasierten Kern des interaktiven Systems vermitteln. Dieser Kern spiegelt die Struktur des entsprechenden zweckbezogenen Modells wider. Die an der Benutzungsschnittstelle beobachtbaren Formen dienen im Verlauf

der Kommunikation zwischen Benutzer und interaktivem System der Identifikation zweckbezogener Werte und Kanäle.

Unter Berücksichtigung der Verträglichkeit mit zweckbasierten Vorgaben sind Abbildungen zweckbezogener Werte und Operationsfolgen auf beliebige Interaktionsfolgen und Formen zulässig. Die Abbildung zweckbezogener Werte muss nicht notwendiger Weise „Eins zu Eins“ auf einzelne Schnittstellenformen erfolgen. Beispielsweise können Kommunikationsprozesse, die bezüglich ihres Zwecks als einschrittig anzusehen sind, durch mehrschrittige Interaktionsfolgen, deren Form besser den menschlichen Fähigkeiten angepasst ist, realisiert sein.

Idealerweise sollte die Kommunikation zwischen einem interaktivem System und seinen Benutzern auf Interaktionsformen basieren, die ein effektives und effizientes Erzielen des Systemzwecks ermöglichen und zur Zufriedenheit der Benutzer beitragen. Bei der Schaffung solcher Systeme sind sowohl das Wissen um die Konzepte und Möglichkeiten der Gestaltung von Benutzungsschnittstellen als auch die Fähigkeiten der zukünftigen Benutzer zu beachten. Zusätzlich muss ein Bewusstsein dafür existieren, dass die Festlegung bedienrelevanter Interaktionsformen nicht mit Entscheidungen über eine interne Realisierung des interaktiven Systems vermischt werden darf. Die explizite Erstellung geeigneter, interaktionsbezogener Modelle ist von großer Wichtigkeit, wobei die Modelle unterschiedlichen Zwecken dienen können:

- Didaktische Modelle als Grundlage der Kommunikation und von Entscheidungen im Entwurfsprozess. Als Grundlage von Entscheidungen dienen sie unter anderem dazu, ihre Abstimmung auf vorgegebene Restriktionen, wie verfügbare Entwicklungsressourcen oder Benutzeranforderungen zu bewerten.
- Didaktische Modelle als Grundlage von Bedienungsanleitungen
- Formale Modelle als Grundlage automatisierter Realisierungen
- Formale Modelle als Grundlage von Analysen zur Ableitung von Aussagen über das erwartete Interaktionsverhalten [JK96, CMN80]

Viele der einschlägig bekannten Ansätze konzentrieren sich auf die Darstellung der Werteverläufe, die an der Benutzungsschnittstelle beobachtbar sind. Dazu werden unter anderem Zustandsdiagramme, Grammatiken, Bildschirmfotos oder Prototypen genutzt. Andere Ansätze basieren auf der Vorstellung eines aus Komponenten aufgebauten interaktiven Systems. Leider entpuppt sich der verwendete Komponentenbegriff dabei häufig als unklar oder als mit einem konkreten Software-technischen Realisierungsansatz verbunden.

Der hier vorgestellte Ansatz zur Findung interaktionsbezogener Modelle ist ebenfalls mit der Vorstellung eines dynamischen Systems verbunden, welches durch Aufbau-, Verhaltens- und Wertestrukturen gekennzeichnet ist. Die Findung interaktionsbezogener Modelle erfolgt auf Basis eines zweckbezogenen Modells durch:

- die Festlegung der Formen bzw. Formverläufe, die an der Benutzungsschnittstelle beobachtbar bzw. beeinflussbar sind. Sie dienen der Identifikation von Informationen und Kanälen, die durch das zweckbezogene Modell definiert sind,

und werden als wahrnehmbare Eigenschaften von aktiven Systemkomponenten, sogenannten *Formadaptern* und *Schrittfolgenadaptern*, angesehen.

- die Integration zusätzlicher interaktionsunterstützender Systemkomponenten in Gestalt sekundärer Kanäle und *Interaktionsassistenten*, die nicht als Realisierung zweckbezogener Kanäle, sondern als Ergebnis von Bedienbarkeitsüberlegungen anzusehen sind.

Interaktionsbezogene und zweckbezogene Modelle sind im Rahmen dieser Arbeit im Bezug auf die interne Realisierung eines interaktiven Systems als potentielle Ersatzmodelle anzusehen. Die tatsächliche innere Struktur eines interaktiven Systems, die interne Realisierung, wird durch ein interaktionsbezogenes Modell nicht festgelegt. Diese Sichtweise unterscheidet sich von der scheinbaren Durchgängigkeit vieler heutiger Ansätze.

Aus Sicht eines Schnittstellendesigners gibt es damit prinzipiell keinerlei Beschränkungen im Repertoire infragekommender Interaktionsformen durch eine unnötig vorweggenommene Wahl einer Realisierungsplattform. Kunden und zukünftige Anwender werden nicht mit einer für sie unbekanntem Begriffswelt konfrontiert und Anwendungs- und Plattformentwickler sind frei in der Wahl einer Realisierungsstruktur.

Abwägungen zwischen verfügbaren Ressourcen, zum Beispiel in Form vorgegebener Plattformen, und dem Streben nach optimalen Benutzungsschnittstellen setzen die Kenntnis denkbarer Alternativen voraus. Auch wenn Monitor, Maus und Tastatur für viele Zwecke geeignet sind, erlauben Sprachsteuerung oder konventionelle Drehknöpfe bei manchen Systemen eine leichtere Bedienung. Formadapter, Schrittfolgenadapter und Interaktionsassistenten sind Abstraktionen, die für die Vielzahl unterschiedlicher Interaktionskonzepte eine gemeinsame Basis können.

Die unmittelbare Darstellung der Benutzungsschnittstelle, die auf der Ebene interaktionsbezogener Modelle beispielsweise durch Bildschirmskizzen möglich ist, verliert durch diese Sichtweise keineswegs ihre wertvolle Bedeutung. Vielmehr erlaubt der geschilderte Ansatz eine Verknüpfung unmittelbar darstellbarer Systemaspekte der Ablauf- und Wertestrukturen mit didaktisch begründeten, anwendungsspezifischen Aufbaumodellen zu einer vollständigen Systemvorstellung. Die Herleitung einer solchen Vorstellung steht im Mittelpunkt der folgenden Überlegungen.

3.4.2 Merkmale von Anwendern und Benutzern

Die Formgestaltung einer Benutzungsschnittstelle, die eine effektive und effiziente Nutzung des interaktiven Systems erlaubt, setzt Wissen um die Fähigkeiten der unterschiedlichen Benutzer voraus. Diese Fähigkeiten lassen sich klassifizieren in:

- Physische Fähigkeiten
- Kognitive Fähigkeiten
- Fach- und Bedienwissen

Die physischen Fähigkeiten bestimmen den Wertebereich physikalischer Größen, die als anwendbare Informationsträger an der Benutzungsschnittstelle in Frage kommen. Der Wertebereich individuell möglicher Sinneswahrnehmungen von Augen, Ohren, Tast-, Riech- und Geschmackssinn grenzt den Bereich individuell angemessener Ausgabeformen des interaktiven Systems ab. Die motorischen und stimmlichen Fähigkeiten grenzen den Bereich individuell angemessener Eingabeformen ab. Darüber hinaus ist eine Erweiterung der natürlich gegebenen Fähigkeiten — insbesondere bei Benutzern mit körperlichen Behinderungen — durch eine direkte Kopplung technischer Sensoren und Aktoren mit dem menschlichen Nervensystem in eingeschränktem Maße möglich [Dob00]. Mit diesem Wissen lässt sich beispielsweise abschätzen, wie Größen, Farben und Kontraste für Symbole oder die Griffform und das mechanische Verhalten von Zeigergeräten angemessenerweise zu wählen sind.

Die kognitiven Fähigkeiten eines Benutzers bestimmen, in welchem Maße und in welcher Zeit ein Benutzer Symbole erkennen, Sachverhalte identifizieren, sich einprägen und sich später an sie erinnern kann. Darüber hinaus zeigen sie, in welchem Ausmaß er zu Abstraktionen, Assoziationen und Ableitung von Konsequenzen fähig ist. Diese Fähigkeiten können wesentlich durch die Umgebung beeinflusst werden, in der das interaktive System genutzt werden soll. Zum einen ergeben sich daraus Richtzeiten für die unterschiedlichen Typen von Benutzeroperationen, die bei der Gestaltung des Interaktionsprozesses mit den zweckbezogenen zeitlichen Anforderungen in Einklang zu bringen sind. Zum anderen ist von diesen Fähigkeiten abhängig, ob zusätzliche Systemkomponenten erforderlich sind, die beispielsweise das Gedächtnis des Benutzers entlasten. Damit verbunden ist die Festlegung von Anzahl und Form zeitgleich erlebbarer Symbole bzw. Interaktionsobjekte.

Das Fachwissen umfasst jenen Teil des erforderlichen Benutzerwissens, der durch den Zweck des System bedingt ist. Durch die Tiefe dieses Fachwissens wird definiert, *was* für eine Funktionalität das interaktive System anbietet. Das mögliche Repertoire dieser Funktionalität ist durch die Ebene zweckbezogener Modelle vorgegeben.

Das Bedienwissen umfasst jenen Teil des für die Benutzung erforderlichen Wissens, der durch die Formen der Interaktion zwischen Benutzer und interaktivem System bedingt ist. Es definiert, *wie* die Funktionalität des Systems genutzt werden kann. Geht man davon aus, dass den Benutzern eines interaktiven Systems der Zweck desselben bekannt ist, so können dennoch Fachwissen und Bedienwissen der Benutzer unabhängig voneinander unterschiedlich ausgeprägt sein. Durch entsprechende Dokumentation und durch zusätzliche unterstützende Systemkomponenten ist es möglich, Fach- und Bedienwissen der Benutzer zu erweitern.

Bei der Gestaltung der Benutzungsschnittstelle eines Systems, das von Benutzern mit unterschiedlichen Fähigkeiten bedient werden soll, ergeben sich konkurrierende Anforderungen. Je größer beispielsweise eine Schrift dargestellt werden muss, umso weniger Information lässt sich in einem flächenbegrenzten Bereich darstellen. Die nebenläufige Vermittlung von Bedienwissen lenkt erfahrene Benutzer bei zeitgleicher Nutzung des Systems vom eigentlichen Zweck ab und senkt damit die Effizienz. Es muss daher ein Abwägungsprozess stattfinden, an dem idealerweise die zukünftigen Benutzer beteiligt sein sollten. Um die Bedienung durch unterschiedliche Typen von Benutzern angemessen zu unterstützen, sind sowohl individuell anpassbare Benut-

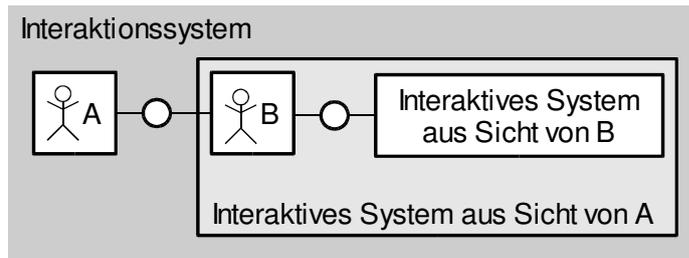


Abbildung 3.10: Anwender und Benutzer

zungsschnittstellen denkbar wie auch Benutzungsschnittstellen, die gleichzeitig unterschiedliche Formen der Interaktion mit Benutzern unterschiedlichen Typs zulassen.

Um die Akzeptanz eines interaktiven Systems sicherzustellen, sind bei der Gestaltung einer Benutzungsschnittstelle ebenfalls emotionale und kulturelle Aspekte zu berücksichtigen. Zum einen sind die Begriffswelt der verwendeten Bezeichner und die Wahl von Symbolen, Farben und Formen zu nennen. Zum anderen gehört zur Akzeptanz interaktiver Systeme die Tatsache, dass manche Menschen zu bestimmten Zwecken lieber mit einem unmittelbaren menschlichen Gegenüber kommunizieren als mit einem technischen System arbeiten.

In manchen Fällen ist es daher sinnvoll, dem Nutzer eines Dienstes einen menschlichen Interaktionspartner gegenüber zu stellen. So ergibt sich das in Abbildung 3.10 gezeigte Szenario. Akteur A nutzt die Dienste eines Akteurs B, der sich seinerseits dazu wieder eines interaktiven Systems bedient. Aus Sicht von A kann das Gebilde aus Akteur B und dem von ihm genutzten System als Ganzes ebenfalls als interaktives System aufgefasst werden. Vergleicht man mit Abbildung 3.1, so entspricht Akteur B dem Schnittstellenadapter SA und das von ihm genutzte interaktive System dem funktionalen Kern. In solchen Fällen entspricht es dem üblichen Sprachgebrauch, den unmittelbaren Benutzer der technischen Komponente, den Akteur B, als Benutzer und den Akteur A als Anwender des interaktiven Systems zu bezeichnen.

3.4.3 Interaktionsadapter

3.4.3.1 Identifikation informationeller Werte

Aus Sicht eines zweckbezogenen Modells besteht eine Benutzungsschnittstelle eines interaktiven Systems aus einem oder mehreren informationellen Kanälen zwischen einem Benutzer und den Komponenten des interaktiven Systems, die im allgemeinen sowohl als Sender als auch als Empfänger von Informationen agieren können. Auf jedem der Kanäle kann man Folgen einzelner informationeller Werte beobachten. Prinzipiell ist es zweckmäßig, auf der Ebene interaktionbezogener Modelle drei Formen der Übertragung, das heißt der Identifikation, eines informationellen Wertes zu unterscheiden:

- Nichtinteraktive einschrittige Wertidentifikation

- Nichtinteraktive mehrschrittige Wertidentifikation
- Interaktive Wertidentifikation

Im Falle einer nichtinteraktiven einschriftigen Wertidentifikation wird die Information, der zweckbezogene Wert, durch genau eine zeitlich konstante Form, den interaktionsbezogenen Wert, identifiziert. Bungert bezeichnet einen solchen Wert als Ortsform [Bun98], die durch eine wertunmittelbare Interpretierbarkeit [Wen91] gekennzeichnet ist. Idealerweise hat man die Vorstellung, dass ein solcher Wert dem Empfänger vom Sender in Folge einer einschriftigen Sendeoperation zur Verfügung gestellt wird. Dies geschieht im Rahmen eines sogenannten elementaren Sendeoperation, der durch Beginn und Ende eines Übergangsintervalles definiert ist. Innerhalb dieses Intervalls besitzen auf dem Kanal beobachtbare Übergangswerte für den Empfänger keine verbindliche Bedeutung. Nach Abschluss des Übergangsintervalles ist auf dem Kanal der informationstragende Wert beobachtbar, den der Empfänger während eines ebenfalls elementaren Empfangsoperation wahrnehmen kann. Die Art des Identifizierens — Zeigen, Benennen oder Umschreiben —, die Struktur der übertragenen Information und der interne Ablauf des Identifikationsoperation beim Empfänger spielen dabei keine Rolle. Abbildung 3.11 liegt ein System zugrunde, in dem ein Mensch die Helligkeit zweier Lampen steuern kann. Im konkreten Fall kann der Benutzer die Helligkeit über je zwei Tasten heller oder dunkler einstellen. Jedem Element des zweckbezogenen Eingaberepertoires („Lampe1 heller“, „Lampe2 heller“, „Lampe1 dunkler“, „Lampe2 dunkler“) ist genau eine einschriftige Eingabeform („A“, „B“, „a“, „b“) zugeordnet. Als Beispiel für eine einschriftige Ausgabe kann das Aufleuchten einer Warnlampe gelten, wodurch ein erläuternder Warnhinweis lesbar wird. Wie ein Leser aus diesem Hinweis die Information ableitet, bleibt ihm überlassen. Durch den Sender ist jedoch sicherzustellen, dass dem Leser ausreichend Zeit zur Wahrnehmung der Nachricht zur Verfügung gestellt wird. Das Übergangsintervall zwischen leuchtendem und nichtleuchtendem Warnhinweis kann in diesem Beispiel aufgrund der geringen Zeitspanne aus Sicht eines potentiellen Warnungsempfängers vernachlässigt werden.

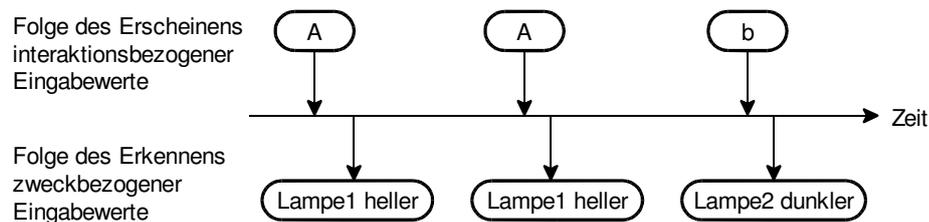


Abbildung 3.11: Einschriftige Identifikation zweckbezogener Werte

Im Falle einer nichtinteraktiven mehrschrittigen Wertidentifikation wird eine Information, der zweckbezogene Wert, durch eine mehrelementige Folge zeitlich voneinander abgrenzbarer Formen identifiziert. Nach [Bun98] wird eine solche Folge als Zeitform bezeichnet. Ein solcher Werteverlauf ist auf der gewählten Betrachtungsebene als Ergebnis einer Folge elementarer Sendeoperationen anzusehen, den ein Empfänger im Rahmen einer entsprechenden Folge von Empfangsoperationen wahrnimmt. Durch

Werteverlaufsinterpretation [Wen91] kann der Empfänger dem Verlauf eine Bedeutung zuordnen. Da der Empfänger auf den Verlauf keinerlei Einfluss nehmen kann, handelt es sich um einen nicht-interaktiven Prozess. Die Realisierungsform der einzelnen Folgeelemente selbst, ist Gegenstand einer tieferen Betrachtungsebene, wobei wiederum alle drei Formen der Wertidentifikation möglich sind. Abbildung 3.12 zeigt eine Möglichkeit der Lampensteuerung durch zweielementige Eingabefolgen, wobei zunächst die Lampe und anschließend die gewünschte Helligkeitsänderung anzugeben ist.

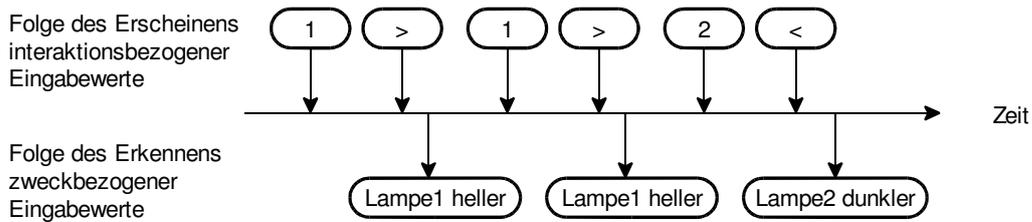


Abbildung 3.12: Nichtinteraktive, mehrschrittige Identifikation zweckbezogener Werte

Im Falle einer interaktiven oder dialoggestützten Wertidentifikation wird eine Information, der zweckbezogene Wert, durch eine mehrelementige Folge abwechselnder Dialogschritte von Sender und Empfänger identifiziert. Da man aus Sicht des zweckbezogenen Modells eindeutig Empfänger und Sender unterscheiden kann, ist es angemessen, von einem interaktiven Sendeprozess zu sprechen. Die Realisierungsform der einzelnen Folgeelemente selbst, ist Gegenstand einer tieferen Betrachtungsebene, wobei wiederum alle drei Formen der Wertidentifikation möglich sind. Eine typische Form der interaktiven Identifikation, die schrittweise Auswahl aus dem eigens zu diesem Zweck präsentierten Repertoire, dem sogenannten Menü, ist am Beispiel einer Lampensteuerung in Abbildung 3.13 gezeigt.

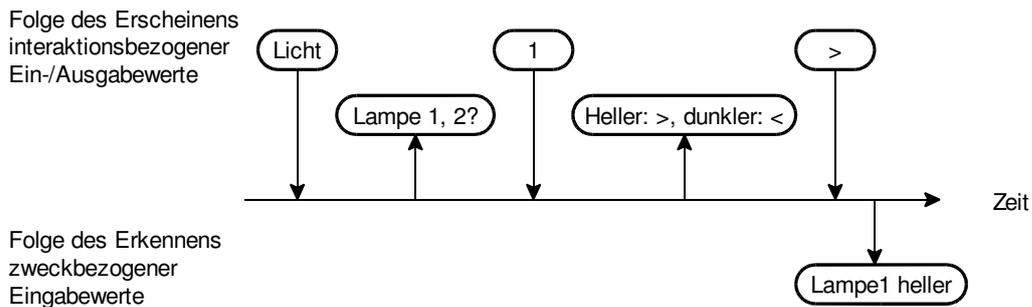


Abbildung 3.13: Interaktive Identifikation zweckbezogener Werte

3.4.3.2 Abgrenzung von Interaktionsadaptern

Der Wechsel von der Betrachtung zweckbezogener Werte auf die Betrachtung interaktionsbezogener Werte äußert sich in einem veränderten Modell der Benutzungsschnittstelle. Wie der vorangegangene Abschnitt gezeigt hat, sind Repertoire und Werteverlauf an der Benutzungsschnittstelle anders geartet als beim zweckbezogenen Modell.

Eine verständliche Aufbaustruktur, bei der ein Bezug zum zweckbezogenen Modell erhalten bleibt, ergibt sich, wenn jeder Kanal, der Bestandteil der Benutzungsschnittstelle des zweckbezogenen Modells ist, ist auf der Ebene des interaktionsbezogenen Modells durch mindestens einen Interaktionsadapter, der zwischen Benutzer und der entsprechenden Komponente des zweckbasierten Kerns vermittelt, realisiert ist. Zweck dieser Komponenten ist die Abbildung zwischen zweckbezogenen Ein-/Ausgabewerten auf Seiten des Kerns und interaktionsbezogenen Ein-/Ausgabewerten auf Seiten des Benutzers. Dabei lassen sich zwei Fälle unterscheiden:

- Im Falle einer einschrittigen Identifikation informationeller Werte ist ein sogenannter Formadapter für die Realisierung des entsprechenden zweckbezogenen Kanal zuständig.
- Im Falle einer mehrschrittigen Identifikation informationeller Werte ist ein sogenannter Schrittfolgenadapter für die Realisierung des entsprechenden zweckbezogenen Kanals zuständig.

Prinzipiell steht der Grundgedanke von Interaktionsadaptern im Einklang zu den heutigen einschlägig bekannten Entwicklungsansätzen. Neu ist die Sichtweise der Realisierung unterschiedlicher informationeller Kanäle, die durch die Aufbaustruktur eines zweckbezogenen Modells identifiziert sind. Die Identifikation von Interaktionsadaptern stellt keine Erweiterung eines zweckbezogenen Modells dar, sondern zeigt ein anderes Modell des Systems auf einer anderen Betrachtungsebene. Dieser wesentliche Unterschied wird in Kapitel 4 durch die Analyse gängiger Beschreibungsansätze verdeutlicht.

3.4.3.3 Formadapter

Abbildung 3.14 zeigt die Realisierungsbeziehung zwischen einem zweckbezogenem Modell und einem interaktionsbezogenem Modell bei ausschließlicher Verwendung von Formadaptern. Aus Gründen der Überschaubarkeit zeigt die Darstellung des zweckbezogenen Modells nur eine einzige Komponente eines interaktiven System, bezeichnet mit A_Z , die über einen Ein- und einen Ausgabekanal mit einem Benutzer verbunden ist. Auf Ebene des interaktionsbezogenen Modells sind diese Kanäle als Bestandteil der Benutzungsschnittstelle realisiert, an welcher die materiell-energetischen Formen zur Identifikation der zweckbezogenen Ein- bzw. Ausgaben wahrnehmbar sind. Die Realisierung der Komponente A_Z enthält zu diesem Zweck je einen Formadapter für die Eingaben und einen für die Ausgaben, die mit Komponente A_K des zweckbasierten Kerns kommunizieren.

Formadapter haben die Aufgabe von Zuordnern, die eine Abbildung von identifizierenden Eingabeformen auf zweckbezogene Eingabewerte bzw. von zweckbezogenen Ausgabewerten auf identifizierende Ausgabeformen vornehmen. Neben der Form abgebildeter Werte besitzt jeder Formadapter eine dauerhafte charakteristische Form. Der Teil dieser Form, der an der Benutzungsschnittstelle wahrnehmbar ist und einem Benutzer unabhängig von der Form der kommunizierten Werte eine Identifikation des

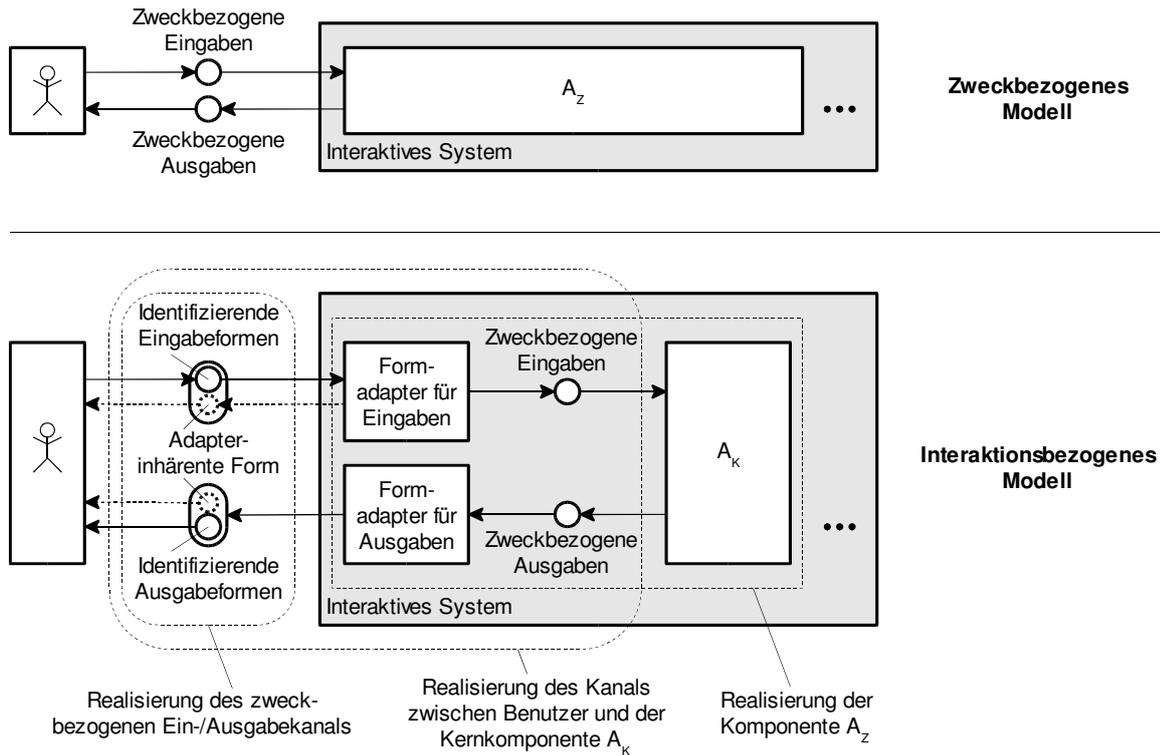


Abbildung 3.14: Realisierung zweckbezogener Kanäle durch Formadapter

Kanals und somit des Kommunikationspartners erlaubt, wird im Rahmen dieser Arbeit als *adapterinhärente Form* bezeichnet. Als Beispiel betrachte man eine Notrufsäule. Eine solche Säule stellt einen Formadapter dar, der einen bidirektionalen Kanal zu einer Notrufzentrale realisiert. Die beobachtbaren Ein- und Ausgabeformen bilden das hörbare Gespräch zwischen einem Hilfesuchenden und der Notrufzentrale, die in diesem Fall die Art der Notlage und Möglichkeiten der Hilfe identifizieren. Die inhärente Form der Notrufsäule ermöglicht einem Hilfesuchenden, eine Notrufsäule als solche und damit einen Kanal zur Notrufzentrale zu erkennen. Sind an einer Benutzungsschnittstelle die Formen unterschiedlicher Interaktionsadapter wahrnehmbar, sollte sich für einen Benutzer durch eine entsprechende Gestaltung der unterschiedlichen adapterinhärenten Formen auch der Eindruck unterschiedlicher Kanäle zum System ergeben.

Aufgrund der eingangs vorgenommenen Ableitung der Komponente A_K aus A_Z folgt, dass Eingabe- und Ausgaberepertoire beider Komponenten identisch ist. Daraus auf eine Identität von A_Z und A_K zu schließen, wäre ein Fehler, der zu Problemen führen kann. Wie Abbildung 3.14 zeigt, sind die Schnittstellen von A_Z und A_K unterschiedlichen Orten zugeordnet. Während die Schnittstelle der Komponente A_Z Bestandteil der Benutzungsschnittstelle ist, befindet sich die Schnittstelle der Komponente A_K im Inneren des interaktiven Systems. Sie ist für seine Benutzer unsichtbar. Entscheidend ist dieser Unterschied, wenn man das zeitliche Verhalten beider Komponenten miteinander vergleicht. Zwischen den Eingaben und Ausgaben an den Schnittstellen eines Formwandlers besteht eine unvermeidliche zeitliche Verzögerung. Diese Verzögerung

ist das Kennzeichen eines jeden realen Zuordners und kann je nach Art der Realisierung unterschiedlich groß ausfallen (vgl. Abbildung 3.11). Aus diesem Grund kann durch geeignete zeitliche Abstimmung der Komponenten A_Z und A_K zwar garantiert werden, dass an deren Schnittstellen dieselben Wertefolgen nicht aber zeitgleiche Werteverläufe beobachtbar sind. Beim Entwurf von A_K verringert sich daher die für A_Z spezifizierte zulässige Zeit einer Ein- bzw. Ausgabeoperation um die für die Formwandlung erforderliche Spanne. Eine Unterscheidung von A_Z und A_K ist somit nicht nur eine Frage der klaren Trennung zweier Betrachtungsebenen. Eine Gleichsetzung ist nicht zulässig, da sie unter Umständen zu zeitlichen Problemen führen kann.

Würde man die Realisierung des Kanals zwischen dem Benutzer und der Komponente A_K als hierarchische Verfeinerung des zweckbezogenen Kanals ansehen, würde dies einer Gleichsetzung von A_Z und A_K gleichkommen, was aufgrund der obigen Betrachtungen unzulässig ist. Ebenfalls ist es unzulässig, die Struktur aus der Kernkomponente A_K und den beiden Formwandlern als hierarchische Verfeinerung von A_Z anzusehen. Eine Verfeinerung des inneren Aufbaus erklärt nicht den Übergang von der Betrachtung einer Folge von Informationen auf zweckbezogener Ebene und zur Betrachtung einer Folge informationstragender Formen auf interaktionsbezogener Ebene. Dieser Übergang ist nur durch einen Betrachtungsebenenwechsel zu erklären.

3.4.3.4 Schrittfolgenadapter

Abbildung 3.15 zeigt den Übergang von einem zweckbezogenem Modell zur Ebene interaktionsbezogener Modelle bei Verwendung von Schrittfolgenadaptern. Aus Gründen der Überschaubarkeit zeigt die Darstellung des zweckbezogenen Modells nur eine einzige Komponente, bezeichnet mit A_Z , die lediglich über einen Eingabekanal mit einem Benutzer verbunden ist. Auf der Ebene des interaktionsbezogenen Modells ist dieser Kanal als Bestandteil der Benutzungsschnittstelle realisiert, an der zur Identifikation eines zweckbezogenen Eingabewertes e_Z die Dialogfolge E_D beobachtbar ist. Der gerichtete Eingabekanal auf der Ebene des zweckbezogenen Modells ist im Falle der dialoggestützten Identifikation von e_Z durch einen Kanal zur Eingabe und einen weiteren zur Ausgabe realisiert. Auf diesen Kanälen sind die Folge von Eingaben e_{Din} und die der Ausgaben e_{Dout} beobachtbar.

Der Schrittfolgenadapter S_D hat die Aufgabe eines Automaten. Bei Eingabe eines Wertes $e_{Din} \in E_D$ reagiert er abhängig vom Dialogzustand mit einer entsprechenden Ausgabe und wechselt in den Folgezustand entsprechend dem durch die Zustandsübergangsfunktion definierten Protokoll. Nach diesem Modell ist jede Ausgabe als strukturierter Wert mit zwei Komponenten anzusehen, einer möglichen Nachricht an den Benutzer und einer möglichen Nachricht an die Kernkomponente A_K . Bei Erkennung einer zweckbezogenen Eingabe wird der entsprechende Wert e_Z der Kernkomponente übergeben. Zusätzlich kann bei jedem Dialogschritt zu Feedback- oder Unterstützungszwecken eine Ausgabe $e_{Dout} \in E_D$ an den Benutzer erfolgen.

Als Beispiel für einen häufig vorkommenden Fall sei das Ausfüllen eines Formulars betrachtet. Auf zweckbezogener Ebene als ein informationeller Wert betrachtbar, wird die Information des Formulars in einem mehrschrittigen Prozess zusammengestellt bis der Ausstellende durch Eingabe eines ausgezeichneten Eingabezeichens signalisiert,

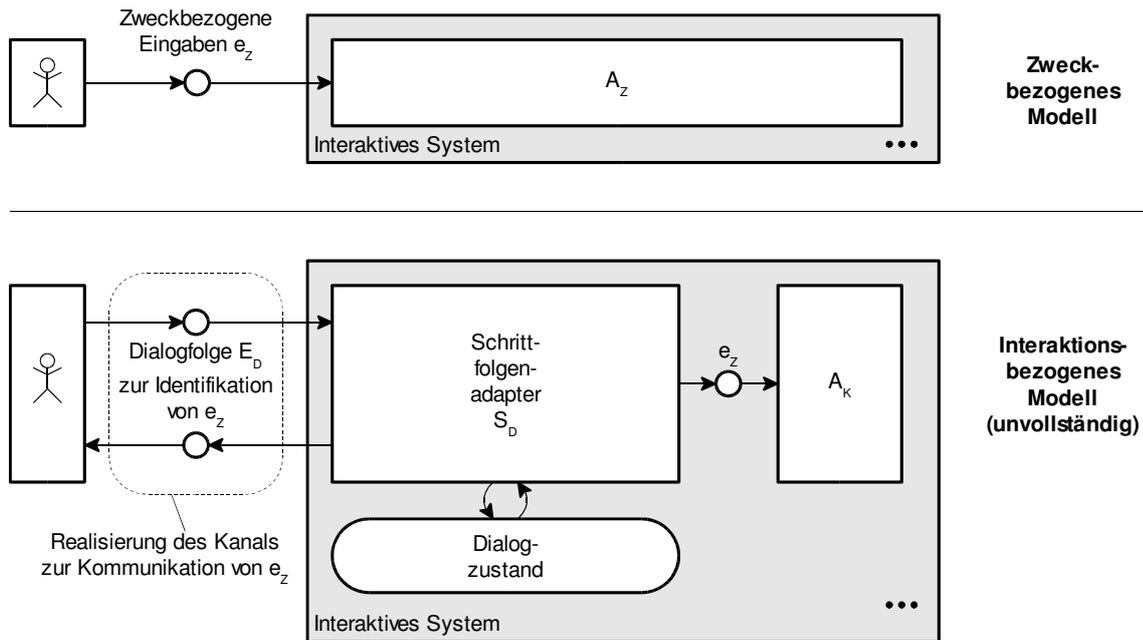


Abbildung 3.15: Realisierung zweckbezogener Kanäle durch Schrittfolgenadapter

dass die Daten vollständig sind und übermittelt werden können. In solchen Fällen ist es zweckmäßig, sich den Dialogzustand getrennt in einen operationellen Anteil und einen Steuerzustandsanteil vorzustellen. Zur Beschreibung des Adapterverhaltens genügt es dann, unabhängig von den jeweiligen operationellen Daten (in diesem Falle der Inhalt des Formulars) den Graph der Steuerzustandsübergänge darzustellen.

Die mehrschrittige Ausgabe zweckbezogener Werte e_z ist in gleicher Weise durch Schrittfolgenadapter realisierbar. Die schriftliche, seitenweise Darstellung einer längeren Beschreibung ist ein typisches Beispiel für eine solche Ausgabe. Eine in diesem Zusammenhang übliche Dialogeingabe von Seiten des Benutzers wäre das Anfordern der nächsten Seite. Hat man die Vorstellung, dass die Struktur des informationellen Wertes quasi einen Raum aufspannt, der vom Benutzer schrittweise erforscht werden kann, wird die schrittweise Wahl des betrachteten Ausschnittes häufig als Navigation bezeichnet [Thi00].

3.4.3.5 Mehrstufige Wertabbildung

Lassen sich bei der Gestaltung informationstragender Formen aufeinander aufbauende Entwurfsentscheidungen von einander abgrenzen, kann man sich die Abbildung einzelner informationeller Werte auf konkrete informationstragende Formen an der Benutzungsschnittstelle prinzipiell als das Ergebnis einer mehrstufigen Kette von Interaktionsadaptern vorstellen. Dies ist immer dann der Fall, wenn sich die Bedeutung einer Symbolkette als Ergebnis einer Interpretationskette (Kapitel 2.2.4) ergibt. Bei konsequenter Anwendung des Konzeptes ergibt sich im Falle einer n -stufigen Interpretationskette eine n -stufige Kette von Interaktionsadaptern zur Realisierung eines zweckbezogenen Kanals.

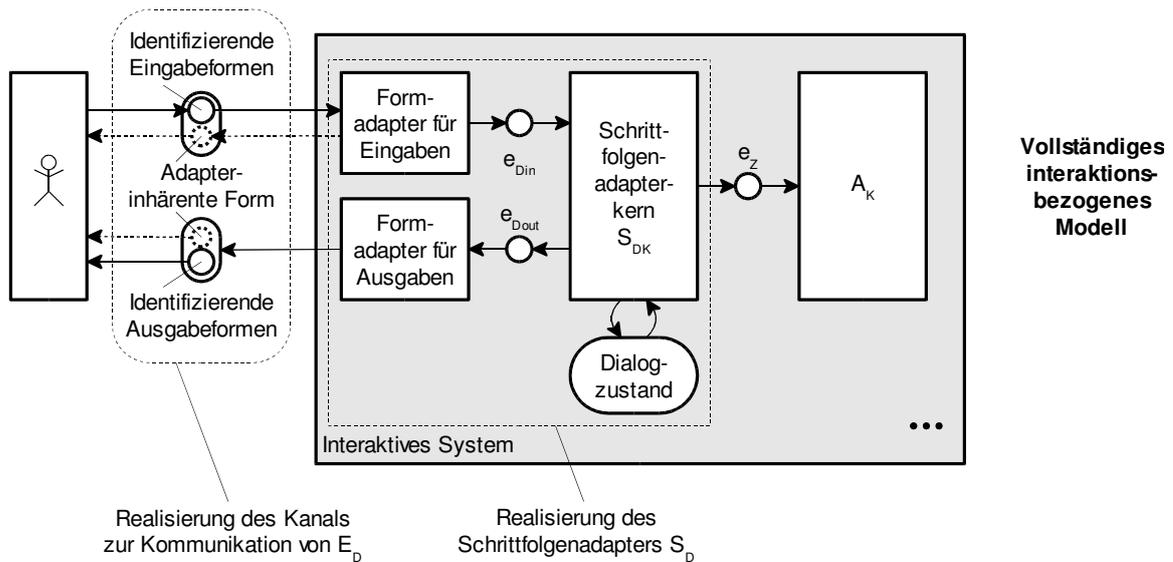


Abbildung 3.16: Trennung von Dialog- und Symboldesign

Betrachtet man den in Abbildung 3.15 gezeigten Schrittfolgenadapter S_D , so kann es zweckmäßig sein, die zur Identifikation eines bestimmten Repertoires informationeller Werte verwendeten Wertefolgen E_D selbst als Folge informationeller Werte zu betrachten. Unter Vernachlässigung der adapterinhärenten Form ist es bei solchen Strukturen möglich, Semantik und Syntax der Folgen unabhängig von der konkreten Form der Symbole an der Benutzungsschnittstelle zu betrachten.

Ein Modell muss als *interaktionsunvollständig* bezeichnet werden, wenn es die informationstragende, materiell-energetische Form der Werte e_{Din} und e_{Dout} nicht festlegt. Durch einen Betrachtungsebenenwechsel innerhalb der Ebene interaktionsbezogener Modelle kann die informationstragende Form der Werte e_{Din} und e_{Dout} sichtbar werden. Das entstehende Modell kann als *interaktionsvollständig* bezeichnet werden. Die in Abschnitt 3.4.3.2 vorgestellten Zweckmäßigkeitsüberlegungen können dabei, wenn es angemessen erscheint, rekursiv angewendet werden. Das Modell eines Interaktionsadapters wie S_D kann, wie Abbildung 3.16 zeigt, auf tieferer Betrachtungsebene als Komposition angesehen werden, die aus einem Adapterkern und einem oder mehreren Interaktionsadaptern, hier Formwandlern, besteht. Übertragen auf das Beispiel der Lampensteuerung könnte so — je nach Wahl der Interaktionsadapter — der Dialog akustisch in englischer oder deutscher Sprache oder visuell und gestenbasiert erfolgen. Das in Abbildung 3.13 gezeigte Protokoll könnte in allen Fällen eingehalten werden.

Die Anzahl unterscheidbarer Betrachtungsebenen innerhalb der Ebene interaktionsbezogener Modelle ist vom betrachteten System abhängig und gleichermaßen eine Frage der Zweckmäßigkeit. Versuche verbindliche Ebenen zu identifizieren sind stets mit der vorgezogenen Festlegung von Entwurfsentscheidungen verbunden. Im Hinblick auf die Nutzer interaktionsbezogener Modelle muss man hinterfragen, ob eine solche Strukturierung der Gestaltung und Vermittlung des Schnittstellenverhaltens dienlich ist oder ob sie als Ergebnis vorweggenommener Realisierungsüberlegungen anzusehen ist.

Im Falle der nicht interaktiven, mehrschrittigen Identifikation von Werten durch Symbolfolgen ist die Unterscheidung einer semantischen, einer syntaktischen und einer lexikalischen Ebene üblich. Die Beschreibung zulässiger Eingaben erfolgt in diesen Fällen häufig über Grammatiken. Die einschlägigen Ansätze zum Entwurf von Benutzungsschnittstellen geben allerdings wenig Auskunft darüber, wie in den Entwurfsprozess eingebundene Anwender solch mehrstufige Ansätze handhaben können. Der hohe Stellenwert von Prototypen bei der Entwicklung von Benutzungsschnittstellen (vgl. Kapitel 4) mag Anzeichen dafür sein, dass viele Menschen das Bedürfnis haben, eine möglichst direkte Abbildung abstrakter Vorstellungen auf konkret wahrnehmbare Formen zu erleben.

Es ist die Aufgabe von Benutzungsschnittstellendesignern hier mit didaktischem Gespür den richtigen Mittelweg zu finden. So wird in [Dzi83] zur Beurteilung von Benutzungsschnittstellen vorgeschlagen, die Betrachtung konkreter Ein- und Ausgabeformen grundsätzlich von der Gestaltung des Dialoges zu trennen. Das in Abbildung 3.16 gezeigte Aufbaumodell kann in diesem Sinne als mögliche Verfeinerung des ebenfalls 3-stufigen „IFIP-Modells für Benutzerschnittstellen“, das in [Dzi83] vorgestellt wird, betrachtet werden. In vielen Fällen wird die Vorstellung ausreichen, dass *ein* zweckbezogener Kanal durch *einen* Interaktionsadapter realisiert wird.

3.4.4 Gebräuchliche Gestaltungsmuster

Im folgenden wird für ausgewählte, häufig angewendete Gestaltungsmuster von Benutzungsschnittstellen der Zusammenhang mit dem vorgestellten Adapteransatz verdeutlicht. Dabei soll zweierlei klar werden: Einerseits lassen sich alle Interaktionsformen verständlich durch die Verwendung von Formadaptern und Schrittfolgenadaptern erklären. Eine Auswahl ist wegen der Fülle unterschiedlicher Gestaltungsmuster [Tid99] erforderlich und nicht als Einschränkung des Ansatzes zu verstehen. Andererseits soll deutlich werden, dass sich manche Interaktionsformen durch unterschiedliche Modelle erklären lassen, unter denen im konkreten Anwendungsfall auszuwählen ist.

3.4.4.1 Abbildung strukturierter Werte

Im Falle strukturierter, informationeller Werte ist es bei der Gestaltung grafischer Benutzungsschnittstellen üblich, einzelnen Komponenten eines solchen Wertes eindeutig bestimmten Bereiche der Benutzungsschnittstelle zuzuordnen. Wie Abbildung 3.17 zeigt, kann man damit zwei prinzipiell unterschiedliche Vorstellungen verbinden.

In einem Falle hat man die Vorstellung eines Formadapters, der eine Abbildung zwischen einem strukturierten informationellen Wert e_Z und einer einzelnen dieser Struktur ähnlichen, informationstragenden Form an der Benutzungsschnittstelle vornimmt.

Im anderen Falle hat man die Vorstellung getrennter Interaktionselemente für den Zugriff auf die einzelnen Komponenten des zu kommunizierenden informationellen Wertes e_Z . Mit jedem dieser Interaktionselemente ist eine inhärente Erscheinung verbunden, die den Benutzer unabhängig von der momentanen Belegung durch einen

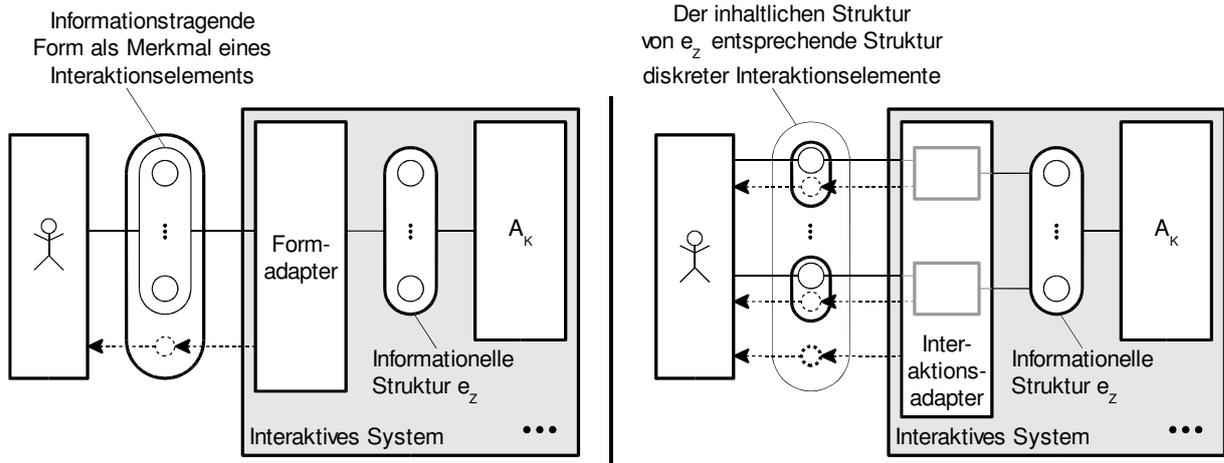


Abbildung 3.17: Alternative Modelle der Kommunikation strukturierter Werte

Wert e_z ein eigenständiges Objekt erkennen lässt. Als Beispiel stelle man sich die Ausgabe eines Adresdatensatzes vor, bei dem für Name, Straße und Wohnort getrennte Anzeigeelemente vorgesehen sind. Zweckbezogene Werte, die eine hierarchische Struktur besitzen, äußern sich bei grafischen Benutzungsschnittstellen üblicherweise in einer grafischen Enthaltenseinsbeziehung einzelner Interaktionselemente, die als Bestandteil eines Interaktionsadapters anzusehen sind. Ein Interaktionsadapter ist in Abgrenzung zu den einzelnen Interaktionselementen als vollständige Realisierung eines zweckbezogenen Kanals anzusehen. Diese Vorstellung lässt sich durch geeignete gestalterische Maßnahmen, wie etwa die räumliche Nähe der einzelnen Interaktionselemente, herstellen. Diese kann durch eine zusätzliche adapterinhärente Form des Gesamtgebildes, beispielsweise in Form einer Umrahmung, noch verstärkt werden.

Die Frage, welches Modell im Hinblick auf ein bestimmtes System zweckmäßiger ist, muss mit Blick auf eine der Bedienung angemessene Vorstellung der Benutzer getroffen werden und darf sich nicht von Realisierungsgedanken leiten lassen.

3.4.4.2 Abbildung von Werten überschaubaren Repertoires

In vielen Fällen sind die Werte, die über einen Kanal der Benutzungsschnittstelle kommunizierbar sind, Elemente eines Repertoires, das auf einen Blick überschaubar und ohne Mühe explizit aufzählbar ist. In solchen Fällen kann es angemessen sein, jedes einzelne Element des Repertoires durch genau ein entsprechendes Interaktionselement zu repräsentieren. Zu jedem beliebigen Zeitpunkt kann maximal eines der Elemente die Übertragung des entsprechenden informationellen Wertes signalisieren. Als Beispiel hierfür kann die Tastatur eines Texterfassungssystem gesehen werden, wobei die einzelnen Tasten die unterschiedlichen Interaktionselemente darstellen.

Mit einer solchen Schnittstelle ist ebenfalls das Modell eines Schnittstellenadapters verbunden, der als Komposition mehrerer Interaktionselemente anzusehen ist. Eine Vorstellung, bei der jedes Interaktionselement einen getrennten Interaktionsadapter als

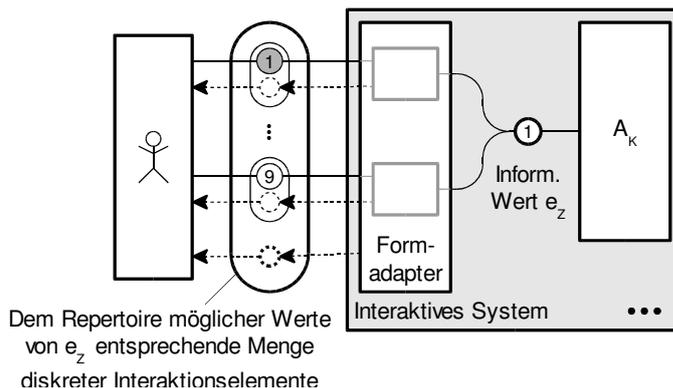


Abbildung 3.18: Repertoire-identifizierende Abbildung von Werten

Realisierung eines separaten Kanals darstellt, ist denkbar aber im Hinblick auf die Abbildung eines zweckbezogenen Modells unzweckmäßig.

Typischerweise wird dieses Konzept zur Identifikation von Eingabeinformationen angewendet. Wie Abbildung 3.18 zeigt, resultiert daraus ein potentieller Konflikt durch zeitgleiche, konkurrierende Eingaben über verschiedene Interaktionselemente. Die Lösung dieses Konfliktes ist sicherzustellen und kann entweder durch eine Kopplung der Interaktionsadapter untereinander oder durch eine zusätzliche Komponente erfolgen. Sie muss auf Ebene des interaktionsbezogenen Modells aber nicht erklärt werden.

3.4.4.3 Interaktion mit virtuell gegenständlichen Objekten

Als virtuell wird ein Objekt bezeichnet, wenn es ein Mensch (ausgehend von seinem Erleben) als der „Möglichkeit nach vorhanden“ [Dro74] betrachtet wird, ohne dass dieses Objekt in Wirklichkeit existieren muss. In diesem Zusammenhang sind Träume als das Erleben virtueller Geschehnisse anzusehen, wenn der Träumende diese als real erlebt. Ein virtuell gegenständliches Objekt ist von einem Menschen (idealerweise) in derselben Art und Weise erlebbar und somit sichtbar, hörbar, fühlbar, begreifbar wie ein entsprechendes tatsächlich existentes Objekt.

Bei vielen interaktiven Systemen wird versucht, einen solchen Eindruck durch eine entsprechend gestaltete Benutzungsschnittstelle zu erzeugen. Eine Voraussetzung dafür ist die Präsentation einer möglichst wirklichkeitsnahen Abbildung des zu kommunizierenden Sachverhaltes beispielsweise auf einem hochauflösenden Bildschirm. Teil des virtuellen Erlebens ist die Möglichkeit eines Benutzers scheinbar auch wirklichkeitsgetreu mit dem virtuellen Objekt interagieren zu können. Dazu gehört beispielsweise die Möglichkeit, das Objekt von allen Seiten in seiner Umgebung betrachten, bewegen oder verformen zu können. Glücklicherweise sind Menschen darin überaus anpassungsfähig, auch auf den ersten Blick unvollkommene Annäherungen an das reale Erleben als virtuelle Realität zu akzeptieren. So ergibt sich beispielsweise der Eindruck einer direkten Manipulation [HHN86] von Objekten, wenn ein interaktives System zur Präsentation von Objekten genutzt wird, das entsprechende Gesten eines Benutzers zum Bewegen oder Verformen der Objekte erkennt. Mit kaum wahr-

nehmbarer Verzögerung wird hier eine Anpassung der Objektdarstellung durchgeführt. In der Vorstellung des Benutzers wird durch die direkte Kopplung von Gesten und Präsentation die Abbildung des Objektes zum virtuellen Objekt. Bei grafischen Benutzungsschnittstellen wird so die Möglichkeit ausgenutzt, anstelle echter Schalter Schaltflächen für virtuelle Schalter [FD82] zu präsentieren. Zur Bedienung eines solchen virtuellen Schalters muss der Benutzer ihn mithilfe eines geeigneten Formadapters, beispielsweise einer sogenannten Maus, identifizieren und betätigen.

3.4.4.4 Sekundäre Kanäle und Interaktionsassistenten

Als sekundäre Kanäle sollen Kanäle bezeichnet werden, die nicht als Realisierung eines Kanals der Benutzungsschnittstelle aus einem zweckbezogenen Modell angesehen werden können. Dabei handelt es sich entweder um sogenannte Feedback-Kanäle, anhand derer ein Benutzer die Wirksamkeit seiner Eingaben erkennen kann, oder um Kanäle zu sogenannten Interaktionsassistenten. Interaktionsassistenten sind aktive Systemkomponenten, die einen Benutzer bei Benutzung des Systems unterstützen. Sie sind aber nicht zur Realisierung des zweckbezogenen Modells erforderlich. Typischerweise handelt es sich um Komponenten, die einem Benutzer Hilfestellung leisten, und so eine effizientere Benutzung des Systems erlauben. Als Beispiele sollen die Vermittlung von Bedien- oder Fachwissen genannt sein. Prinzipiell können diese Aktivitäten nebenläufig zu den Aktivitäten der übrigen Komponenten des interaktiven Systems erfolgen. Angesichts der Fülle denkbarer Interaktionsassistenten ist eine Klassifikation entsprechend dem Grad der Kopplung mit dem übrigen interaktiven System zweckmäßig. Eine solche Klassifikation ist in Abbildung 3.19 dargestellt.

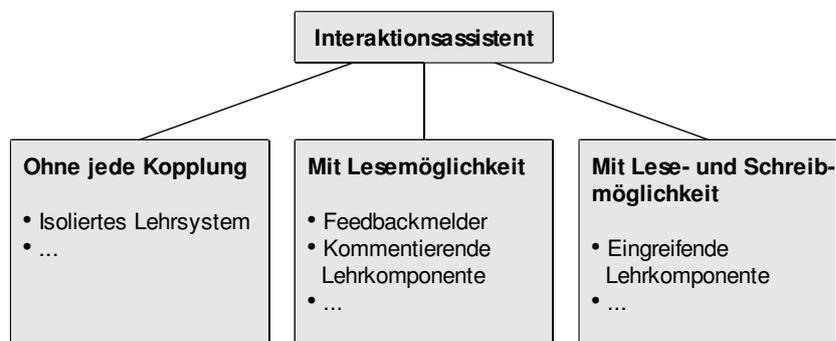


Abbildung 3.19: Klassifikation von Interaktionsassistenten

Im einfachsten Fall könnte ein Interaktionsassistent ein Hilfesystem darstellen, das isoliert vom übrigen System betrachtbar ist und das ein Benutzer bei Fragen zu Rate ziehen kann. In anderen Fällen kann ein Interaktionsassistent den Informationsfluss zwischen Benutzer und einer oder mehreren Kernkomponenten des Systems sowie deren Zustand beobachten. Abbildung 3.20 zeigt eine mögliche Einbindung, bei der sich die Beobachtung des gezeigten Interaktionsassistenten auf eine einzige Kernkomponente beschränkt. Je nach Art der angebotenen Unterstützung kann es sich dabei um eine Komponente des zweckbasierten Kerns oder den Kern eines Schrittfolgenadapters handeln. Aus dem beobachtbaren Verhalten lassen sich Vermutungen über Infor-

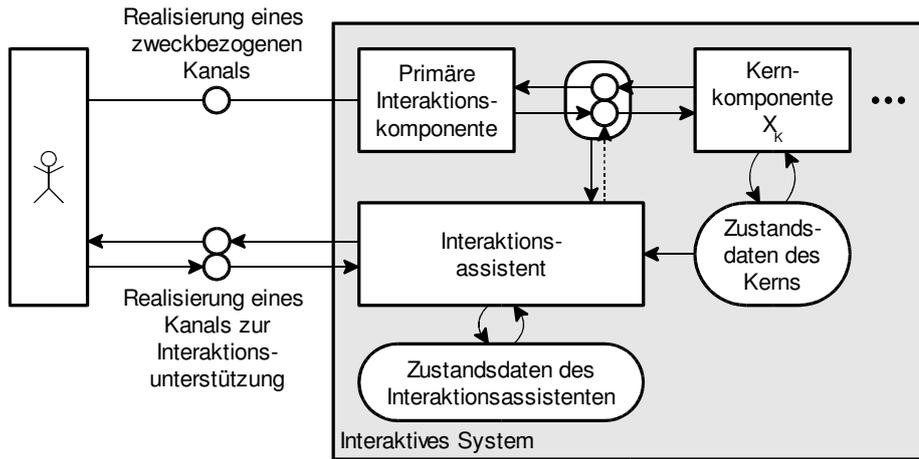


Abbildung 3.20: Integration von Interaktionsassistenten

mationen ableiten, die ein Benutzer zu einem bestimmten Zeitpunkt wahrscheinlich benötigt. Ein solcher Interaktionsassistent kann von sich aus den Benutzer mit einer geeigneten Auswahl von Informationen unterstützen.

Ein Beispiel für den Einsatz eines Interaktionsassistenten ist die Einblendung beschreibender Hilfstexte, sogenannter Tooltips, an den Interaktionselementen bei manchen Systemen mit grafischer Benutzungsschnittstelle. Der Vorteil, solche unterstützenden Informationen als das Ergebnis der Aktivität einer zusätzlichen Komponente anzusehen, besteht darin, dass dieses Verhalten andernfalls nur durch eine Erweiterung der bereits vorhandenen Komponenten möglich gewesen wäre. Die betroffenen Komponenten müssten dann neben ihrer eigentlichen Kernfunktionalität zusätzliche Unterstützungsdienste leisten. Dadurch würde die Komplexität ihrer Verhaltensstruktur steigen, was entgegen dem beabsichtigten Ziel der Benutzerunterstützung ihr Verständnis erschweren würde [Kam85]. Um im Falle wechselnder Benutzer eine individuelle Unterstützung zu ermöglichen, kann es daher sogar zweckmäßig sein, ein Repertoire verschiedenartiger Interaktionsassistenten zur Auswahl anzubieten.

In Ausnahmefällen kann es angemessen sein, eine Einflussnahme von Interaktionsassistenten auf das übrige interaktive System vorzusehen. Nimmt die Erklärung eines Interaktionsassistenten beispielsweise Bezug auf bestimmte im aktuellen Dialogzustand deaktivierte Formadapter, so kann es der Wissensvermittlung förderlich sein, in einen Dialogzustand zu wechseln, in dem diese Adapter aktiviert sind. Ein naheliegender Ansatz besteht darin, die Aktivierung durch den Interaktionsassistenten zu unterstützen. Sind beispielsweise bei einem Bildbearbeitungssystem bestimmte Interaktionselemente zur Verwendung von Zeichenebenen nur in einem bestimmten Dialogzustand sichtbar, so könnte ein Hilfesystem, das die Verwendung von Zeichenebenen erklärt, dem Benutzer einen Wechsel in diesen Dialogzustand vorschlagen und diesen gegebenenfalls durchführen. In seiner Wirkung auf das interaktive System übernimmt ein solcher Interaktionsassistent dabei implizit die Rolle eines Interaktionsadapters zur Realisierung eines zweckbezogenen Kanals, auch wenn er im Bezug auf den Zweck dieser Aktion seine Rolle als Interaktionsassistent beibehält. Wie Abbildung 3.20 zeigt, resultiert daraus ein potentieller Konflikt zwischen konkurrierenden Eingaben über

unterschiedliche Kanäle. Die Lösung eines solchen Konfliktes, die in jedem Falle zu gewährleisten ist, ergibt sich häufig bereits implizit durch die sequentielle Nutzung beider Kanäle durch den Benutzer.

Für die Form des Informationsaustausches zwischen Interaktionsassistent und Benutzer gelten die bereits vorstellten Prinzipien der ein- und mehrschrittigen Identifikation und deren Realisierung mithilfe von Form- und Schrittfolgenadaptern. Die Realisierung eines Feedback-Kanales mithilfe eines Formadapters kann als Sonderfall eines Interaktionsassistenten angesehen werden. Als Beispiel für einen solchen Feedback-Kanal kann die Status-Leiste eines Text-Editors gelten, der einen Benutzer über den aktuellen Dialogzustand, wie Bearbeiten oder Einfügen, informiert.

3.4.4.5 Angemessene Schnittstellengestaltung

Die Entwicklung optimaler Benutzungsschnittstellen erfordert das Wissen entsprechend geschulter Spezialisten, die die Möglichkeiten der Form- und Interaktionsgestaltung kennen. Ihnen muss bekannt sein, welche Wirkungen bestimmte Maßnahmen bei den Benutzern hervorrufen und welche Anforderungen sie an die Benutzer stellen. Sie müssen die Fähigkeit besitzen, diese Kenntnisse angepasst auf die konkreten Fähigkeiten und Bedürfnisse der zukünftigen Benutzer umzusetzen. Dies geht weit über die Konzepte der Modellierung und Darstellung, die im Zentrum dieser Arbeit stehen, hinaus. Der interessierte Leser sei auf die einschlägige Literatur verwiesen [Nor01, Shn98]. Die vorgenommene Klassifikation von Benutzerfähigkeiten und Interaktionskomponenten erlaubt jedoch, wie Tabelle 3.1 andeutet, eine zweckmäßig erscheinende Grobeinteilung der bekannten Gestaltungsmuster, die ihre prinzipielle Wirkung erkennen lässt.

	<i>Physische Benutzbarkeit</i>	<i>Kognitive Benutzbarkeit</i>	<i>Reduktion erforderlichen Bedienwissens</i>	<i>Reduktion erforderlichen Fachwissens</i>
<i>Formadapter</i>	durch ergonomische Formwahl, wie z.B. Tastaturdesign	durch Verwendung übersichtlicher, verständlicher Formen, wie z.B. Piktogramme	z.B. durch Verwendung bekannter Symbole und Bedienkonzepte, wie Drehknöpfe	z.B. durch Assoziierbarkeit mit zweckbezogenen Komponenten, z.B. durch Beschriftungen
<i>Schrittfolgenadapter</i>	z.B. durch Bündelung sich wiederholender Operationsfolgen	durch schrittweise Formulierbarkeit des Gewollten	z.B. durch Menüsteuerung	z.B. durch geführte Schrittfolgen mit Erklärungen
<i>Assistenten</i>	---	durch Vermittlung potentiell relevanter Anwendungsdaten, z.B. durch Syntaxprüfer	durch Vermittlung potentiell relevanten Bedienwissens, z.B. durch Tooltips	durch Vermittlung potentiell relevanten Fachwissens, z.B. durch Hilfesystem

Tabelle 3.1: Benutzerspezifische Unterstützung durch Interaktionskomponenten

3.4.5 Kanalmultiplex

Von Kanalmultiplex spricht man dann, wenn in einem System in dem unterschiedliche Gesprächspartner entkoppelte Dialoge über getrennte Kanäle führen, diese abschnittsweise durch eine einzige Vermittlungskomponente realisiert werden. Bezüglich der vermittelten Werte arbeitet die Vermittlungskomponente als reiner Durchreicher. Beispielsweise realisiert ein Telefon mit Anschluss an eine zentrale Vermittlungsstelle einen Kanal, der nacheinander für Gespräche mit unterschiedlichen Personen genutzt wird. Dieses Konzept wird immer dann angewandt, wenn die Realisierung getrennter Kanäle zu aufwendig oder auch sinnlos wäre. Wäre dies nicht so, müsste man ein Telefon für jede Person haben, mit der man irgendwann reden wollte. Dies wäre nicht nur extrem teuer, es wäre auch unübersichtlich.

Bei der Gestaltung von Benutzungsschnittstellen wird dieses Konzept daher gleichermaßen zur Kostenreduktion und Wahrung der Übersichtlichkeit eingesetzt. Als eigenständiges Konzept neben Interaktionsadaptern und Assistenten bildet der Kanalmultiplex den dritten und letzten Baustein in der Gestaltung von Benutzungsschnittstellen. Unterschiedliche Kanäle, die innerhalb eines zweckbezogenen Modells identifiziert werden können, werden im interaktionsbezogenen Modell, wie Abbildung 3.21 zeigt, auf eine einzige Interaktionskomponente abgebildet. Prinzipiell sind dabei zwei Fälle zu unterscheiden:

- Im Falle des sogenannten Zeitmultiplex werden die Folgen einzelner Werte, die im zweckbezogenen Modell unterschiedlichen Kanälen zugeordnet sind, nacheinander oder abschnittsweise abwechselnd über eine einzige Komponente kommuniziert, welche die jeweiligen zweckbezogenen Kanäle realisiert.
- Im Falle des sogenannten Ortsmultiplex werden die Folgen einzelner Werte, die im zweckbezogenen Modell unterschiedlichen Kanälen zugeordnet sind, zeitgleich über unterschiedliche Schnittstellenmerkmale einer einzigen Komponente kommuniziert, welche die jeweiligen zweckbezogenen Kanäle realisiert. Als Beispiel hierfür kann die zeitgleiche Ausgabe verschiedener Komponenten eines interaktiven Systems auf einem Bildschirm gedeutet werden.

Darüber hinaus gilt: Sollen mehrere Ausgabekanäle auf diese Weise realisiert werden, ist eine Komponente zur Kanalbündelung erforderlich (Multiplex). Im Eingabefall muss es eine zuständige Komponente für die Nachrichtenverteilung (Demultiplex) geben. Beim Ortsmultiplex ist ein zweckbezogener Kanal fest bestimmten Merkmalen der realisierenden Interaktionskomponente zugeordnet. Im Falle des Zeitmultiplex gibt es unterschiedliche Verfahren zu entscheiden, welcher zweckbezogene Kanal zu einem bestimmten Zeitpunkt realisiert ist. Ein vorgegebenes Zeitraster zum Wechsel der Kanäle ist für interaktive Systeme kaum praktikabel. Bei Eingaben über Kanäle, die im Zeitmultiplex genutzt werden, findet daher fast immer eine explizite Kanalwahl durch den Benutzer statt. Zu diesem Zweck besitzt der Nachrichtenverteiler eine separate Schnittstelle, über die der gewünschte Kanal identifiziert wird. Ein solcher Kanal wird beispielsweise am Tastenfeld eines Telefons sichtbar. In gleicher Weise kann die

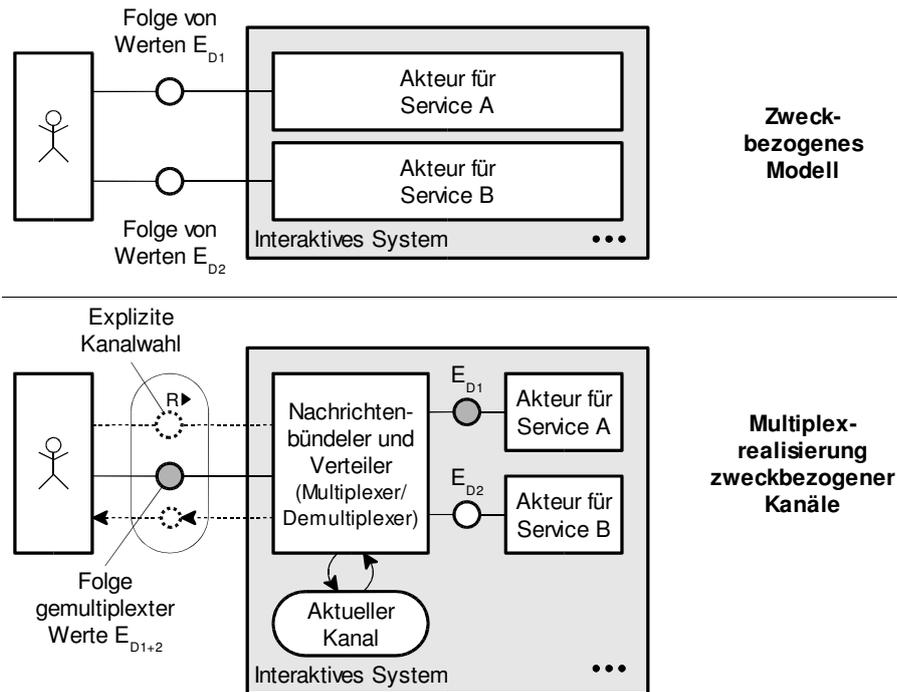


Abbildung 3.21: Multiplexrealisierung von Kanälen

Funktion einer besonderen Taste auf vielen Computertastaturen („Alt“) gedeutet werden, durch deren Betätigung man die zentrale Menüauswahl vieler Anwendungen aktivieren kann. Je nach Situation kann ein Benutzer hiermit die Vorstellung eines Kanalaufbaus verbinden oder die Erregung der Aufmerksamkeit seines Gesprächspartners. Ist das Repertoire aller adressierten Akteure paarweise disjunkt, so ist eine implizite Identifikation des Kanals möglich.

Abbildung 3.22 zeigt den Ausschnitt eines interaktionsbezogenes Modells des bekannten Autorensystems, in dem beide Konzepte zur Anwendung kommen. Eine spezielle Menüwahltaste aktiviert die Menüsteuerung. Bis zur Auswahl eines Menüpunktes werden alle Tastatureingaben an die Menüsteuerung weitergeleitet, danach gibt die Menüsteuerung den Kanal wieder ab. Findet keine Menüauswahl statt, sollen alle Tastatureingaben mit Ausnahme zweier Tasten zum Blättern („Bild-Hoch“ und „Bild-Runter“) an den Zeichenfolgeneditor durchgereicht werden. Bei Ausgaben kann anstelle einer Kanalwahl durch den Benutzer eine willkürliche Sequentialisierung durch einen Multiplexer zweckmäßig sein. So könnten auf einem Bildschirm nacheinander Nachrichten unterschiedlicher Systemkomponenten ausgegeben werden.

Während Zeitmultiplex durch einen Benutzer in den meisten Fällen bewusst erlebt wird, ist das Erleben von Ortsmultiplex von einer bewussten Wahl der Betrachtungsebene abhängig. Es obliegt dem Benutzer, ob er den Bildschirm als einen Kanal zum System oder n Kanäle zu unterschiedlichen Interaktionsadaptern sehen möchte. Welche Vorstellung sich dabei entwickelt, wird wesentlich durch die Form der Bildschirmdarstellung, aber auch durch die Dokumentation des Systems und durch die Unterstützung durch Hilfesysteme beeinflusst. Als ein Sonderfall der Anwendung des Ortsmultiplexprinzips kann das partielle Überlagern der Ausgabe von Interaktionsas-

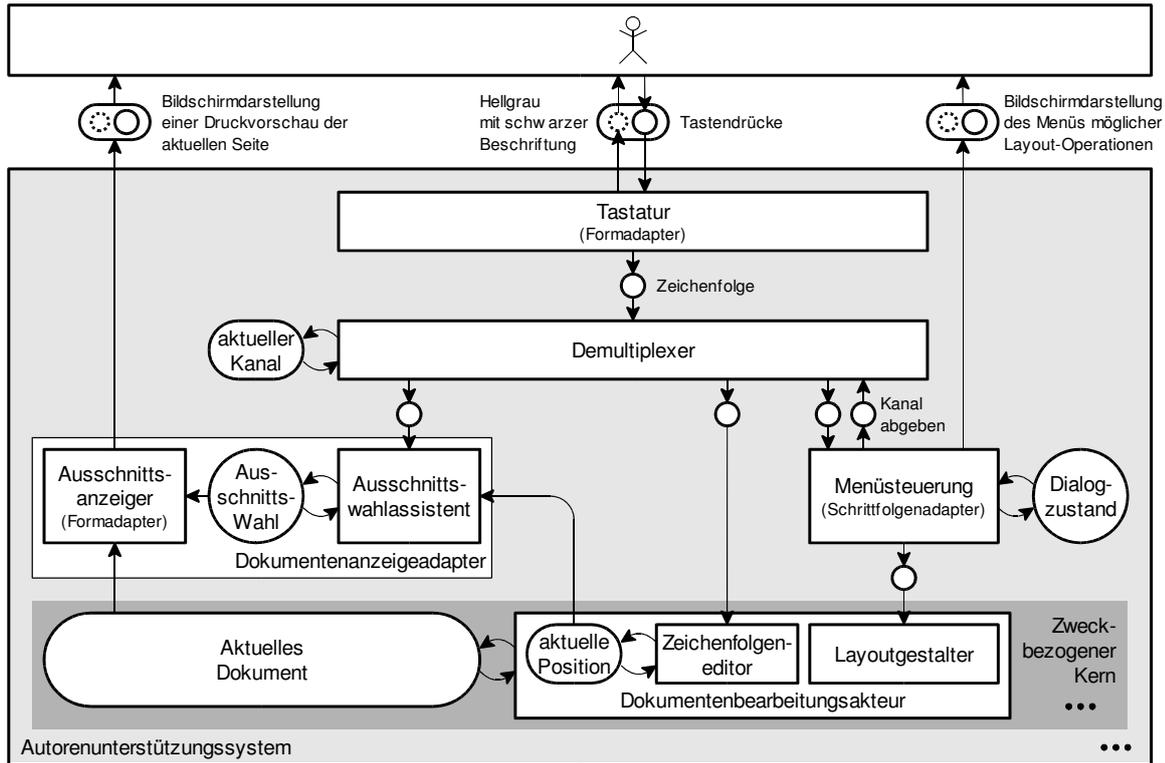


Abbildung 3.22: Autorensystem — Ausschnitt des interaktionsbezogenen Modells

sistenten und sonstigen Ausgaben angesehen werden. Als Beispiel hierzu sind durchscheinende Tooltips oder auch Anwendungen im Bereich der sogenannten Augmented Reality [Azu97] denkbar.

3.5 Realisierungsbezogene Modelle

3.5.1 Zweck und Charakterisierung der Modelle

Ein interaktionsvollständiges Modell beschreibt ein Interaktionssystem und damit ein interaktives System vollständig im Hinblick auf das gewollte Verhalten an der Benutzungsschnittstelle. Interne Realisierungsstrukturen werden auf der Ebene interaktionsbezogener Modelle nicht festgelegt. Praktisch sind fast alle dieser Modelle als Ersatzmodelle anzusehen. Zur Schaffung eines interaktiven Systems müssen zusätzliche Entwurfsentscheidungen getroffen werden. Das Ziel der damit verbundenen Modelltransformationen besteht darin, schrittweise eine Abbildung auf tiefere Betrachtungsebenen vorzunehmen, auf der das interaktive System als Struktur verschalteter und gegebenenfalls programmierter Systemkomponenten, die zu einem Repertoire gegenständlicher Fertigungsbausteine gehören [Gal98], angesehen werden kann. Mehrere Betrachtungsebenen ergeben sich aus der Verwendung programmierbarer Komponenten als Trägersystem zur Realisierung übergeordneter Rollenmodelle. Nur auf unterster Ebene einer solchen Trägersystemsichtung besteht eine Isomorphie zwischen

der Aufbaustruktur der entsprechenden Modelle und den gegenständlich erlebbaren internen Aufbaustrukturen interaktiver Systeme.

Viele interaktive Systeme besitzen bezüglich der Bedienkonzepte eine ähnliche Benutzungsschnittstelle. So ist die Bedienung mittels Maus und Tastatur bei einer gleichzeitigen Darstellung der Ausgaben über einen Bildschirm ein häufig angewandtes Interaktionsmuster. Die Menge zu lösender Realisierungsprobleme, sowie das Repertoire möglicher Entwurfsentscheidungen ist für solche Systeme weitgehend ähnlich. Daraus resultieren Bestrebungen, interaktive Systeme ausgehend von einem interaktionsvollständigen Modell weitgehend schematisch — idealerweise automatisch — abzuleiten. Das Ideal wäre die Schaffung anwendungsunabhängiger Abwickler zur direkten Ausführung beliebiger interaktionsvollständiger Modelle. Eine völlige Anwendungsunabhängigkeit solcher Abwickler ist aufgrund frei gestaltbarer Benutzungsschnittstellen jedoch praktisch unmöglich. Ein solcher Abwickler würde die unmittelbare Erzeugbarkeit beliebiger Interaktionsadapter mit beliebiger materiell-energetischer Erscheinung erfordern.

In der Praxis ist es üblich, zur Realisierung ähnlicher Systeme den Typ einer Realisierungsstruktur festzulegen, von der anwendungsspezifische Ausprägungen abgeleitet werden. Ein solcher Strukturtyp legt das Repertoire möglicher bzw. erforderlicher Systemkomponenten, ihre Schnittstellen und Zuständigkeiten, sowie die Art der Kopplung zwischen den einzelnen Komponenten fest. Das Realisierungsmodell eines interaktiven Systems kann als Ergebnis einer typkonformen anwendungsspezifischen Abbildung eines interaktionsvollständigen Modells angesehen werden. Häufig beschreibt ein solches Modell ein Rollensystem. Auf tieferer Betrachtungsebene bildet ein solches Modell dann die Grundlage für das Programm eines Trägersystems. Eine formale Beschreibung des Rollensystems, die häufig mit einer Definition bzw. Erweiterung des Repertoires anwendungsspezifischer Realisierungskomponenten verbunden ist, bildet das Programm. Im Umfeld des Software Engineering wird die zugrundeliegende anwendungsunabhängige Systemstruktur, die häufig mit einer typischen Beschreibungsstruktur verbunden ist, vielfach als die *Architektur*⁴ des Systems bezeichnet. Mit einem bestimmten Architekturtyp ist meist eine anwendungsunabhängige charakteristische Begriffswelt verbunden, die zur Beschreibung der Modelle dient und die ein erkennbares Zeichen der Ebene realisierungsbezogener Modelle ist. Eine Auswahl bekannter Architekturtypen (Architekturstile in der Terminologie nach [BCK98]) wird in Kapitel 4.3 vorgestellt. Häufig kann zwischen einem Realisierungsmodell der Anwendungsentwickler, das einen möglichst direkt Bezug zum interaktionsvollständigen Modell erkennen lässt, und nachgelagerten Trägersystemmodellen der Plattformentwickler unterschieden werden. Jede dieser Betrachtungsebenen besitzt im allgemeinen eine spezifische Architektur.

Entwurfsentscheidungen auf der Ebene realisierungsbezogener Modelle sind besonders geprägt vom Bewusstsein für eine ressourcensparende Realisierung. Durch die Verwendung geeigneter verfügbarer Komponenten und Konzepte, lassen sich Entwicklungszeit und -kosten einsparen. Durch geschickte Aufgabenverteilung und Fin-

⁴Der Begriff Architektur ist im Bereich des Software-Engineering nicht eindeutig definiert (vgl. Abschnitt 3.2, über die Rolle des Systemarchitekten). In der einschlägigen Literatur zum Thema interaktiver Systeme [BC92, Sta96] wird der Begriff jedoch zumeist in seiner obigen Bedeutung verwendet.

dung optimaler Algorithmen zur Lösung zweck- und interaktionsmodellbedingter Aufgaben kann das interaktive Systems hinsichtlich erforderlicher Betriebsmittel (wie zum Beispiel Speicherbedarf und Rechenleistung) optimiert oder für den nachträglichen Austausch von Komponenten konzipiert werden. Komplexe Systeme lassen sich innerhalb einer vertretbaren Zeitspanne nur arbeitsteilig entwickeln. Ermöglicht wird Arbeitsteilung durch die Abgrenzung getrennt entwickelbarer Aufbaukomponenten und Beschreibungseinheiten aber auch durch aufeinander aufbauende Betrachtungsebenen. Letzteres erlaubt beispielsweise die Entwicklung zweck- und interaktionsbezogener Modelle unabhängig von der Entwicklung einer geeigneten Abwicklerplattform.

Je komplexer ein Modell ist, umso komplexer ist dessen Beschreibung. Geschickte Beschreibungsstrukturen sind deshalb eine Voraussetzung für die arbeitsteilige Erstellung umfangreicher Systembeschreibungen auf jeder Betrachtungsebene. Die getrennte Beschreibung unabhängig betrachtbarer Entwurfsentscheidungen [Par72] kann sich nicht nur in der getrennten Beschreibung einzelner Systemkomponenten äußern, sondern auch in der herausgelösten Beschreibung identifizierbarer Typen, die der Erhöhung der Semantik einer verwendeten Sprache dient. Da zumeist erst auf Ebene realisierungsbezogener Modelle alle Entwurfsentscheidungen zum Erreichen einer Fertigstellungsvollständigkeit getroffen werden, sind die Beschreibungen auf dieser Modellebene vergleichsweise umfangreich. Die Strukturierung der Beschreibung prägt daher auch die Gestaltung geeigneter Abwicklerplattformen. Trotz des starken Gewichtes von Beschreibungsstrukturen ist es jedoch für ein Systemverständnis wichtig, sich des Unterschiedes zwischen Beschreibungs- und Systemstrukturen bewusst zu bleiben.

3.5.2 Konzepte der Realisierung interaktiver Systeme

Die Realisierung eines interaktionsbezogenen Modelles wird bei unterschiedlichen Randbedingungen zu unterschiedlichen Realisierungsmodellen führen. Dennoch gibt es bestimmte Grundmuster, die sich System übergreifend als zweckmäßig erwiesen haben und die den verschiedenen Realisierungsmodellen zugrundeliegen.

3.5.2.1 Schnittstellengeräte und informationeller Kern

Auf der Ebene interaktionsbezogener Modelle wird die materiell-energetische Form einer Benutzungsschnittstelle durch das Vorhandensein von Interaktionsadaptern bestimmt, die zwischen Benutzer und zweckbezogenem Kern vermitteln. Bei der Realisierung interaktiver Systeme hat es sich als zweckmäßig erwiesen, eine ähnliche Trennung in einen *informationellen Kern* und *Schnittstellengeräte* mit Aktoren und Sensoren vorzunehmen. Die Realisierung des informationellen Kerns erfolgt häufig auf der Basis programmierbarer, anwendungsunabhängiger Trägersysteme, die von unterschiedlichen Herstellern in großer Stückzahl und damit kostengünstig angeboten werden. In gleicher Weise gibt es Hersteller, die sich auf die Herstellung von Schnittstellengeräten spezialisiert haben, die gleichermaßen für unterschiedliche Anwendungsbereiche eingesetzt werden können. Daher ist mit der Entwicklung neuer interaktiver

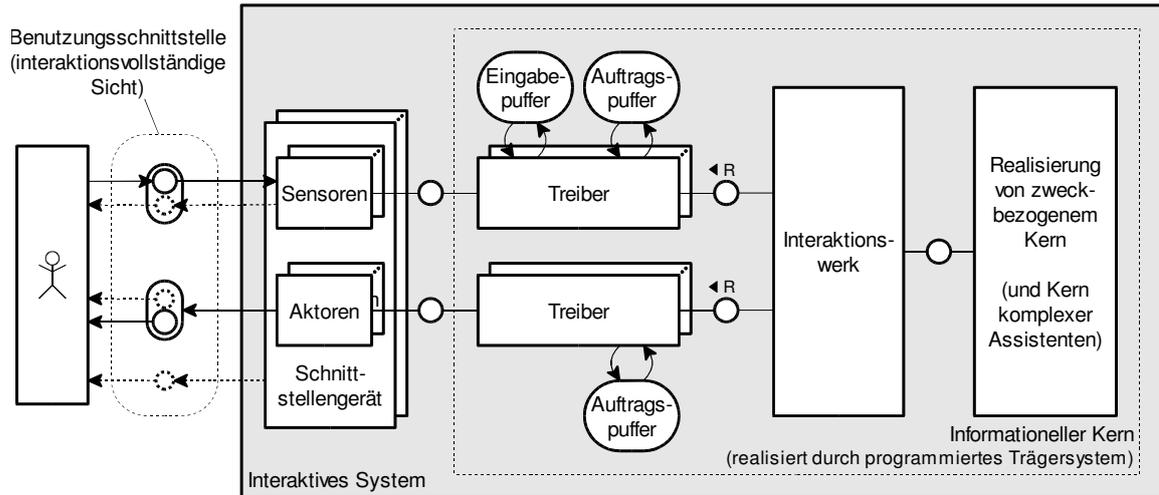


Abbildung 3.23: Basismodell der Realisierung interaktiver Systeme

Systeme nur im Ausnahmefall die vergleichsweise teure Entwicklung neuer Schnittstellengeräte verbunden.

Schnittstellengeräte bestimmen wesentlich die inhärente Form einer Benutzungsschnittstelle. Aus Sicht eines Benutzers ergibt sich in vielen Fällen eine scheinbare Identität zwischen Schnittstellengeräten mit Aktoren und Sensoren und Interaktionsadaptern mit Bedienelementen. Tatsächlich gibt es insbesondere für Eingabegeräte häufig eine Eins-zu-Eins-Beziehung zu primitiven Formadaptern. Typische Beispiele sind Tastatur und Maus. Schnittstellengeräte zur Grafikausgabe werden hingegen häufig im Ortsmultiplex zur Realisierung mehrerer zeitgleich wahrnehmbarer Interaktionsadapter eingesetzt. Vereinfachend soll die Komposition aus Grafikkarte und Monitor in diesem Zusammenhang als *ein* Schnittstellengerät betrachtet werden. Die inhärente Form anwendungsspezifischer Formadapter entsteht in diesen Fällen, ebenso wie die Form der zu kommunizierenden Informationen, erst durch entsprechend formulierte Aufträge des informationellen Kerns an das Schnittstellengerät. Im allgemeinen ist die Funktionalität von Schnittstellengeräten auf die Erzeugung bzw. Erfassung von elementaren Formen ausgelegt. Beispiele sind die Fähigkeit zur Erzeugung von Punkten oder Linien oder die Fähigkeit zur Erkennung von Tastendrücken. Schnittstellengeräte stellen in der Mehrzahl der Fälle anwendungsunabhängig einsetzbare Formwandler dar. Sind komplexere, anwendungsspezifische Formen erforderlich, müssen sie auf diese Elementarformen zurückgeführt werden.

Es hat sich als zweckmäßig erwiesen, innerhalb des informationellen Kerns neben dem zweckbezogenen Kern als eigenständiger Komponente zwei weitere Komponenten oder Schichten abzugrenzen. Wie Abbildung 3.23 zeigt, wird die Schicht der Interaktionsadapter und Multiplexer durch drei aneinander angrenzende Schichten von Systemkomponenten realisiert: Die Schicht der Schnittstellengeräte, die Schicht der *Treiber* und eine dritte Schicht, die hier als *Interaktionswerk* bezeichnet ist. An der Realisierung eines zweckbezogenen Kanals sind stets alle drei Schichten beteiligt.

Aufgabe eines Treibers ist die Entkopplung zwischen einem Actor oder Sensor und

der Komponente zur Ansteuerung bzw. Auswertung. Zum einen kapselt ein Treiber durch Bereitstellung einer genormten Auftragschnittstelle das Wissen um das häufig herstellerspezifische Protokoll zur Kommunikation mit Aktoren bzw. Sensoren eines bestimmten Typs. Zum anderen erlauben Treiber im allgemeinen eine asynchrone Auftragsannahme und -ausführung, so dass Ein- und Ausgaben nebenläufig zueinander und zu sonstigen Aktivitäten des informationellen Kerns erfolgen können. Die Normung der Treiberschnittstellen und die Austauschbarkeit von Treibern ermöglichen unabhängig vom Rest des Systems die Austauschbarkeit der Schnittstellengeräte. Durch entsprechenden Aufträge des Interaktionswerkes an die Treiber zur Erzeugung bzw. Erfassung elementarer Formen findet die Abbildung auf zweckbezogener Ein- und Ausgabewerte statt.

3.5.2.2 Offene Entwurfsentscheidungen

Das vorgestellte Modell lässt viele Entwurfsentscheidungen offen. Die folgende Übersicht erläutert wichtige Aspekte, die bei jeder Realisierung berücksichtigt werden müssen.

Die innere Struktur der identifizierten Komponenten ist weitgehend offen. Nach [BC92] lassen sich zwei Ansätze unterscheiden. Beim monolithischen Ansatz wird jede Ebene als eine aktive gegebenenfalls mehrstufige Komponente angesehen, so dass sich eine Linearordnung von Systemkomponenten ergibt. Ein Beispiel hierfür ist das „Referenzmodell für die graphische Datenverarbeitung“ (Computer Graphics Reference Model, CGRM) [DIN93]. Im Gegensatz zum monolithischen Ansatz besteht bei einem sogenannten „Multiagent Model“ jede Ebene aus einer Vielzahl von Komponenten unterschiedlichen Typs, die untereinander kommunizieren können. Im allgemeinen erleichtert dies den Bezug zu den Komponenten interaktionsbezogener Modelle. Interaktionsadapter und deren Interaktionselemente werden dazu auf eine oder mehrere Komponenten des Interaktionswerkes abgebildet, welche dann das entsprechende Schnittstellenverhalten unter Verwendung der Schnittstellengeräte realisieren. So ergeben sich zur Realisierung grafischer Benutzungsschnittstellen mit hierarchischen Strukturen aus Interaktionsadaptern und -elementen häufig korrespondierende Hierarchien aus Realisierungskomponenten. Eine Differenzierung zwischen Interaktionsadaptern und Interaktionselementen findet auf dieser Modellebene in der Regel nicht mehr statt. Bekannte Vertreter dieses Ansatzes sind *Model-View-Controller (MVC)* und *Presentation-Application-Control (PAC)*. Beispielsweise enthält das Interaktionswerk bei einem gemäß MVC strukturiertem Systemmodell eine Hierarchie von View-Controller-Paaren. Beide Ansätze werden in Kapitel 4.3 beschrieben.

Abhängig von Zweck und Interaktionsmodell hat man die Vorstellung, dass der Interaktionsprozess zwischen Benutzer und einem interaktiven System entweder vom Benutzer oder vom interaktiven System getrieben wird. Beispielsweise geht bei einem Zeicheneditor die Aktivität im allgemeinen vom Benutzer aus. Das System erkennt Eingaben, modifiziert entsprechend die Zeichnung und kehrt danach in einen unspezifischen Wartezustand zurück. Die Steuerung liegt beim Benutzer, der nach Belieben den operationellen Zustand des Systems verändern kann. Im entgegengesetzten Fall wird der Ablauf durch das interaktive System vorgegeben. Unabhängig davon

stellt sich die Frage, welche Komponente innerhalb eines interaktiven Systems dafür verantwortlich ist, den Interaktionsprozess voranzutreiben. Aufgrund der Entkopplung durch die Treiberschicht muss entweder das Interaktionswerk oder der zweckbezogene Kern die Rolle eines Steuerwerkes (vgl. [Wen70]) übernehmen. Je nachdem spricht man von externer oder interner Kontrolle [Str85]. In heutigen Systemen wird überwiegend das Prinzip der externen Kontrolle angewendet. Der zweckbezogene Kern wird bei dieser Sichtweise zur operationellen Komponente.

Ein Realisierungsmodell legt neben den erforderlichen Systemkomponenten und ihren Zuständigkeiten, die Art der Kopplung, das Protokoll, zwischen den Komponenten und die Form bzw. die Codierung der systeminternen Informationen fest. Es wird festgelegt, wie und zu welchen Zeitpunkten, welche Komponenten welche Informationen austauschen. Dabei spielen Überlegungen, wie die Erkennung bzw. Erzeugung von Formen mittels der Schnittstellengeräte effizient gestaltet werden kann, eine große Rolle. So lässt sich die Tatsache, dass bei vielen interaktiven Systemen nicht in jedem Dialogzustand alle Komponenten und Interaktionsadapter aktiv bzw. wahrnehmbar sind.

Bei interaktiven Systemen mit grafischer Benutzungsschnittstelle soll häufig die Änderung eines darzustellenden informationellen Wertes unmittelbar sichtbar sein. Diese Anforderung und der zur Umsetzung erforderliche Aufwand ist sinnlos, wenn diese Änderungen schneller erfolgen, als ein Mensch sie wahrnehmen oder ein Schnittstellengerät sie darstellen könnte.

Großer Aufwand ist im allgemeinen auch mit der Darstellung umfangreicher strukturierter Informationen verbunden. Werden nur einzelne Komponenten einer solchen Struktur geändert, ändert sich an der Darstellung häufig nur wenig, oder bei einer Ausschnittsdarstellung manchmal gar nichts. Auf diesen und ähnlichen Überlegungen basieren Pufferungs- und Synchronisationskonzepte. Von zentraler Bedeutung ist dabei die Trennung von Erfassung und Behandlung von Ereignismeldungen über erfolgte Eingaben und Datenänderungen.

Im Falle von „Multiagent Modellen“ ist wegen der Komponentenkopplung die Frage, inwieweit die Aufbaustruktur des Systems, nach einer anfänglichen Inbetriebnahme, während des Betriebes unverändert fortbesteht oder ob während des Betriebes Komponenten ausgetauscht, hinzugefügt oder entfernt werden können, zu beantworten. So mag es auf Ebene eines realisierungsbezogenen Modell zweckmäßig sein, eine Komponente zur Realisierung eines nur zeitweise wahrnehmbarem Interaktionsadapters erst im Bedarfsfalle zu erzeugen und nach der Nutzung wieder zu entfernen. Hierzu ist zu klären, wie Komponenten ihre Kommunikationspartner oder gemeinsam genutzte Speicher identifizieren. Typischerweise erhalten die Systemkomponenten das erforderliche Wissen erst im Rahmen eines Initialisierungsschrittes nach ihrer Erzeugung durch eine übergeordnete Instanz. Dies ermöglicht eine von einander weitgehend unabhängige und damit arbeitsteilige Beschreibung der verschiedenen Typen von Komponenten. So wird bei einem gemäß MVC strukturiertem Systemmodell eine Modell-Komponente erst durch Registrierung während des Betriebes mit der entsprechenden View-Komponente verkoppelt.

3.5.2.3 Trägersystemmodell und Entwicklungsumgebung

Der informationelle Kern eines interaktiven Systems ist im allgemeinen durch Programmierung eines Abwicklers für Beschreibungen interaktiver Systeme realisiert. Dabei ist die in Abbildung 3.24 gezeigte Vorstellung angemessen, nach der einem Abwickler die Beschreibung von Interaktionswerk und Kernrealisierung eines realisierungsbezogenen Rollenmodells übergeben wird. Eine solche Beschreibung enthält im allgemeinen nicht nur Programm-Module, sondern zum Beispiel auch Bild- und Tondateien sowie Parametrisierungsinformationen. Die Wahl geeigneter Beschreibungsstrukturen erlaubt die isolierte Handhabbarkeit von Beschreibungskomponenten und so deren arbeitsteilige Erstellung oder Änderung. Die Erstellung solcher Beschreibungen wird durch entsprechende Entwicklungswerkzeuge ermöglicht, die selbst nicht als Bestandteil des zu schaffenden Systems anzusehen sind. Zur Unterstützung des Entwicklungsprozesses sind die benötigten Werkzeuge häufig in eine sogenannte Entwicklungsumgebung integriert, die sowohl unterstützend als auch synchronisierend bei den verschiedenen Entwicklungsaktivitäten wirken kann.

Idealerweise benötigt ein Anwendungs- oder Benutzungsschnittstellenentwickler wenig Wissen um die Struktur des zugrundeliegenden Trägersystems. In der Praxis sind Form und Inhalt eines Programms häufig nur vor dem Hintergrund eines geeigneten Trägersystemmodells verständlich. Dazu gehört das Wissen, wie welche Beschreibungskomponenten ausgewertet werden und wo sie wann verfügbar sein müssen. Bei einer explizite Bezugnahme in Programmen auf Strukturen des Trägersystems, wie dies beispielsweise zur Programmierung verteilter Netzwerkkomponenten erforderlich sein kann, liegt eine Vermischung der Beschreibung von Rollen- und Abwicklermodell vor. Eine Entkopplung lässt sich nur durch die Bereitstellung von Abwicklern bzw. Übersetzern für realisierungsunabhängige Beschreibungsformen erreichen. Moderne Entwicklungsumgebungen versuchen so Schritt für Schritt die Annäherung von interaktionsbezogenen Modellen und realisierungsbezogenen Rollenmodellen voranzutreiben.

3.5.3 Einfluss auf die Benutzungsschnittstelle

Wenn man den bekannten Ansätzen zur Entwicklung interaktiver Systeme folgt, dann sollten Entwurfsentscheidungen auf der Ebene der realisierungsbezogenen Modelle idealerweise keinen Einfluss auf die Benutzungsschnittstelle besitzen. Praktisch ist dies nicht immer zu vermeiden. Aus diesem Grunde muss die Erscheinung bzw. das Verhalten des interaktiven Systems im Normalbetrieb von seinem Fehl- bzw. Ausnahmeverhalten unterschieden werden.

Die Auswahl von Komponenten aus einem zur Realisierung vorgegebenen, verfügbaren Repertoire prägt häufig — wenn auch nur im Detail — implizit die Form der Benutzungsschnittstelle. Dieser Weg wird häufig aus Kostengründen beschritten, wenn eine präzise Umsetzung der spezifizierten Benutzungsschnittstelle zu aufwendig ist. In vielen Fällen werden Details der Benutzungsschnittstelle auch bewusst bis zur Realisierung offen gelassen. Im Falle grafischer Benutzungsschnittstellen ergibt

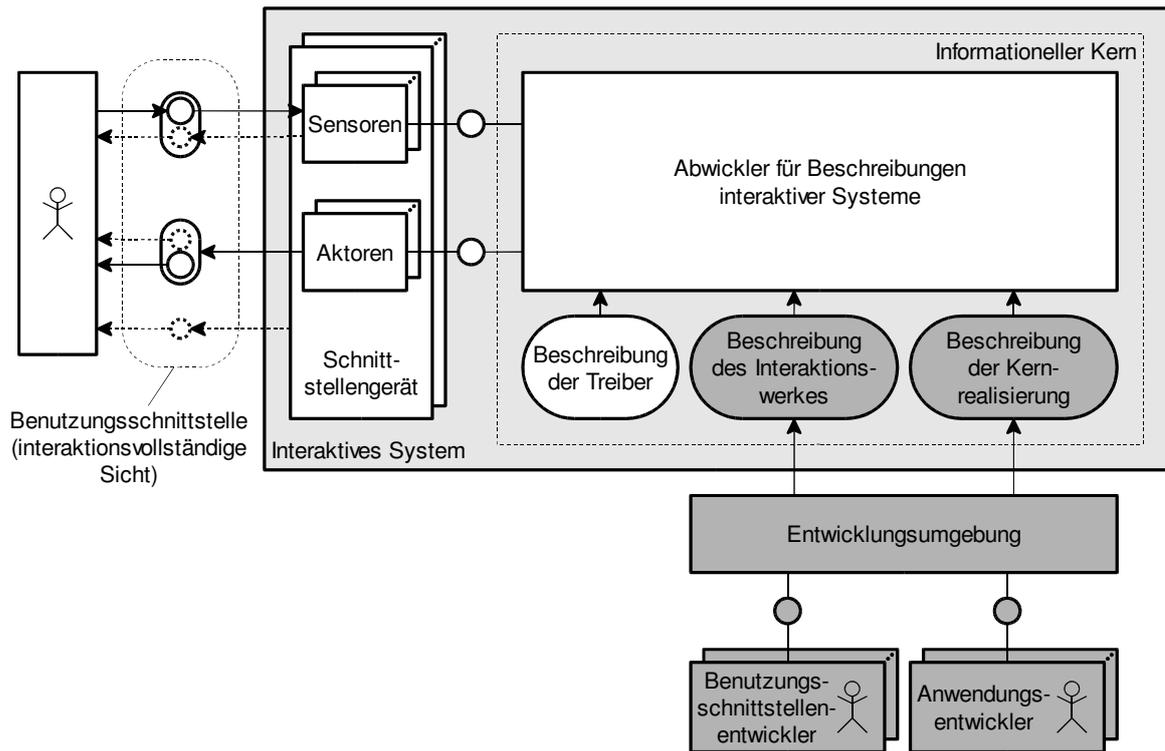


Abbildung 3.24: Trägersystemmodell und Entwicklungsumgebung

sich so das endgültige Aussehen meist implizit durch die Verwendung eines sogenannten „GUI-Frameworks“⁵. Ein solches Framework bildet ein vorgefertigtes anwendungsunabhängiges Programm-Gerüst zur Beschreibung von Interaktionswerk (und häufig auch Kern), das die Architektur des Systems und das Erscheinungsbild der elementaren Interaktionselemente prägt. In Unkenntnis des Entwicklungsprozesses könnte jemand die resultierende Form der Benutzerschnittstelle dann als das Ergebnis einer von der Realisierung unabhängigen Schnittstellengestaltung ansehen. Dies gilt allerdings nur für den Normalbetrieb des Systems.

Zeigt ein interaktives System ein Ausnahme- oder Fehlverhalten, so kann dies nur aufgrund des Wissens um die zugrundeliegende Realisierung verstanden werden. Als Fehlverhalten eines interaktiven Systems soll an dieser Stelle jedes unvorhersehbare Systemverhalten bezeichnet werden, welches gegen den Systemzweck gerichtet ist und nicht auf fehlerhafte Benutzung zurückzuführen ist. Häufig ist Ursache für ein solches Verhalten ein konstruktiver Fehler. Kann eine Fehlersituation durch das interaktive System selbst erkannt werden, sollte idealerweise durch konstruktive Maßnahmen der Übergang in einen unkritischen Systemzustand sichergestellt und der Benutzer entsprechend informiert werden.

Das Ausnahmeverhalten eines Systems unterscheidet sich für einen Benutzer vom Fehlverhalten dadurch, dass es zum bestimmungsgemäßen Betrieb des Systems gehört, auch wenn es nicht oder nur indirekt dem Zweck dienlich ist. Als Beispiel kann die Ausgabe einer Warnmeldung beim Unterschreiten des Ladezustandes eines akku-

⁵GUI = graphical user interface.

betriebeben System gelten. Die Entscheidung, einen Akkumulator zur Energieversorgung des Systems zu verwenden, hat weder unmittelbar etwas mit dem Systemzweck noch mit dem gewählten Interaktionsmodell zu tun. Dennoch muss sich ein Benutzer dieser Tatsache und den damit verbundenen Konsequenzen bewusst sein. Ausnahmeverhalten wird zum Fehlverhalten, wenn das System dadurch seinen Zweck nicht erfüllen kann. Ein Beispiel hierfür ist der Übergang von einer fließenden zu einer ruckhaften Darstellung bei der Präsentation komplexer Grafiken aufgrund mangelnder Rechenleistung. In vielen Anwendungsbereichen ist dies ein zulässiges Ausnahmeverhalten. Bei einem System zur Durchführung bildschirmgestützter Gehirnoperationen wäre dies ein inakzeptables Fehlverhalten.

Kapitel 4

Beschreibungsansätze in der Literatur

In diesem Kapitel wird eine repräsentative Auswahl von Konzepten zur Beschreibung bzw. fertigen Beschreibungen interaktiver Systeme aus der einschlägigen Literatur daraufhin untersucht, in welchem Umfang durch sie die vorgestellte Vorstellungswelt interaktiver Systeme zum Ausdruck kommt. Für jeden Fall wird zunächst mit Bezug auf Abbildung 3.2 Gegenstand und Zweck der Beschreibung identifiziert. Danach werden die im Rahmen einer solchen Beschreibung vermittelten Sachverhalte, sowie die dabei verwendete Form vorgestellt. Letztlich wird untersucht, welche Art von Systemvorstellung durch die jeweilige Beschreibung vermittelt wird und wie der Bezug zu benachbarten Modellebenen hergestellt wird. Konkrete Beispiele, ergänzt um eine FMC-basierte Darstellung, sollen dazu dienen, die Aussagen zu untermauern bzw. zu veranschaulichen.

Eine Übersicht der untersuchten Beschreibungsansätze ist in Tabelle 4.1 gezeigt. Die Auswahl der untersuchten Beschreibungsansätze basiert auf folgenden Überlegungen: Es existiert eine kaum überschaubare Fülle unterschiedlicher Ansätze zur Entwicklung interaktiver Systeme¹. Dennoch basieren die meisten auf einer recht überschaubaren Anzahl zugrundeliegender Modellfindungs- und Beschreibungskonzepte. Die hier getroffene Auswahl soll das Spektrum dieser Konzepte ebenso wie die drei vorgestellten Modellebenen abdecken. Dabei wurden insbesondere Ansätze berücksichtigt, denen aufgrund ihres Bekanntheitsgrades oder ihrer Relevanz für den Entwicklungsprozess in der Praxis eine besondere Bedeutung zukommt.

Insgesamt werden neun Fälle betrachtet: Die ersten zwei [Par69, Shn82] sind allgemeine Ansätze zur Beschreibung des Verhaltens interaktiver Systeme an der Benutzungsschnittstelle. Vier weitere Ansätze [SM91, Joh92, Sta96, KAB01] definieren Konzepte zur Beschreibung zweck- und interaktionsbezogener Modelle im Rahmen eines Entwicklungsprozessmodells. Zuletzt wird die Beschreibung von drei unterschiedlichen Architekturstilen, MVC [KP88], PAC [Cou87] und Seeheim [Gre85], durch die einschlägige Literatur unter die Lupe genommen.

¹siehe [Shn98]: „For the moment, the field is filled with hundreds of theories competing for attention [...]“

	Parnas, 1969 <i>Use of Transitions- digrams</i>	Interaktions- bezogenes Modell eines interaktiven Systems	Evaluation alternativer Interaktionsmodelle	Verhalten an der Benutzungsschnittstelle	Zustands- diagramme	Trivialaufbau (mit Bezug auf mögliche Realisierung)
Gegenstand der Beschreibung	Shneiderman, 1982 <i>Multiparty grammars</i>	Zweck- und interaktions- bezogenes Modell eines Interaktions- systems	Automatisierte Auswertbarkeit (Programm- generierung, Tests)	Verhalten an der Benutzungsschnittstelle	Grammatiken in Backus-Naur-Form	Trivialaufbau (mit Bezug auf mögliche Realisierung)
Zweck der Beschreibung	Sutcliffe, 1991 <i>Integrating HCI design with SA</i>	Zweck-, interaktions- und realierungs- bezogenes Modell eines Interaktions- systems	Prototyp-Evaluation, Software- entwicklung	Datenfluss- und Kausalstruktur Verhalten und Form der Benutzungsschnittstelle	Data Flow Diagrams, Structured English, Zustands- diagramme, Structure charts, Bildschirm- skizzen	Vorstellung kooperierender Komponenten <i>kann</i> in DFDs zum Ausdruck kommen
Vermittelte Sachverhalte	Johnson, 1992 <i>Application of Task analysis to HCI design</i>	Zweck- und interaktions- bezogenes Modell eines interaktiven Systems	Prototyp-Evaluation, Software- Entwicklung	Primär hierarchische Aufgabenstruktur Atomare Objekt- und Aktionstypen Verhalten und Form der Benutzungsschnittstelle	Text, Aufgaben- diagramme, Klassifikationen Structured English, Bildschirm- skizzen (sonstiges ?)	Vorstellung kooperierender Komponenten <i>kann</i> zum Ausdruck kommen
	Stary, 1996 <i>Modell- basiertes Design</i>	Zweck- und interaktions- bezogenes Modell eines interaktiven Systems	Software- Entwicklung, -generierung	Klassifikationen von Aufgaben und Daten Zuordnung zu Interaktions- elementtypen Verhalten an der Benutzungsschnittstelle	Klassendiagramme, State charts, freie Notation	Vorstellung kooperierender Komponenten kommt nicht zum Ausdruck
	Kruchten, 2001 <i>iRUP</i>	Interaktions- bezogenes Modell eines Interaktions- systems	Prototyp-Evaluation, Software- entwicklung	Benutzbarkeits- anforderungen zu den gegebenen Use Cases Verhalten und Form der Benutzungsschnittstelle Identifikation von Boundary Objects	Storyboard: Text, Bildschirm- skizzen, Klassen- und Sequenzdiagramme	Vorstellung kooperierender Komponenten <i>kann</i> in Use Cases zum Ausdruck kommen
	PARC, ~1980 <i>MVC</i>	Realisierungs- bezogenes Modell interaktiver Systeme	Dokumentation von Konzepten zur Realisierung	Systemaufbau, und -verhalten, (erst später: Softwarestruktur)	Ursprünglich: Code, Text, freie Grafik Heute auch: Klassendiagramme	Vorstellung kooperierender Komponenten Integration in das interaktive Gesamtsystem unklar
	Coulaz, 1987 <i>PAC</i>	Realisierungs- bezogenes Modell interaktiver Systeme	Dokumentation von Konzepten zur Realisierung	Systemaufbau, und -verhalten, (erst später: Softwarestruktur)	Ursprünglich: Text, freie Grafik Heute auch: Klassendiagramme	Vorstellung kooperierender Komponenten Integration in das interaktive Gesamtsystem unklar
	Green, 1985 <i>Seeheim</i>	Realisierungs- bezogenes Modell interaktiver Systeme	Dokumentation von Konzepten zur Realisierung	Systemaufbau, und -verhalten (Trägersystem), Softwarestruktur (Rollensystem)	Ursprünglich: Text, freie Grafik Heute auch: Klassendiagramme	Vorstellung kooperierender Komponenten Einzelne Sachverhalte unklar

Tabelle 4.1: Vergleich ausgewählter Beschreibungsansätze

4.1 Beschreibung von Interaktionsprozessen

4.1.1 Transitionsdiagramme — Parnas

Parnas schlägt in [Par69] „die Benutzung von Transitionsdiagrammen beim Design von Benutzungsschnittstellen interaktiver Computersysteme“ vor. Gegenstand einer Beschreibung durch Transitionsdiagramme ist nach dem vorgestellten Ansatz die Benutzungsschnittstelle interaktiver Computersysteme auf der Ebene interaktionsbezogener Modelle.

Primärer Zweck der Transitionsdiagramme ist die Beurteilbarkeit von Entwürfen des geplanten Verhaltens interaktiver Systeme auf der Ebene interaktionsbezogener Modelle durch einen Designer zur Verbesserung der Systembenutzbarkeit. Als sekundäre Anwendungsmöglichkeiten schlägt Parnas die Diagramme als Bestandteil von Dokumentation (in Benutzerhandbüchern), als Gegenstand automatisierter Testverfahren und als Programmkomponente zur Systemrealisierung durch spezielle Abwickler vor.

Ein vor diesem Hintergrund erstelltes Transitionsdiagramm beschreibt ein Automatenmodell des interaktiven Systems aus Sicht eines Benutzers, der an der Benutzungsschnittstelle konkrete Symbolfolgen als Ein- und Ausgaben erlebt. Die Aufbaustruktur, die dem Modell von Parnas zugrunde liegt, ist in Abbildung 4.1 gezeigt. Der Benutzer kommuniziert mit einer Komponente, die als Terminal bezeichnet wird und die das entsprechende Automatenmodell realisiert. Auf eine Eingabe des Benutzers reagiert das Terminal, abhängig von seinem Zustand, mit dem Anstoß einer entsprechenden Systemoperation und gibt danach eine korrespondierende Rückmeldung als Antwort aus oder erzeugt selbstständig eine zustandsabhängige Antwort. In beiden Fällen wechselt das Terminal danach in den vordefinierten Folgezustand. Terminal und „System“ zusammen bilden einen Steuerkreis. Das Terminal agiert dabei in der Rolle des Steuerwerks. Das Automatenmodell bleibt dadurch — unabhängig von der Komplexität des „Systems“ in der Rolle des Operationswerkes — vergleichsweise überschaubar.

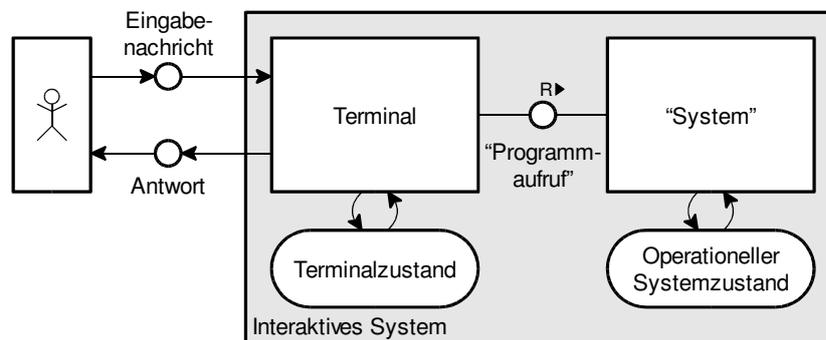


Abbildung 4.1: Aufbau des Terminal-Systems

Ein Transitionsdiagramm ist die grafische Beschreibung eines Automatenmodells, bei der jedem Terminalzustand ein kreisrunder Knoten zugeordnet ist. Diese Knoten sind

durch gerichtete Kanten, die mögliche Zustandsübergänge symbolisieren, miteinander verbunden. Abbildung 4.2 zeigt ein Beispiel aus [Par69].

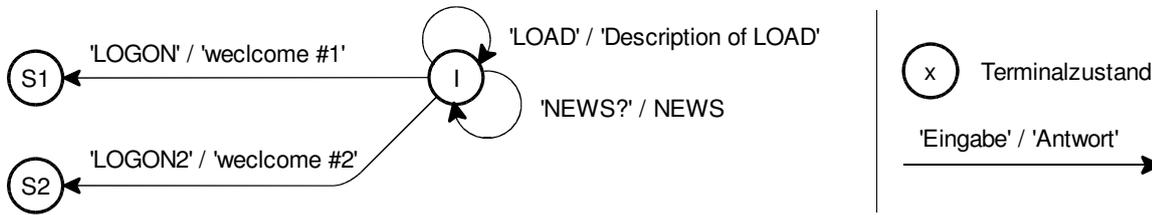


Abbildung 4.2: Ausschnitt aus einem Transitionsdiagramm

Der gezeigte Ausschnitt aus einem Transitionsdiagramm beschreibt die Symbolfolgen möglicher Eingaben eines Benutzers, die Symbolfolgen dadurch verursachter Ausgaben, sowie die Zustandsübergänge ausgehend vom Initialzustand I des betrachteten interaktiven Systems. In diesem Zustand ist es möglich, Neuigkeiten und die Systemauslastung abzufragen, sowie durch Anmeldung in einen von zwei nicht näher spezifizierten Bedienzuständen zu gelangen. Zur Beschreibung von Eingaben und Ausgaben sind neben der Angabe konkreter Symbolfolgen (durch „“ geklammert) auch Platzhalter möglich (ohne „“), die wahrscheinlich namensgleiche, operationelle Speichervariablen identifizieren.

Unklar ist, inwieweit die Aufbaustruktur, welche Parnas beschreibt, als interaktionsbezogenes Modell oder als realisierungsbezogenes Modell zu deuten ist. Im ersten Fall wäre das Terminal Resultat einer Entwurfsentscheidung auf Ebene interaktionsbezogener Modelle. Damit wäre es als Sonderfall eines Schrittfolgenadapters anzusehen und auf Ebene des zweckbezogenen Modells wären Aufträge in beliebiger Reihenfolge an das interaktive System denkbar. Andererseits reflektiert das Automatenmodell auch zweckbezogenes Verhalten. Die Aufteilung des Systems in Terminal und „System“ wäre dann als realisierungsbezogene Entscheidung zu werten, auch wenn Parnas sie im Hinblick auf das Steuerkreismodell durch die Unterscheidung von Steuerzustand und operationellem Zustand begründet. Tatsächlich gibt es in [Par69] Passagen, die den Eindruck vermitteln, dass Parnas einen realisierungsbezogenen Aufbau vor Augen hatte². Die Beschreibung des Systemverhaltens an der Benutzungsschnittstelle durch Transitionsdiagramme kann hingegen eindeutig der Ebene interaktionsbezogener Modelle zugeordnet werden.

Es ist offensichtlich, dass die Anwendung dieses Ansatzes ohne Veränderungen nur für Systeme geringer Komplexität zweckmäßig ist. So gibt es weder ein Hierarchisierungskonzept (siehe [Was85, Har88]) noch besitzt der Systemkern eine explizite innere Struktur. Das interaktive System wird als *ein* monolithischer Dialogpartner angesehen, mit dem ein Benutzer in einem streng sequentiellen Interaktionsprozess kommuniziert. Die Beschreibung nebenläufiger Interaktionsprozesse ist nicht vorgesehen. Für das Design erschwerend erweist sich die fehlende Trennung von zweckbezogenen und interaktionsbezogenen Modellen.

²siehe [Par69], S. 383: Three types of input [...]: Calls on programs in some library [...]

4.1.2 Multiparty–Grammatiken — Shneiderman

In [Shn82] wird eine Möglichkeit der Beschreibung von Interaktionsprozessen durch Grammatiken vorgeschlagen. Gegenstand solcher Beschreibungen ist das Verhalten von Benutzern und interaktivem System, das an der Benutzungsschnittstelle auf der Ebene zweckbezogener und interaktionsbezogener Modelle beobachtbar ist.

Primärer Zweck solcher Grammatiken ist die automatische Auswertung. Dazu gehört die Überprüfung der Grammatiken auf Konsistenz und Vollständigkeit hinsichtlich der spezifizierten Interaktionsprozesse sowie die Ableitung ausführbarer Programme. Sekundäre Möglichkeiten der Verwendung sieht Shneiderman in der Ableitung von Dokumentation, der Erstellung von Hilfsassistenten und in der Anwendung von Metriken zur Bewertung der Benutzbarkeit.

Zur Definition unendlicher Mengen endlicher Symbolfolgen haben sich *Grammatiken* als spezielle *axiomatische Systeme* durchgesetzt [Wen91]. Symbolfolgen, die auf einer *attribuierten Grammatik* basieren, identifizieren darüber hinaus einen oder mehrere Sachverhalte. Shneiderman betrachtet alle Interaktionsformen, die zwischen den kommunizierenden Komponenten eines Interaktionssystems nacheinander beobachtbar sind, als eine einzige ununterbrochene Symbolfolge. Einzelne Abschnitte dieser Folge entsprechen Nachrichten bzw. Aktionen der einzelnen Systemkomponenten. Basierend auf dieser Vorstellung definiert eine Multiparty–Grammatik die Menge aller möglichen Interaktionsprozesse, die zwischen den agierenden Komponenten innerhalb des Interaktionssystems beobachtbar sind. Typischerweise treten bei Shneiderman dabei zwei Beteiligte in Erscheinung: Der Benutzer („Human“) und das interaktive System („Computer“). Ausgehend vom Axiom der Grammatik, das für den Interaktionsprozess als Ganzes steht, sind zunächst Ableitungsregeln anwendbar, welche die semantische Struktur des Interaktionsprozesses als Folge entsprechend benannter Superzeichen sichtbar werden lassen. Davon ausgehend sind Ableitungsregeln anwendbar, welche die syntaktische Ebene der konkreten Protokolle und Terminalfolgen festlegen.³

Abbildung 4.3 zeigt das in [Shn82] vorgestellte Beispiel der Anmeldung eines Benutzer an einem Computersystem. Auf oberster Betrachtungsebene besteht der Anmeldeprozess aus einer Startphase (START) und einer Zugangscodeprüfung (ACCT für account). Durch eine entsprechende Benennung der Superzeichen wird die Semantik der beiden Prozessabschnitte festgelegt. Im Sinne von Kapitel 3 wird hierdurch das Verhalten auf zweckbezogener Ebene beschrieben. Für jede der beiden Phasen sind zwei alternative Ausprägungen eines Dialoges zwischen Benutzer und Computer vorgesehen. Die Zuordnung einzelner Superzeichen zum jeweiligen Sender erfolgt in der Grammatik durch einen sendertypischen Präfix in der Symbolfolge, die das Superzeichen repräsentiert. Solche Grammatiken sind im Hinblick auf ihren primären Zweck primär als maschinen–lesbare Spezifikationen der zulässigen Interaktionsprozesse zu werten. Eine für Menschen verständlichere Form ist sicherlich die gezeigte Alternativdarstellung durch ein Transitionsdiagramm. Die Ablaufstruktur, welche bei grafischer Darstellung sofort erkennbar ist, gewinnt man bei der Grammatik erst durch

³„Multiparty grammars describe the syntax of interactions and a small portion of what might be called the semantics“. Eine vertiefende Erklärung dieser Aussage findet sich in [Shn82] nicht. Die folgenden Ausführungen basieren daher vorallem auf der Interpretation des angeführten Beispiels.

1. <LOGON> ::= <START><ACCT>
2. <START> ::= <H: INITIATE><C: READY-ACCT> | <H: INVALID-INITIATE><C: CRLF-REQUEST>
3. <H: INITIATE> ::= I␣
4. <H: INVALID-INITIATE> ::= <H: *>␣
5. <C: READY-ACCT> ::= READY FOR ACCOUNT NUMBER␣
6. <C: CRLF-REQUEST> ::= TO SIGNON TYPE AN "I" AND HIT ENTER
7. <ACCT> ::= <H: VALID-ACCT><C: ACCEPT-ACCT> | <H: * ><C: ACCT-REQUEST>
8. <H: VALID-ACCT> ::= <H: NUM><H: NUM><H: LETTER>␣
9. <C: ACCEPT-ACCT> ::= LAST SIGNON FOR [<H: VALID-ACCT>] WAS <LAST-SIGNON-INFO>
10. <C: ACCT-REQUEST> ::= ACCOUNT NUMBERS ARE TWO DIGITS FOLLOWED BY A LETTER␣ <C:READY-ACCT>

Superzeichen der Form <H: ...> repräsentieren Terminalfolgen des Benutzers (H für Human), <C: ...> Terminalfolgen des Computers.

<H: *> steht für beliebige Terminalfolgen des Benutzers, wenn keine der anderen Ersetzungsregeln anwendbar ist.

<H: NUM> ::= 0|1|2|3|4|5|6|7|8|9
 <H: LETTER> ::= A|B|C...|Z
 ␣ steht für den Carriage Return und Linefeed Code

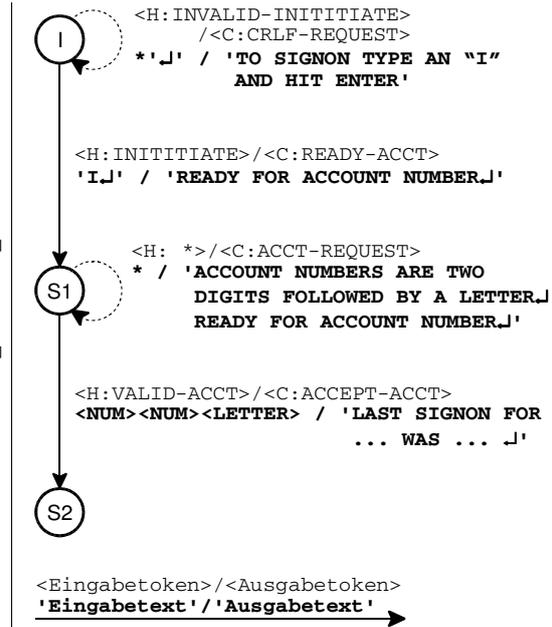


Abbildung 4.3: Multiparty-Grammatik (links) mit Alternativdarstellung (rechts)

schrittweises Anwenden der verschiedenen Produktionsregeln.

Zur Vermittlung einer Systemvorstellung ist der vorgestellte Ansatz nicht geeignet. Die Komponenten eines Interaktionssystems werden benannt und das Repertoire möglicher Interaktionsprozesse wird beschrieben. Dennoch bleibt die Kausalstruktur des Systems unsichtbar. Vorhersagen über das Verhalten von interaktiven Systemen, die zustandsbehaftet sind, sind auf diesem Wege nicht möglich und würden zusätzliche Beschreibungsmittel erfordern. Eine Möglichkeit, einzelne Regeln einer Multiparty-Grammatik mit anwendungsspezifischen zustandsverändernden Operationen zu assoziieren, wird daher bereits in [Shn82] als notwendige Erweiterung angesehen. Insofern stellt eine Multiparty-Grammatik im allgemeinen kein vollständiges Modell eines interaktiven Systems dar. Die Beschreibung nebenläufiger Interaktionsprozesse ist zwar prinzipiell möglich aber aufwendig und im Ergebnis unverständlich. Darüber hinaus muss bezweifelt werden, ob es angemessen ist, das gesamte Verhalten eines beliebigen Interaktionsaktionssystems, das beliebig lange andauern und eine Vielzahl semantisch unabhängiger Interaktionsprozesse umfassen kann, als eine einzige lineare Symbolfolge anzusehen. Der Ansatz ist daher vorrangig als Formalismus zu verstehen, dessen Anwendung einzelne Aspekte bestehender Modelle von überschaubaren interaktiven Systemen einer automatischen Auswertung zugänglich macht. Konzepte zur Realisierung dazu erforderlicher Systeme werden angedeutet, Systemmodelle werden nicht vermittelt.

4.2 Beschreibung und Design interaktiver Systeme

4.2.1 SA/SD und HCI-Design — Sutcliffe

Sutcliffe stellt in [SM91] einen Ansatz zur Entwicklung interaktiver Systeme vor, der Methoden zum Design von Benutzungsschnittstellen (Human computer interface design, kurz HCI-Design) und die Entwicklung informationeller Systeme nach dem Prinzip von „Strukturierter Analyse und Design“ (SA/SD) [DeM78] vereint. Gegenstand von Beschreibungen, die auf der Anwendung dieses Ansatzes beruhen, sind Interaktionssysteme auf der Ebene zweckbezogener, interaktionsbezogener und zum Teil realisierungsbezogener Modelle.

Primärer Zweck der verschiedenen Arten von Beschreibung ist das Festhalten von Modellierungs- und Entwurfsentscheidungen als Grundlage nachgelagerter Phasen der *Softwareentwicklung*. Unklar ist, in welchem Maße die Beschreibungen als gestalterisches Hilfsmittel im Rahmen der Modellfindung, zur Abstimmung unterschiedlicher Vorstellungen oder lediglich zum Festhalten von Ergebnissen dienen. Ungeklärt bleibt, inwiefern die Beschreibungen zur Kommunikation mit potentiell zukünftigen Benutzern, die gegebenenfalls in den Entwicklungsprozess integriert sind, genutzt werden.

Auf oberster Modellebene wird das zukünftige Interaktionssystem als Struktur gekoppelter Prozesse angesehen, während derer Daten ermittelt, verarbeitet, transportiert und gespeichert werden. Durch funktionale Verfeinerung lassen sich Strukturen hierarchisch untergeordneter Prozesse identifizieren. Dargestellt werden solche Strukturen in Form sogenannter Data Flow Diagrams. Abbildung 4.4 zeigt das oberste Data Flow Diagram (DFD) eines Bibliothekssystems aus [SM91]. Die Prozessfindung basiert auf der Analyse der Aufgaben, die zur Erzielung eines beabsichtigten Zwecks durchzuführen sind. Nach [DeM78] geschah dies ursprünglich auf der Basis bereits bestehender Systeme, wie Firmen oder Behörden, die durch Unternehmensstrukturen geprägt sind. Darauf aufbauend wird festgelegt, welche Prozesse allein durch das zu entwickelnde interaktive System, welche allein durch Komponenten bzw. Benutzer der Umgebung und welche interaktiv durchgeführt werden.

Eine Kausalstruktur, welche das Verhalten des Interaktionssystems definiert, ergibt sich erst durch die prozedurale Beschreibung der Elementarprozesse, die innerhalb der DFDs identifiziert werden. Die dazu verwendete formale an englischen Text angelehnte Notation wird als Structured English bezeichnet. Jeder so identifizierte Prozedurschritt stellt entweder eine Aktivität des interaktiven Systems oder eines Benutzers dar. In die Festlegung dieser Sachverhalte fließt noch kein Wissen um Konzepte und Möglichkeiten der Gestaltung von Benutzungsschnittstellen und um die Fähigkeiten zukünftiger Benutzer ein. Sie sind daher nach Abbildung 3.2 der Ebene zweckbezogener Modelle zuzuordnen. Die Festlegung der Systemgrenze bildet natürlich einen Berührungspunkt zur Ebene interaktionsbezogener Modelle. Dies ist grundsätzlich unvermeidbar, da bei der Aufteilung der Aktivitäten zwischen Benutzern und interaktivem System, der Zweck des interaktiven Systems durch Nutzbarkeitserwägungen definiert wird.

Auf der Ebene interaktionsbezogener Modelle wird der konkrete Interaktionsprozess zwischen Benutzern und interaktivem System in Form von Automatenmodellen be-

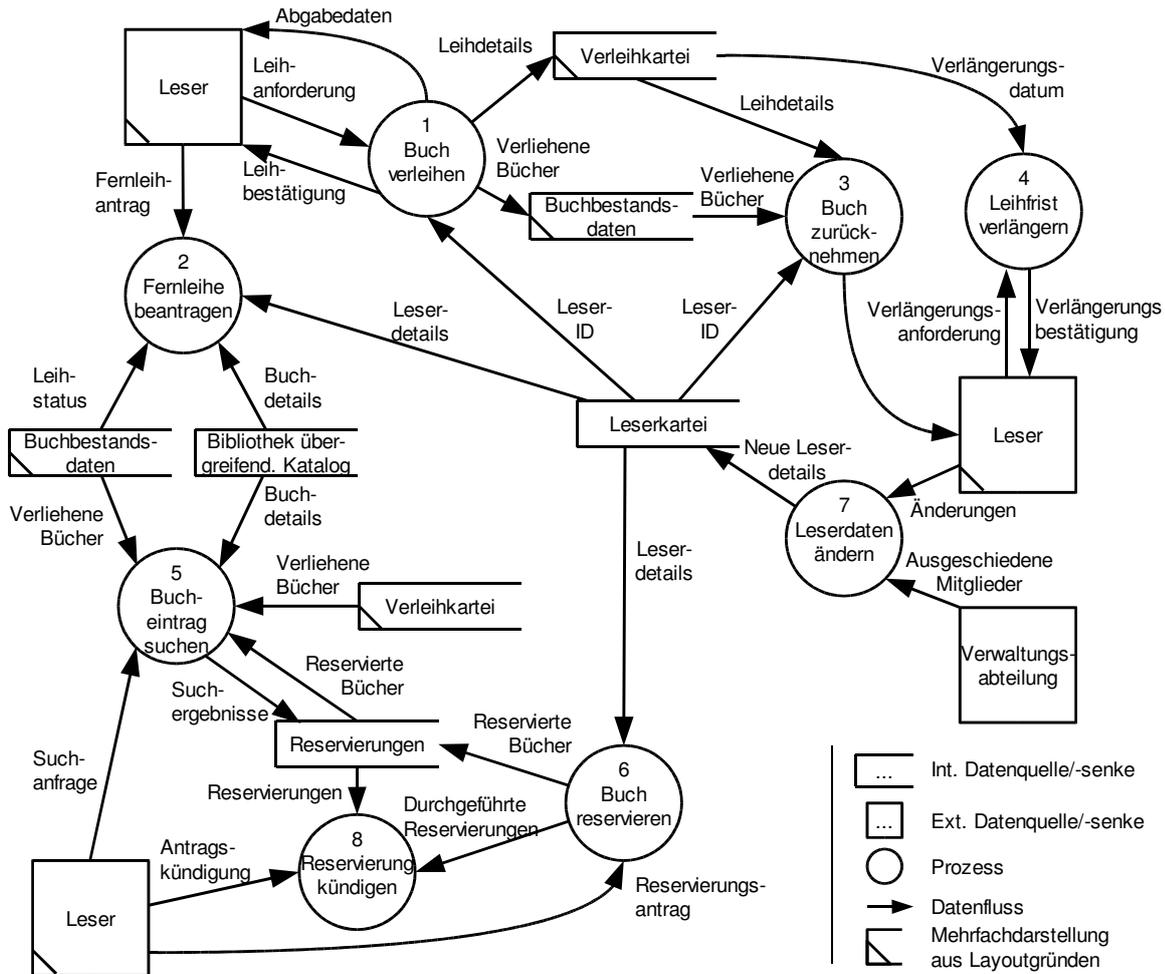


Abbildung 4.4: Data Flow Diagramm für ein Bibliothekssystem

schrieben. Grundlage dieser Modelle ist eine formale Abbildung von DFDs und prozeduralen Prozessbeschreibungen auf Dialogdiagramme, eine spezielle Form von Transitionsdiagrammen. Die verschiedenen Ausgabeaktivitäten des interaktiven Systems werden auf korrespondierende Dialogzustände abgebildet. Zustandsübergänge werden durch Benutzereingaben ausgelöst. Die konkrete Form von Eingaben und Ausgaben ist durch Zeichenketten und Bildschirmmasken definiert, die mit den Kanten bzw. Knoten der Dialogdiagramme verknüpft sind. Die Berücksichtigung von Konzepten zur Steigerung der Benutzbarkeit äußert sich in der Erweiterung der Dialogdiagramme um zusätzliche Interaktionsschritte beispielsweise zur Abkürzung von Bedienfolgen oder zur Ausgabe von Fehlermeldungen. Die mehrstufige Gruppierung voneinander unabhängig betrachtbarer Dialoge führt zu einer Dialoghierarchie mit übergeordneten Auswahldialogen (Menüs). Die Ausgabe zweckbezogener Informationen wird abhängig vom Dialogzustand um Inhalte ergänzt, die zukünftigen Benutzern den Umgang mit dem System erleichtern sollen.

Dialogdiagramme, Structured English und Bildschirmdesigns bilden die Grundlage für eine Realisierung auf der Basis des Structured Designs. Dabei wird auf der Ebene realisierungsbezogener Modelle ein interaktives Systems als Hierarchie aktiver Kom-

ponenten zur Realisierung des zweckbezogenen Verhaltensmodells angesehen, denen entsprechende Dialogkomponenten zur Ein- und Ausgabe zugeordnet sind. Die Darstellung dieses Modells erfolgt durch sogenannte Structure Charts, die Ausdruck einer strikten Aufrufhierarchie sind und welche die Modul- bzw. Prozedurstruktur der zu erstellenden Programme definieren. Der in DFDs gezeigte Datenfluss spielt auf dieser Ebene der Modul- und Kontrollstrukturen keine Rolle⁴.

Die nach Sutcliffe entstehenden Beschreibungen vermitteln die Vorstellung von einem System, dessen Aufbaustruktur ein interaktives System mit unterschiedlichen Umgebungskomponenten darstellt. Eine innere Aufbaustruktur des interaktiven Systems wird auf der Ebene zweck- und interaktionsbezogener Modelle bei strenger Betrachtung nicht festgelegt. Durch Datenflussdiagramme werden die für den Betrieb des interaktiven Systems relevanten Daten sowie damit in Verbindung stehende Prozesse identifiziert. Jeder Prozess identifiziert einen Ausschnitt aus dem beobachtbaren Verhalten des Systems. Bei der expliziten Zuordnung von Prozessen zu Aufbaukomponenten wird nur zwischen Umgebungskomponenten und interaktivem System unterschieden. In diesem Sinne besitzt das betrachtete Interaktionssystem auf beiden Modellebenen dieselbe Aufbaustruktur. Das interaktionsbezogene Modell entsteht aus dem Zweckbezogenen durch Verfeinerung bzw. Erweiterung des beobachtbaren zweckbezogenen Verhaltens an der Benutzungsschnittstelle.

Natürlich lässt sich zu jedem Datenflussdiagramm eine zweite isomorphe sehr feine Aufbaustruktur assoziieren, indem man jedem Prozessknoten eins-zu-eins einen Akteur und jeder Datenflusskante einen Kanal oder eine Zugriffsmöglichkeit auf einen Speicher zuordnet. Mit dieser Art der Komponentenfindung ist aber keine Abstraktion im Sinne von Kapitel 3.3 verbunden. Sie trägt nicht zu einer Vereinfachung der Systemvorstellung bei. Zur Veranschaulichung dieser Aussagen ist in Abbildung 4.5 eine zweckmäßigere Aufbaustruktur des aus Abbildung 4.4 bekannten Bibliothekssystem gezeigt. Dazu wurden jeweils mehrere, eng miteinander gekoppelte Aktivitäten einer kleineren Zahl zuständiger aktiver Systemkomponenten zugeordnet, ohne das dabei wesentliche Informationen verloren gegangen wären. Die überschaubare an ästhetischen und inhaltlichen Kriterien ausgerichtete Form der Darstellung erlaubt ein vergleichsweise leichtes Erkennen der gezeigten Struktur.

Unklar bleibt bei Sutcliffe, zumindest in dem beschriebenen Beispiel, der Unterschied zwischen materiellem und informationellem Systemzweck, das heißt, zwischen Objekten und Informationen über Objekte. So bleibt offen, ob das beschriebene interaktive System lediglich zur Verwaltung von Informationen in einer Bibliothek oder auch zur Ausgabe und Rücknahme von Büchern und zur Annahme von Gebühren dient. Stellenweise scheint es, als besäßen die Mitglieder ein virtuelles Konto zur Verrechnung von Gebühren — oder werden Scheine und Münzen getauscht? Abhängig davon zeigt Abbildung 4.5 entweder das Ganze oder nur einen Ausschnitt des beschriebenen Interaktionssystems: Es ist eindeutig auf den rein informationellen Verwaltungsaspekt beschränkt.

⁴Problematisch daran ist, dass nicht erkennbar ist, welche Module von welchen Daten abhängen. Dies fördert die Verwendung global adressierbarer Speichervariablen und führt zu den bekannten damit verbundenen Problemen.

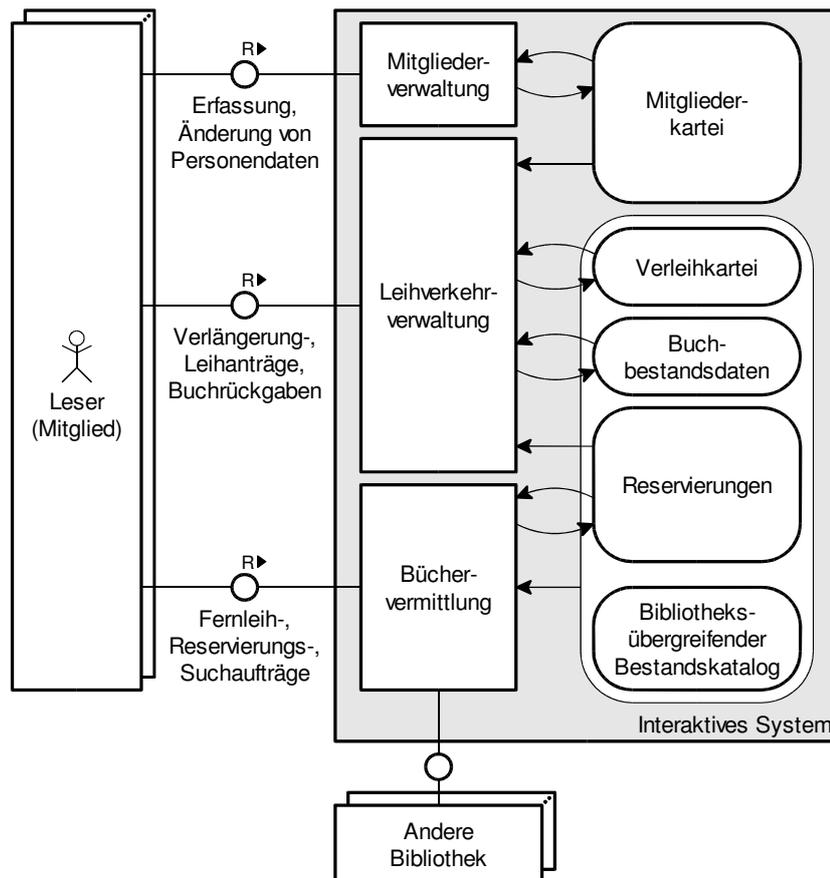


Abbildung 4.5: Zweckbezogene Aufbaustruktur des Bibliothekssystems

4.2.2 Task analysis und HCI-Design — Johnson

Johnson stellt in [Joh92] einen Ansatz zur Entwicklung interaktiver Systeme vor, der Methoden der sogenannten Aufgabenanalyse (engl. task analysis, siehe [KA92]) an den Anfang der Entwicklung interaktiver Systeme stellt. Gegenstand von Beschreibungen, die auf der Anwendung dieses Ansatzes beruhen, sind die Interaktionsprozesse und -formen, die an der Benutzungsschnittstelle interaktiver Systeme auf der Ebene zweckbezogener und interaktionsbezogener Modelle beobachtbar sind.

Primärer Zweck der verschiedenen Arten von Beschreibung ist das Festhalten von Ergebnissen der Analyse-Phase, sogenannter *Aufgabenmodelle* (engl. task models), und von darauf aufbauenden Entwurfsentscheidungen zur Gestaltung von Benutzungsschnittstellen. Diese dienen der Entwicklung von Prototypen und als Grundlage nachgelagerter Phasen der Softwareentwicklung. Die Entwicklung unterstützender Werkzeuge zur automatisierten Ableitung von Prototypen (ADEPT, [JWMP93]) lässt vermuten, dass die Beschreibungen in der Kommunikation mit zukünftigen Benutzern eine untergeordnete Rolle spielen.

Ein Aufgabenmodell umfasst das Wissen, das ein Mensch zur Durchführung einer bestimmten Aufgabe ('task') benötigt. Der Zweck einer solchen Aufgabe besteht darin, den vorgegebenen Ausgangszustand eines betrachteten Aktionsfeldes in einen gefor-

derten Endzustand ('goal') zu überführen. Die Identifikation verständlicher Zwischenzuständen ('subgoals') führt zu einer hierarchischen Zerlegung in Teilaufgaben. Eine Prozedur ('procedure') beschreibt ein Verfahren zur Lösung einer Teilaufgabe auf der Basis elementarer Operationen ('actions'), die auf Handlungsgegenstände ('domain objects') innerhalb des Aktionsfeldes angewendet werden. Ihren Ursprung hat die Aufgabenmodellierung in den fünfziger und sechziger Jahren des zwanzigsten Jahrhunderts zum arbeitsplatzspezifischen Training von Mitarbeitern in Militär und Industrie. Für jeweils einzelne Aufgaben wird unter Berücksichtigung der individuellen, kognitiven und physischen Fähigkeiten einer Person wird eine idealerweise optimale Methode zur Durchführung einer Aufgabe definiert. Johnson wendet diesen Ansatz in prinzipiell gleicher Weise zur Aufgabenmodellierung für Benutzer interaktiver Systeme an. An der Benutzungsschnittstelle werden Handlungsgegenstände und Aktionen allerdings durch den Austausch von Informationen definiert. Die Form dieser Informationen ist daher ein zusätzlicher Bestandteil der nach Johnson entstehenden Beschreibungen. Vereinfacht gesprochen werden dadurch Objekte und Aktionen des Aufgabenmodells auf Objekte und Aktionen an der Benutzungsschnittstelle abgebildet (vgl. [Shn98] Object-Action Interface Model).

Eine verbindliche Form der Beschreibung wird in [Joh92] nicht explizit vorgeschrieben. Die Beschreibung des Aufgabenmodells in zwei Fallstudien (Simulation eines Rubik-Würfels und computerunterstütztes Schmuck-Design) erfolgt teils in textueller tabellarischer Form, teils in Form eines graphischen Plans zur Darstellung der Hierarchie- und Folgebeziehungen zwischen den identifizierten Aufgaben bzw. Teilaufgaben. Die Form des Informationsaustausches an der Benutzungsschnittstelle kann durch „formale Spezifikationen, informale Spezifikationen, Bildschirmzickzen, Storyboards, Simulationen oder Prototypen“ festgelegt werden.

Johnson beschreibt die Entwicklung eines Systems zur Simulation und interaktiven Lösung eines Rubik-Würfels. Jede Seite eines solchen Würfel wird durch neun kleinere Würfel mit unterschiedlich farbigen Seiten gebildet. In jeder Raumrichtung ergeben sich so drei aneinander grenzende Ebenen kleinerer Würfel. Durch die Möglichkeit diese Ebenen gegeneinander zu verdrehen, lässt sich die Position der kleineren Würfel zueinander verändern. Die Aufgabe ('task') im Sinne des beschriebenen Aufgabenmodells besteht darin, ausgehend von einem beliebig verdrehten Würfel zu erreichen, dass jede Seite des Rubik-Würfels eine einheitliche Farbe ausweist ('goal'). Teilaufgaben ('subgoals') bestehen beispielsweise darin, die obere, mittlere und untere Ebene zu vervollständigen. Handlungsgegenstände ('objects') sind der Würfel als Ganzes, seine Seiten, die Ebenen und die einzelnen kleinen Farbflächen. Darauf anwendbare Operationen ('actions') sind unter anderem die Auswahl einer geeigneten Seite, das Verdrehen einer Ebene oder Drehen des ganzen Würfels.

Diesem Aufgabenmodell muss, wie links oben in Abbildung 4.6 gezeigt, die Vorstellung von einem System zugrunde liegen, dessen Aufbaustruktur lediglich aus einem Menschen besteht, in dessen Aktionsfeld sich der Rubik-Würfel befindet. Explizit sichtbar wird diese Struktur nirgendwo.

Der Zweck des Simulationssystems ist rein informationeller Natur und nicht durch die Veränderung bestimmter materiell-energetischer Größen oder Objekte definiert. Das System soll einem Benutzer die Vorstellung von einem Rubik-Würfel und dessen

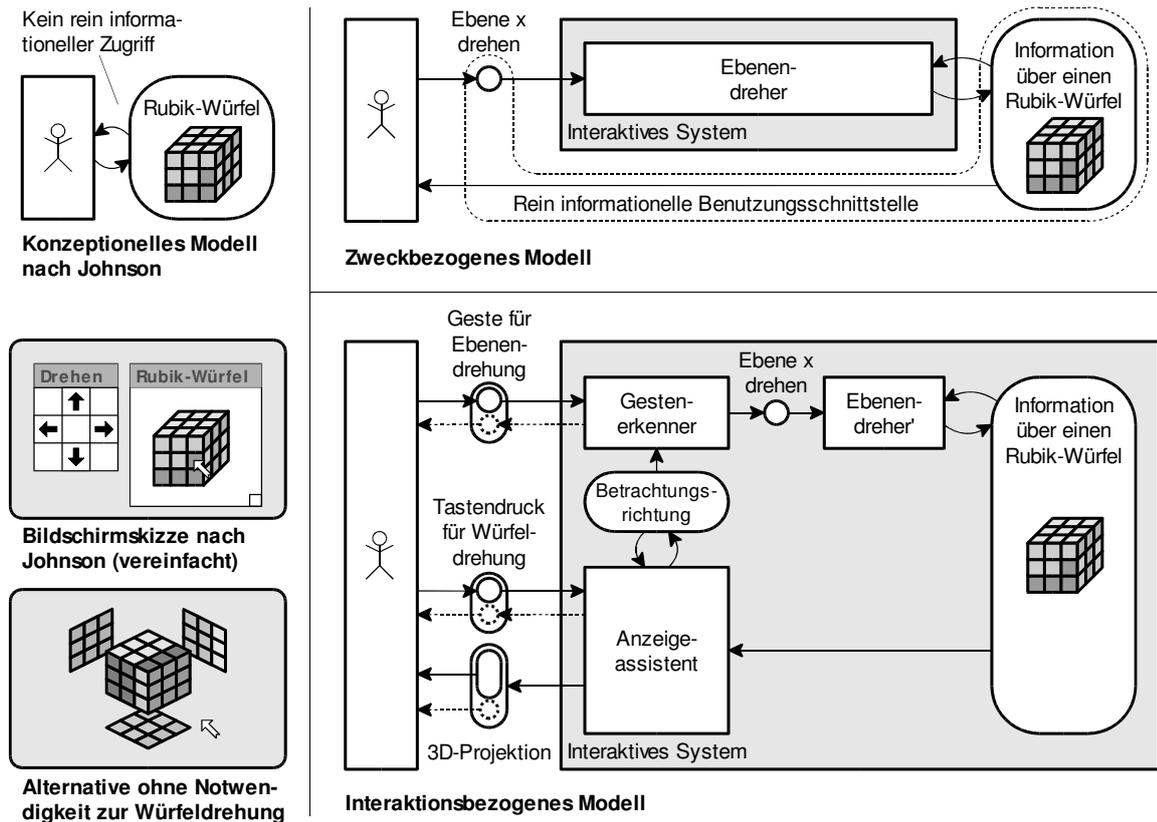


Abbildung 4.6: Interaktives System zur Simulation eines Rubik-Würfels

interaktiver Lösung geben. Eine Unterscheidung zwischen gegenständlichen Objekten und Informationen über gegenständliche Objekte, die über die Benutzungsschnittstelle des interaktiven Systems austauschbar sind, findet in [Joh92] aber nicht statt. In Folge dieser Gleichsetzung ist der informationelle Charakter des interaktiven Systems mit seiner rein informationellen Benutzungsschnittstelle im Aufgabenmodell nirgendwo erkennbar. Deshalb kann es keinen Übergang von ausgetauschter Information zu informationstragender Form geben, so dass eine klare Abgrenzung von zweckbezogener und interaktionsbezogener Modellebene unmöglich ist. Das Aufgabenmodell enthält daher einerseits zweckgebundene elementare Operationen und andererseits interaktionsgebundene elementare Operationen. So ist das Verdrehen einer Ebene eine Operation, die unabhängig davon erforderlich ist, ob man einen realen Würfel in der Hand hält oder ob man ein interaktives System benutzt, das einem lediglich die Vorstellung eines solchen Würfels vermittelt. Ein Drehen des gesamten Würfels, um ihn von allen Seiten betrachten zu können, ist bei einem interaktiven System, wie die untere Bildschirm-skizze in Abbildung 4.6 andeutet, nicht in jedem Falle erforderlich. Ein solches System entspräche nicht dem beschriebenen Aufgabenmodell, obwohl es zweifelsfrei seinen Zweck erfüllen würde. So wird es in [Joh92] sogar ebenfalls als denkbare Lösung angesprochen. Das zugrundeliegende konzeptionelle Modell, das auf der Gleichsetzung von Objekten und Information über selbige Objekte beruht, muss daher entweder als unzulässige Vereinfachung oder als Ergebnis einer vorweggenommenen interaktionsbezogenen Entwurfsentscheidung angesehen werden. Auch

wenn diese Entwurfsentscheidung offensichtlich nicht als verbindlich angesehen werden darf, kann sie die Sicht auf alternative Entwurfsentscheidungen verstellen.

Diese Probleme lassen sich vermeiden, wenn man das interaktive System als eigenständige Systemkomponente mit einer rein informationellen Benutzungsschnittstelle im Sinne von Kapitel 3 auffasst. Zur Veranschaulichung ist im rechten Bereich von Abbildung 4.6 ein zweckbezogenes Modell des betrachteten Interaktionssystems gezeigt. Anstelle des Würfels tritt die Information über einen Würfel. Ein realer Würfel wäre in diesem Sinne nur ein möglicher Informationsträger. Die bei Johnson materiell-energetische Operation des Verdrehens einer Würfelebene findet nun, vom Benutzer durch eine informationelle Schnittstelle entkoppelt, durch das interaktive System statt. Das darunter dargestellte interaktionsbezogene Modell zeigt darauf aufbauend eine mögliche Benutzungsschnittstelle, deren Erscheinung der oberen Bildschirmkizze entsprechen könnte.

4.2.3 Task analysis, Objekt-Orientierung und HCI-Design — Stary

Stary beschreibt in [Sta96] einen Ansatz zur Entwicklung interaktiver Systeme, der Methoden der Aufgabenanalyse mit Konzepten der Objekt-orientierten Softwareentwicklung verbindet. Gegenstände der Beschreibungen, die auf der Anwendung dieses Ansatzes beruhen, sind das Interaktionssystem auf der Ebene zweckbezogener Modelle und das interaktive System auf der Ebene interaktionsbezogener Modelle.

Primärer Zweck der entstehenden Beschreibungen ist das Festhalten von Ergebnissen der Aufgabenanalyse und der darauf aufbauenden OO-Analyse. Diese Ergebnisse werden zur Generierung von Prototypen (TADEUS, [Sta00]) und als Grundlage nachgelagerter Phasen in der Softwareentwicklung (OO-Design) genutzt. Die verwendete graphische Notation soll „Experten des Anwendungsbereiches, Benutzern, Analysten, Designern, Ergonomen und Programmierern“ den „Prozess und die Resultate der Entwicklungsaktivitäten leicht zugänglich und nachvollziehbar“ erscheinen lassen.

Ein *Aufgabenmodell* ist das Ergebnis einer Aufgabenanalyse, wobei die einzelnen erkannten Aufgaben Typen unterschiedlicher Prozesse darstellen, die sich der Ebene zweckbezogener Modelle zuordnen lassen. Durch ein Aufgabenmodell werden die Aufgaben identifiziert, zueinander in Beziehung gesetzt und einer Benutzergruppe zugeordnet. Unterschiedliche Aufgaben können zueinander entweder auf einer Ebene in einer Abfolge-Beziehung oder auf unterschiedlichen Ebenen in einer Teilaufgaben-Beziehung stehen. Das sogenannte *Problembereichsmodell*, sowie das *Benutzermodell*, gehören ebenfalls zur Ebene zweckbezogener Modelle. Beide Modelle definieren zusammen die Wertebereichsstruktur für alle im Interaktionssystem beobachtbaren zweckbezogenen operationellen Daten. Der Name Benutzermodell ist irreführend. Es handelt sich dabei nicht um ein Modell des Benutzers, sondern um die Struktur benutzerspezifischer Daten, wie zum Beispiel das Passwort oder eine persönliche Systemkonfiguration, die im interaktiven Systems gespeichert sind. Diese Daten sind durch die Verwendung eines interaktiven Systems zu erklären, welches im Rahmen der Aufgabenanalyse bei [Sta96] aber nicht explizit berücksichtigt wird.

Das *Interaktionsmodell* identifiziert die Typen aller Interaktionselemente, die zur Realisierung der Benutzungsschnittstelle verwendet werden, sowie deren anwendungs-

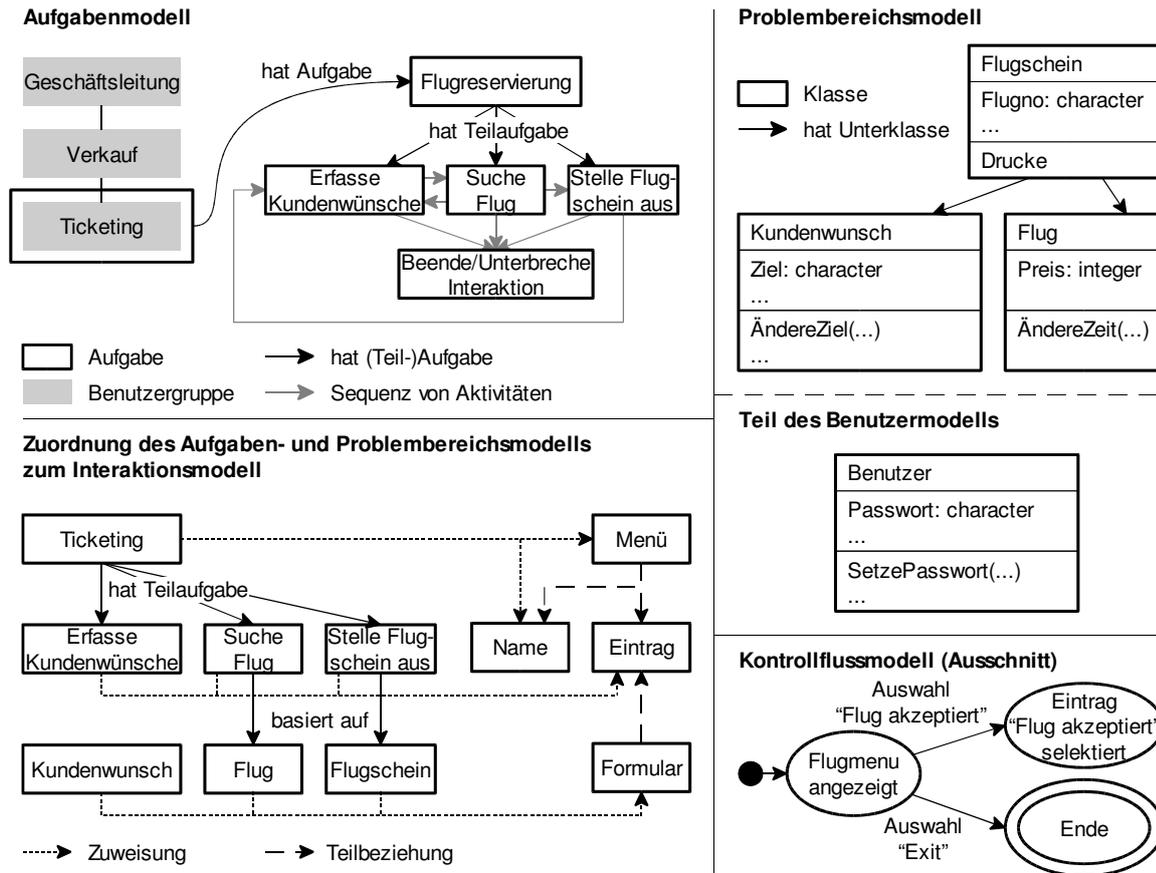


Abbildung 4.7: Aufgabenmodell und OO-Design-Modell für ein Flugreservierungssystem

unabhängige Beziehungen. Elemente eines solchen Modells sind beispielsweise die Typen unterschiedlicher Knöpfe, Menüs und Textfelder. Durch die Zuordnung des Aufgaben- und Problemsbereichsmodells zum Interaktionsmodell wird das interaktive System erstmals auf der Ebene interaktionsbezogener Modelle sichtbar. Dabei werden den Typen unterschiedlicher Aufgaben und Daten eindeutig Typen bestimmter Interaktionselemente zugeordnet. Das Verhalten des interaktiven Systems wird in Bezug auf seine Interaktionselemente und seine operationellen Daten durch das sogenannte Kontrollflussmodell in Form von Zustandsdiagrammen definiert.

Zur Beschreibung nutzt Story für die Darstellung des Aufgabenmodells hierarchische Graphen. Für die übrigen (Teil-)Modelle werden Klassendiagramme und Zustandsgraphen eingesetzt. Kausale Abhängigkeiten zwischen den Aufgaben äußern sich ebenso wie die Zuordnungsbeziehungen zwischen den verschiedenen (Teil-)Modellen durch zusätzliche gerichtete Kanten. Abbildung 4.7 zeigt als Beispiel die Beschreibung eines Flugreservierungssystems aus [Sta96].

Zur Vermittlung einer aufbaubasierten Systemvorstellung ist der vorgestellte Ansatz nicht geeignet. Die Phase der Aufgabenanalyse dient dazu, Informationen über eine bestehende Organisationsstruktur zu sammeln, die einem bestimmten Zweck dienlich ist. Davon ausgehend werden das Aufgabenmodell sowie — unter Anwendung von Methoden der Objekt-orientierten Analyse ([CY91]) — das Problembereichsmodell er-

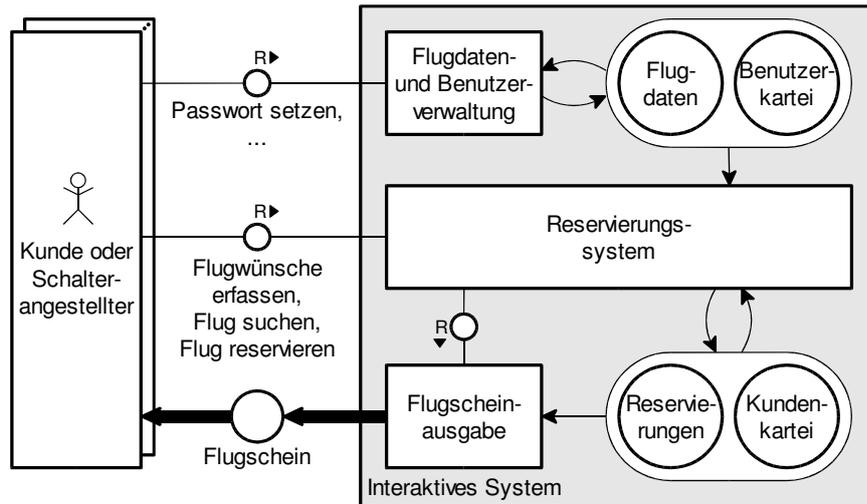


Abbildung 4.8: Zweckbezogene Aufbaustruktur des Flugreservierungssystems

stellt. Wie schon bei Johnson führt auch bei Stary die Anwendung der Aufgabenmodellierung dazu, dass weder das interaktive System explizit als eigenständige Systemkomponente, noch irgendeine andere Aufbaustruktur des Interaktionssystems identifiziert wird. Darüber, von wem welche Daten zu welchem Zweck benötigt werden, welche Daten flüchtig auf Kanälen beobachtbar sind und welche dauerhaft gespeichert werden, kann ausgehend vom Systemzweck nur spekuliert werden. Wie eine solche Struktur aussehen könnte, ist in Abbildung 4.8 gezeigt. Anders als bei Johnson scheint das Aufgabenmodell jedoch keine interaktionsbezogenen Entscheidungen vorwegzunehmen.

Auf der Ebene interaktionsbezogener Modelle wird das interaktive System implizit durch die Abbildung von Aufgaben und Datentypen auf Interaktionselemente, wie Menüs und Knöpfe, eingeführt. Das interaktionsbezogene Modell entsteht durch eine Ergänzung des zweckbezogenen Modells. Im Bezug auf [You83], der zwischen Mappings (Abbildungen) und Surrogates (Aufbauvorstellungen) unterscheidet, kann Stary's Ansatz als Beispiel eines reinen Mapping-Ansatzes verstanden werden. Eine Vorstellung von der Aufbaustruktur des interaktiven Systems wird folglich auch auf dieser Modellebene nicht vermittelt.

Die formulierte Absicht, „den Prozess und die Resultate der Entwicklungsaktivitäten leicht zugänglich und nachvollziehbar werden zu lassen“, setzt ein vorab erworbenes Systemverständnis voraus. Die tragende Stellung von Klassendiagrammen in der Beschreibung betont Typhierarchien. Das Verständnis von Instanzbeziehungen wird hierdurch zurückgedrängt, was den Aufbau einer Systemvorstellung erschwert. Diese Einschätzung der Verständlichkeit objekt-orientierter Beschreibungsansätze, die von Klassendiagrammen dominiert sind, ist in einer Reihe von Veröffentlichungen dokumentiert [Moy96, Hat98, Gla02].

4.2.4 Rational Unified Process und HCI-Design — Kruchten

In [KAB01] wird ein Ansatz zur Entwicklung interaktiver Systeme vorgestellt, der auf dem Rational Unified Process (RUP) basiert. Der RUP ist ein Software-Entwicklungsprozess⁵, der zur Entwicklung objekt-orientierter Softwaresysteme dient und von der Beschreibung einzelner Anwendungsszenarien (engl. Use Cases) ausgeht. Kruchten et al. erweitern diesen Prozess durch die Einführung und Festlegung der Rolle eines Benutzungsschnittstellendesigners. Zu dessen Aufgaben gehört die Erstellung von Beschreibungen des Interaktionssystem auf der Ebene interaktionsbezogener Modelle.

Primärer Zweck der entstehenden Beschreibungen ist das Festhalten von Entwurfsentscheidungen zur Ableitung von Prototypen und zur Weitergabe an Personen, die im Prozess nachgelagert tätig sind, sowie ihre Verwendung als Ausgangsmaterial bei Iterationszyklen. Auf diese Weise erlauben die Beschreibungen zudem die Nachvollziehbarkeit des Entwicklungsprozesses. Der RUP legt für die Personen, die am Entwicklungsprozess beteiligt sind, auszufüllende Rollen und deren Zusammenspiel fest. Zukünftige Benutzer treten im Hinblick auf die Gestaltung der Benutzungsschnittstelle nur als Prototyp-Tester in Erscheinung. Allerdings werden in [KAB01] auch schon Bildschirmsskizzen auf Papier als Prototyp bezeichnet.

Auf der Ebene interaktionsbezogener Modelle wird ein Interaktionssystem durch eine Menge sogenannter *Use Case Storyboards* beschrieben. Jedes dieser Use Case Storyboards beschreibt das Interaktionssystem auf der Grundlage eines vorgegebenen Anwendungsszenarios, welches jeweils einen geschlossenen verständlichen Ausschnitt der Systemnutzung auf der Ebene zweckbezogener Modelle beschreibt. Den Anfang eines Use Case Storyboards bildet eine umgangssprachliche Verhaltensbeschreibung der Interaktionen zwischen einem Benutzer und dem interaktiven System, das sogenannte *Flow of events-storyboard*. Durch Verfeinerung der vorgegebenen Anwendungsszenarien wird festgelegt, *wie* ein Benutzer bestimmte Dinge unter Benutzung des Systems erreichen kann. Merkmale zukünftiger Benutzer oder anderen mit der zukünftigen Nutzung verbundene Anforderungen werden identifiziert.

Alle Komponenten des interaktiven Systems, die durch eine solche Beschreibung identifiziert werden und die an der Benutzungsschnittstelle wahrnehmbar sein sollen, werden auf eine Hierarchie von Interaktionselementen, sogenannte Boundary Objects, abgebildet. Die Wurzel wird durch das Exemplar einer sogenannten zentralen Boundary-Klasse gebildet, welches das gesamte Szenario an der Benutzungsschnittstelle repräsentiert. Diesem entspricht im allgemeinen das zentrale Fenster einer interaktiven Anwendung bei einem Computersystem mit grafischer Benutzungsschnittstelle. Alle Interaktionsschritte, die in der Beschreibung des Szenarios identifiziert werden, werden mit entsprechenden Interaktionselementen in Beziehung gesetzt.

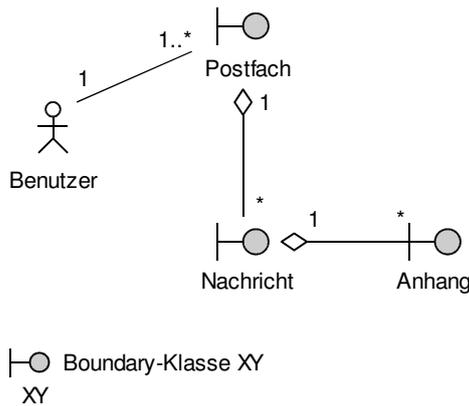
Ein Use Case Storyboard enthält formlose Texte, Klassendiagramme und Kollaborationsdiagramme. Zur Beschreibung der Typen unterschiedlicher Interaktionselemente

⁵Der Rational Unified Process wurde von der Firma Rational Software entwickelt und in Verbindung mit einer Software zur Prozessunterstützung vermarktet. Bekannt wurde die Firma durch Rational Rose, ein Werkzeug zur Modellierung objekt-orientierter Softwaresysteme mithilfe der Unified Modeling Language (UML). 2003 wurde Rational Software Teil der IBM.

(1) Formlose Beschreibung des Szenarios (Ausschnitt, erweitert um Benutzbarkeitsanforderungen (4))

- a) Das Szenario (der "Use case") beginnt damit, dass der Benutzer eine Anforderungen stellt, eMail-Nachrichten zu verwalten, und das System die Nachrichten auflistet. [Der Benutzer soll zwischen neuen, gelesenen und ungelesenen Nachrichten unterscheiden können; der Benutzer soll außerdem Absender, Betreffzeile und Priorität jeder Nachricht sehen.] {Durchschnittlich werden 100 ungelesene Nachrichten gleichzeitig angezeigt. In 90% der Fälle ist die Betreff-Zeile kürzer als 40 Buchstaben.}
- b) Der Benutzer kann danach einen oder mehrere der folgenden Schritte ausführen:
- 1) Die Nachrichten entsprechend Absender oder Betreffzeile sortieren
 - 2) Den Text einer Nachricht lesen
 - ...

(2) Klassendiagramm



(3) Kollaborationsdiagramm

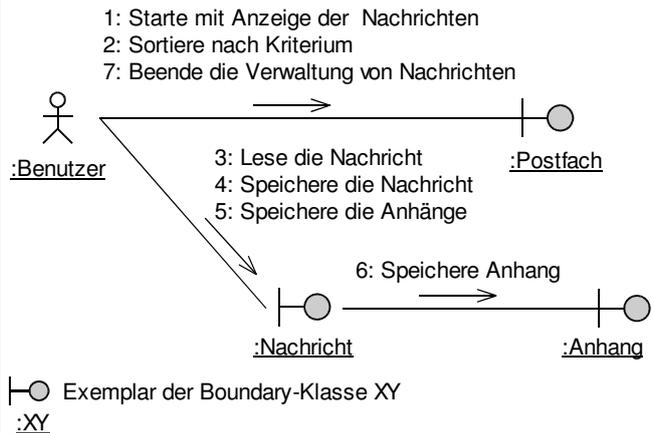


Abbildung 4.9: Use Case Storyboard zur Handhabung eingegangener eMail-Nachrichten

werden Verweise auf entsprechende Prototypen (inkl. Skizzen) empfohlen. Das Beispiel eines Use Case Storyboards, das den Umgang mit einem elektronischen Postfach beim Eingang und der Archivierung von Nachrichten zeigt, ist in Abbildung 4.9 dargestellt.

Welche Vorstellung von einem bestimmten interaktiven System durch eine Menge von Use Case Storyboards vermittelt werden kann, hängt wesentlich davon ab, welche Vorstellung der umgangssprachlichen Szenarienbeschreibung zugrundeliegt und wie sich diese Vorstellung in dieser Beschreibung widerspiegelt. In [KAB01] gibt es dafür nur wenige Anhaltspunkte: „Die Aussagen über das Systemverhalten“ sollen dazu dienen, „ein Verständnis von dem Szenario zu gewinnen“. Einem übergeordnetem Systemverständnis steht entgegen, dass die Menge aller Szenarios zwar das gesamte System erfassen soll, aber die einzelnen Szenariobeschreibungen sich in ihren Aussagen nicht überlappen sollen. Das Wissen um eine übergreifend gültige Aufbaustruktur, die gleichermaßen als Bühne für *alle* denkbaren Szenarios dient, wäre Kennzeichen einer aufbauverwurzelten Systemvorstellung. Eine solche Struktur wird in [KAB01] nirgends explizit sichtbar. Abbildung 4.10 kann daher nur den Ausschnitt einer solchen Aufbaustruktur zeigen, wie er sich aus dem geschilderten Szenario ableiten lässt.

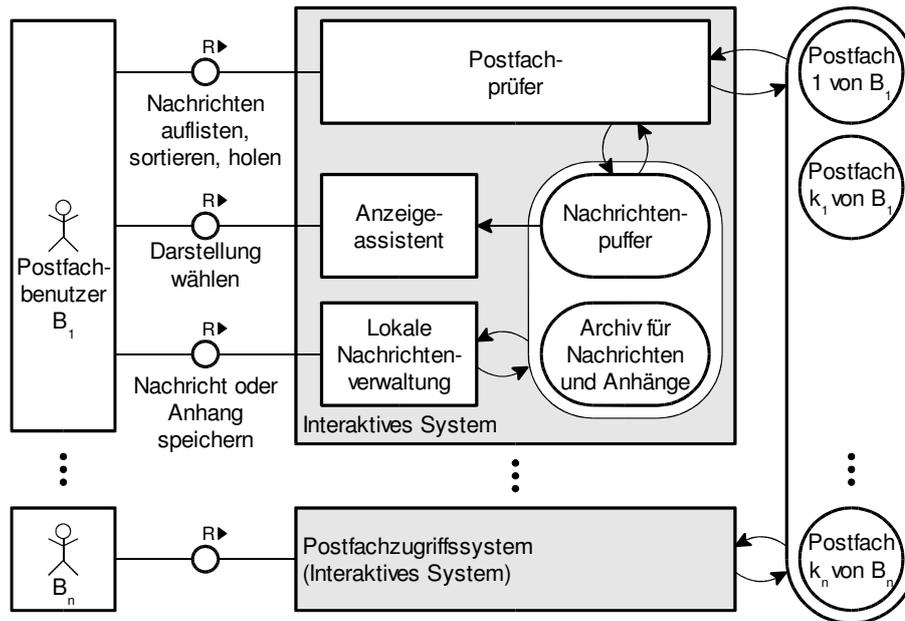


Abbildung 4.10: Aus dem Postfach-Szenario ableitbare Aufbaustruktur des zugrunde liegenden Interaktionssystems

4.3 Beschreibung von Architekturmodellen

In diesem Abschnitt wird die Darstellung von drei Architekturmodellen untersucht, die in der Literatur häufig beschrieben werden. Die vorgestellten Modelle sind der Ebene realisierungsbezogener Modelle zuzurechnen. Keines dieser Modelle zeigt ein konkretes System, sondern vielmehr den Typ einer bestimmten Realisierung beliebiger interaktiver Systeme. Die verständliche Beschreibung der zugrundeliegenden Konzepte ist Voraussetzung für die Schaffung von Systemen, welche die Merkmale eines bestimmten Architekturmodells erfüllen sollen. Der Wert eines Architekturmodells ist daher untrennbar mit der Qualität seiner Beschreibung verbunden. Dazu gehört ebenfalls die Beantwortung der Frage, wie die Abbildung zweck- und interaktionsbezogener Entwurfsentscheidungen zu erfolgen hat.

4.3.1 Model-View-Controller (MVC)

Model-View-Controller ist die Bezeichnung für eine recht bekannte Art der Strukturierung interaktiver Systeme auf der Ebene realisierungsbezogener Modelle. Im Kern handelt es sich um eine objekt-orientierte Realisierung, bei der der zweckbezogene Kern des interaktiven Systems durch Objekte vom Typ Model realisiert ist und das Interaktionswerk aus Objekten vom Typ View und Controller besteht. Besonders geeignet für diese Art der Realisierung sind interaktive Systeme mit einer grafischen Benutzungsschnittstelle, bei der viele Interaktionselemente gleichen oder ähnlichen Typs realisiert werden müssen. Durch das Konzept der Vererbung erlauben objekt-orientierte Programmiersprachen in diesen Fällen eine besonders effiziente Art der

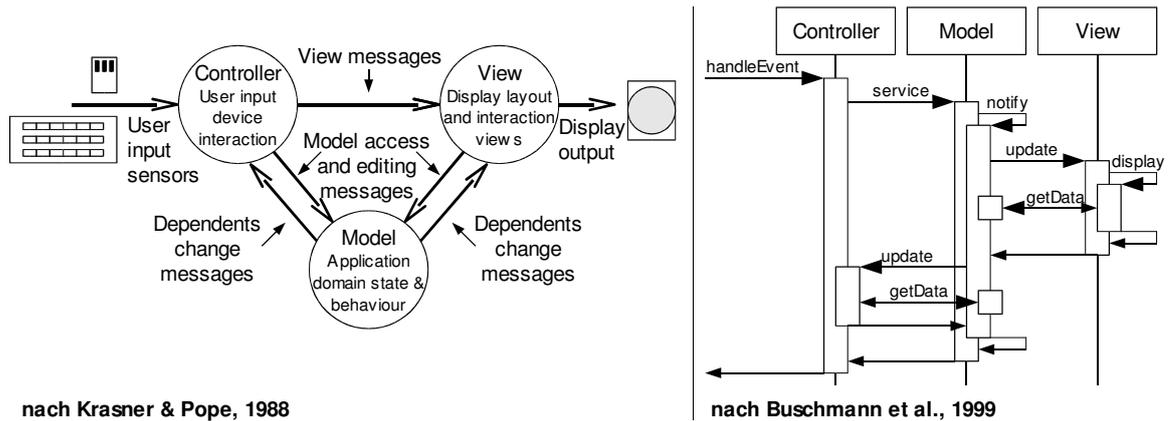


Abbildung 4.11: Model-View-Controller in der Literatur

Systembeschreibung.

Frühe Beschreibungen von MVC [Gol84, KP88, Bur92] hatten den Zweck, Programmierern des Smalltalk-80-Systems die Nutzung des darin erstmals enthaltenen MVC-Frameworks zu demonstrieren. Der Schwerpunkt dieser Beschreibungen liegt auf Beispielen, die zeigen, wie Programme unter Bezug auf das vorgegebene MVC-Gerüst und die zu diesem Zweck vordefinierten Typen zu schreiben sind. Die Struktur MVC-basierter Systeme soll eine leichte Abbildung interaktionsbezogener Modelle erlauben. Spätere Beschreibungen von MVC [BMR⁺99, BC92] abstrahieren von der Smalltalk-Implementierung und beschreiben Modelle, die gemeinhin mit MVC assoziiert werden, zur Entwicklung unabhängiger MVC-Realisierungen. Eine genaue Definition, was MVC ist, ist vor diesem Hintergrund problematisch. Dass für manche Entwickler MVC mittlerweile sogar zu einem Synonym für die getrennte Beschreibung von zweckbezogenem Kern und Interaktionswerk bei der Programmierung unter Nutzung vorgegebener GUI-Frameworks geworden ist, dürfte allerdings zu weit gehen.

Im einfachsten Fall besteht ein MVC-System aus einem Model, einem View und einem Controller, die miteinander in einer bestimmten Art und Weise interagieren und zusammen eine sogenannte MVC-Triade bilden. Angemessenerweise hat man dabei die Vorstellung, dass das Model eine Komponente des zweckbezogenen Kerns realisiert. Der View realisiert durch Ansteuerung eines Rastergrafiksystems einen Formadapter zur Ausgabe zweckbezogener Werte des Kerns. Der Controller realisiert in Verbindung mit den angeschlossenen Eingabegeräten einen Formadapter zur Identifikation zweckbezogener Eingaben. View und Controller sind nach gegenseitiger Registrierung quasi miteinander verschaltet. Beide sind als abhängige Objekte des Models registriert. Ausgehend von den Eingabeformen des Benutzers erkennt der Controller zweckbezogene Eingabewerte. Diese leitet er in Form einer Botschaft an das Model weiter, welches entsprechend darauf reagiert. Jede Zustandsänderung des Models wird mit dem Versenden einer Aktualisierungsbotschaft (*update message*) an alle registrierten Objekte abgeschlossen. Der View aktualisiert sich daraufhin selbstständig. Das Model muss über die Art und Weise der Darstellung nichts wissen. Dieses einfache Modell, dem die überschaubare Aufbauvorstellung einer isolierten MVC-Triade zugrunde liegt, vermitteln alle bekannten Beschreibungen von MVC (vgl. Abbildung 4.11).

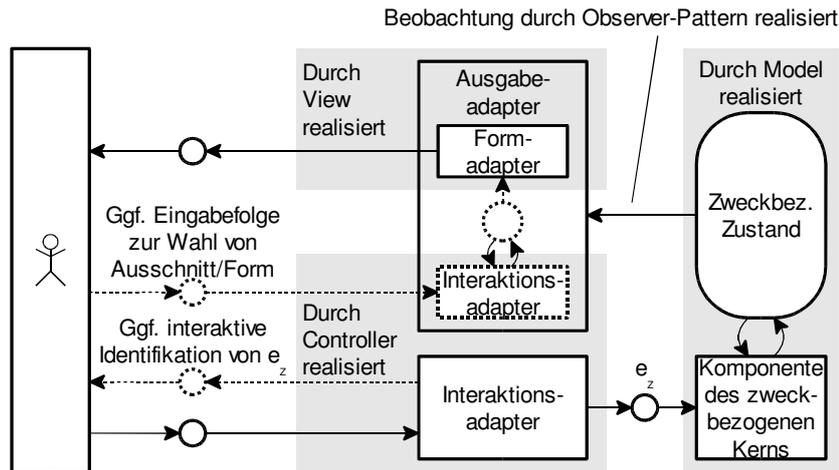


Abbildung 4.12: Zusammenhang von View, Controller und Interaktionsadaptern

Das zweckbezogene Modell früher MVC-basierter Systeme entspricht zumeist einem Steuerkreis. In diesen Fällen ist der durch Model-Komponenten realisierte zweckbezogene Kern rein operationeller Natur. Der Benutzer bildet zusammen mit dem Controller das Steuerwerk. Die Abfolge zweckbezogener Operationen wird dabei ausschließlich durch den Benutzer bestimmt. Definiert das zweckbezogene Modell einen bestimmten Ablauf, so wird durch MVC prinzipiell nicht verbindlich festgelegt, welche Komponente — Model oder Controller — den damit verbundenen zweckbezogenen Steuerzustand verwaltet und somit letztlich den Ablauf vorgibt.

Den Bezug zur Ebene interaktionsbezogener Modelle, wie in Abschnitt 3.4 vorgestellt, verdeutlicht Abbildung 4.12. Sie zeigt die MVC-basierte Realisierung einer zweckbezogenen Komponente, auf die über zwei Interaktionsadapter zur Identifikation von Ein- und Ausgaben zugegriffen wird. Alle Eingaben sind an den Controller gerichtet. Die Analyse zeigt, dass dabei Eingaben, die einen zweckbezogenen Wert identifizieren, von Eingaben unterschieden werden können, welche die gewünschte Anzeigeform identifizieren. Im Falle der interaktiven Identifikation zweckbezogener Eingaben mithilfe sogenannter Kontextmenüs gehört neben der Eingabeerkennung auch die Menüdarstellung zu den Aufgaben eines Controllers. Dieses Verhalten verträgt sich nicht mit der üblicherweise verbreiteten Vorstellung von einem Controller als einfachem Formwandler zur Erkennung zweckbezogener Werte. Es wird aber verständlich, wenn man Controllern potentiell die Rolle interaktiver Schrittfolgenadapter für zweckbezogene Eingaben zukommen lässt. Ebenso ist die Eingabebehandlung eines interaktiven Schrittfolgenadapter für Ausgaben durch einen Controller zu realisieren.

Ein MVC-System kann beliebig viele Models und View-Controller-Paare enthalten. Prinzipiell ist die Anzahl an View-Controller-Paaren, die sich bei einem Model registrieren können, unbegrenzt. Alle registrierten Objekte werden bei einer Zustandsänderung des Models benachrichtigt und können entsprechend darauf reagieren. Dabei ist, wie Abbildung 4.13 zeigt, nicht jedes Model als Komponente des zweckbezogenen Kerns anzusehen. Interaktionselemente, die als Bestandteil eines Interaktionsadapters anzusehen sind (vgl. Abschnitt 3.4.4), werden durch zusätzliche unterge-

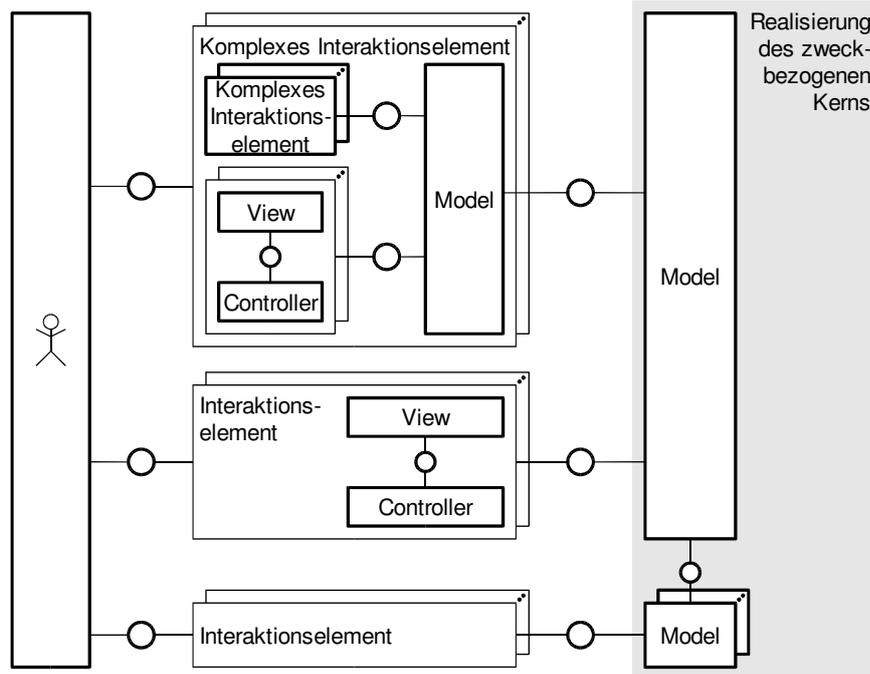


Abbildung 4.13: Abstrakte Sicht auf ein MVC-basiertes System

ordnete View–Controller–Paare realisiert. In manchen Fällen ist es zweckmäßig, den Dialogzustand eines Schrittfolgenadapters in einen steuernden und einen operationellen Teil aufzuteilen. In solchen Fällen ist es üblich, den operationellen Anteil durch ein eigenes Model zu realisieren. Dieses ist dann als Komponente des Schrittfolgenadapters anzusehen.

Durch die einschlägige Literatur bleibt die Frage unbeantwortet, ob und wenn ja welche Art der Kopplung zwischen den einzelnen MVC–Triaden existiert. Können mehrere MVC–Triaden zueinander nebenläufig interagieren? Welche Kopplung existiert zwischen Eingabegeräten und Controllern? Keine der Veröffentlichungen über MVC, die dem Autor bekannt sind, stellt ein Systemmodell vor, anhand dessen sich all diese Fragen zufriedenstellend beantworten ließen. [Bur92] geht zwar auf die Hierarchiebeziehung von Views bezüglich Bildschirmdarstellung und Datenstruktur ein, erklärt aber lediglich die Erzeugung einer solchen Hierarchie. Vergleicht man [KP88] und [BMR⁺99], so finden sich bezüglich der Kopplung zwischen Eingabegeräten und Controllern widersprüchliche Aussagen. Ein Modell, das mit MVC verträglich ist und anhand dessen sich die Fragen beantworten lassen, ist in Abbildung 4.14 und 4.15 gezeigt. Dieses Modell ist das Ergebnis einer Analyse, die der Autor ausgehend von einem Squeak–Smalltalk–System⁶ [Guz01] durchgeführt hat.

Entsprechend der Hierarchie von Interaktionselementen auf dem Bildschirm existiert eine Hierarchie von View–Controller–Paaren. Zu einem bestimmten Zeitpunkt ist ge-

⁶Squeak wurde 1995 geboren, als sich drei der Urväter von Smalltalk, Alan Kay, Dan Ingalls und Ted Kaehler, zusammenfanden um die ursprüngliche Idee des Dynabook von Alan Kay wiederzubeleben. Zur Analyse hat der Autor eine sehr frühe Version (1.2) von Squeak herangezogen, die ihren Ursprung in einer Originalportierung von Smalltalk–80 durch die Firma Apple hat.

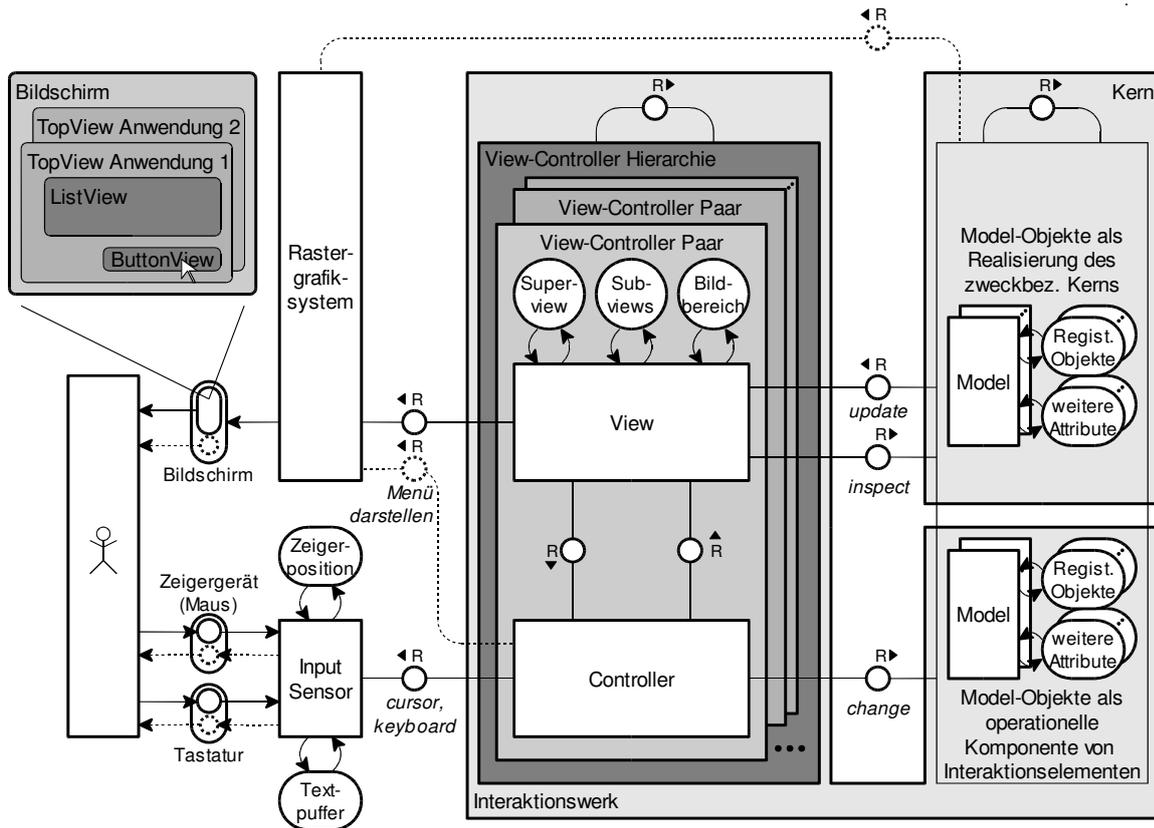


Abbildung 4.14: MVC-basierte Aufbaustruktur eines interaktiven Systems

nau der Controller aktiv, dessen assoziierter View den Bildschirmbereich verwaltet, in dem sich der Mauszeiger gerade befindet. Dieser Controller durchläuft eine Eingabebehandlungsschleife, in der er Tastatur- und Mauseingaben analysiert. Stellt er dabei fest, dass er aufgrund einer veränderten Zeigerposition nicht mehr für die Behandlung zuständig ist, gibt er die Kontrolle an seinen übergeordneten bzw. einen untergeordneten Controller ab. Die Schnittstellengeräte des interaktiven Systems werden auf diesem Wege im Multiplex genutzt.

Wird ein Controller zur Realisierung eines interaktiven Schrittfolgenadapters verwendet, ist der Controller dafür zuständig, die Systemausgaben zu veranlassen, die eine interaktive Identifikation des zweckbezogenen Wertes ermöglichen. Denkbar ist, dass ein Controller zu diesem Zweck ein untergeordnetes View-Controller-Paar aktiviert. Ein zu diesem Paar assoziiertes Model wäre als Bestandteil des Interaktionswerkes anzusehen. Wie bereits angedeutet, wird dieser Fall in den bekannten Veröffentlichung zu MVC weitgehend ausgespart bzw. steht er im Widerspruch zum üblichen Verständnis von Controllern. Tatsächlich zeigt die Analyse des untersuchten Squeak-Systems, dass im Falle einfacher Kontextmenüs die Darstellung durch den Controller selbst erfolgt. Die Analyse zeigt ebenfalls, dass die Anzeige von Modelzustandsdaten in Ausnahmefällen unter Umgehung des Views auch direkt über das Grafiksystem erfolgen kann. Ein View spielt in einem solchen Fall nur die Rolle eines Bildschirmausschnittsverwalters.

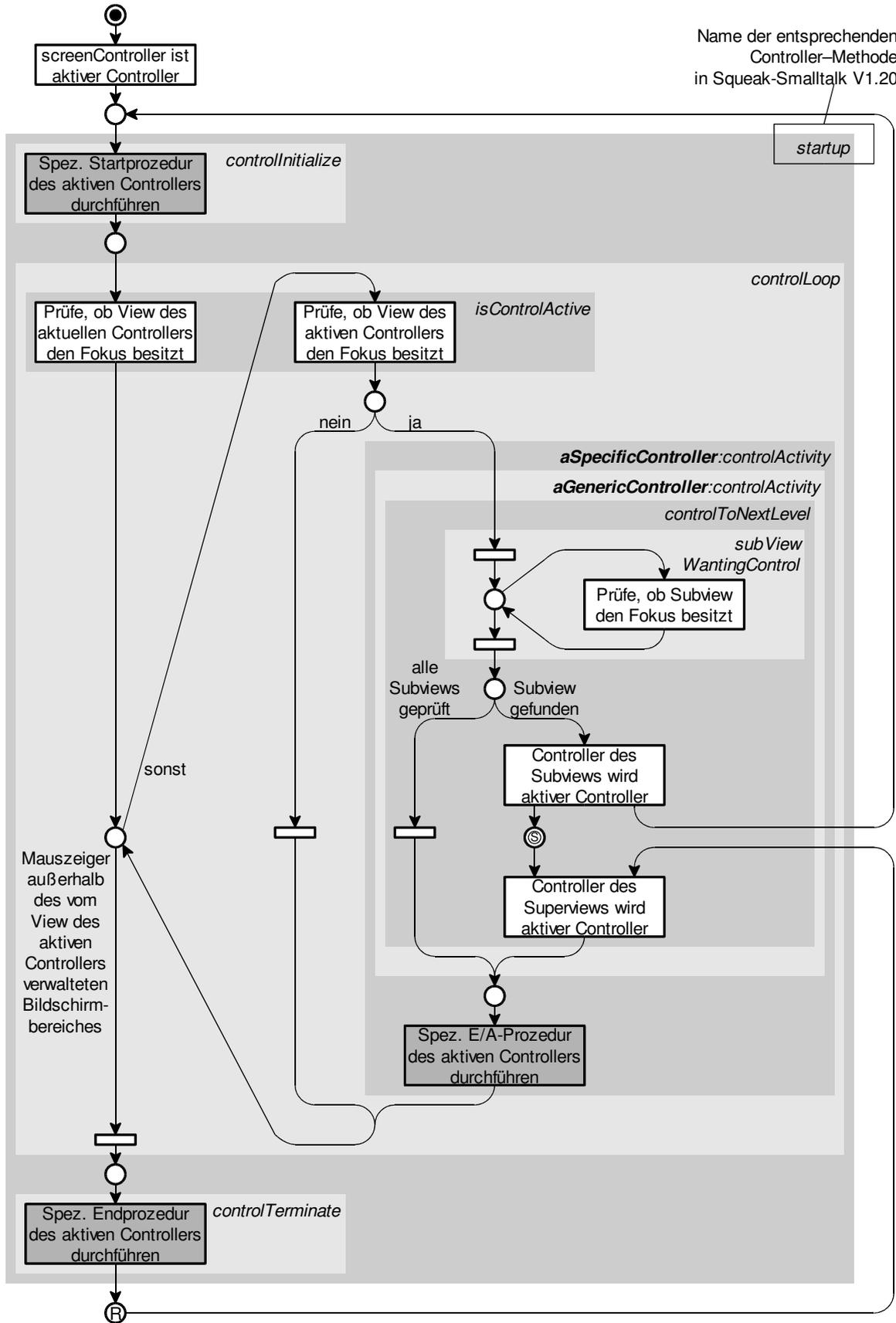


Abbildung 4.15: Zentraler Steuerablauf eines MVC-basierten Systems

Die Systemvorstellung, die in den Veröffentlichungen über MVC vermittelt wird, beschränkt sich überwiegend auf die Betrachtung einer isolierten MVC-Triade. Dabei steht die Beschreibung des Observer-Patterns, auf dem der Aktualisierungsmechanismus von View und Controller beruht, im Mittelpunkt. Davon ausgehend wird beschrieben, wie eine zweckmäßig Programmstrukturierung (basierend auf Klassen und Vererbungsbeziehungen) erfolgen kann. Dieses Wissen reicht aus, um interaktive Systeme unter Bezug auf bekannte Model-, View- und Controllertypen zu realisieren. Es reicht nicht aus, um selbstständig neue Typen zu beschreiben oder Vorhersagen über das Zeitverhalten des gesamten interaktiven Systems zu treffen.

4.3.2 Presentation–Application–Control (PAC)

Presentation–Application–Control ist die Bezeichnung für eine Art der Strukturierung interaktiver Systeme, auf der Ebene realisierungsbezogener Modelle, die erstmals in [Cou87] vorgestellt wurde. PAC-Modelle sind, wie MVC, Multiagent-Modelle.

Presentation (Präsentation), Application (Anwendung) und Control (Steuerung) sind die Namen der Typen unterschiedlicher Komponenten, aus denen ein interaktives System aufgebaut ist, das gemäß PAC strukturiert ist. Verglichen mit MVC ist PAC eher unbekannt. Dennoch kann man das zugrundeliegende Konzept in der Architektur vieler Systeme, zumindest ansatzweise, erkennen. Hierzu gehören beispielsweise Systeme, die auf dem recht verbreiteten GUI-Framework der Win32-API von Microsoft [MC02] basieren.

Ursprünglich wurde PAC als realisierungsbezogenes Konzept mit der Absicht die Kluft zwischen der „abstrakten Sphäre theoretischer Modelle und den praktischen Problemen beim Bau benutzerfreundlicher Benutzungsschnittstellen“ (Übersetzung des Autors aus [Cou87]) zu verringern, vorgestellt. Im Kern steht eine in Aufbaustrukturen verwurzelte Systemvorstellung. Einerseits kann damit ein direkter Bezug zur Ebene interaktionsbezogener Modelle hergestellt werden. Andererseits wird Antwort auf bestimmte realisierungsbezogene Fragen, wie etwa die Kopplung zwischen Kern und Interaktionswerk oder die Nutzung vorgegebener GUI-Frameworks, gegeben. Die Form der Beschreibung ist durch formlosen Text und freie Kommentargrafiken (vgl. Abbildung 4.16) geprägt. In späteren Beschreibungen des Konzeptes im Rahmen des AMODEUS-Projektes [Buc94] bilden PAC-Modelle lediglich eine Komponente des Interaktionswerkes im Zusammenspiel mit einem vorgegebenen GUI-Framework [Cou96]. In [BMR⁺99] wird die ursprüngliche Idee wieder aufgegriffen, nach der ein PAC-Modell ein interaktives System vollständig beschreibt. Der Schwerpunkt liegt nun auf der verbindlichen Festlegung von Interfaces und Klassenbeziehungen als Grundlage eines objekt-orientierten Programm-Designs.

Nach [Cou87] besteht jedes System, das gemäß PAC strukturiert ist, bei niedriger Auflösung aus drei Komponenten. Die Abstraction-Komponente „spiegelt Semantik der Anwendung wider“ (Übersetzung des Autors) und muss in diesem Sinne als Realisierung des zweckbezogener Kerns eines interaktiven Systems verstanden werden. Das Repertoire möglicher Operationen, die eine solche Komponente ausführen kann, wird im Rahmen einer vorab durchzuführenden Aufgabenanalyse (vgl. Abschnitt

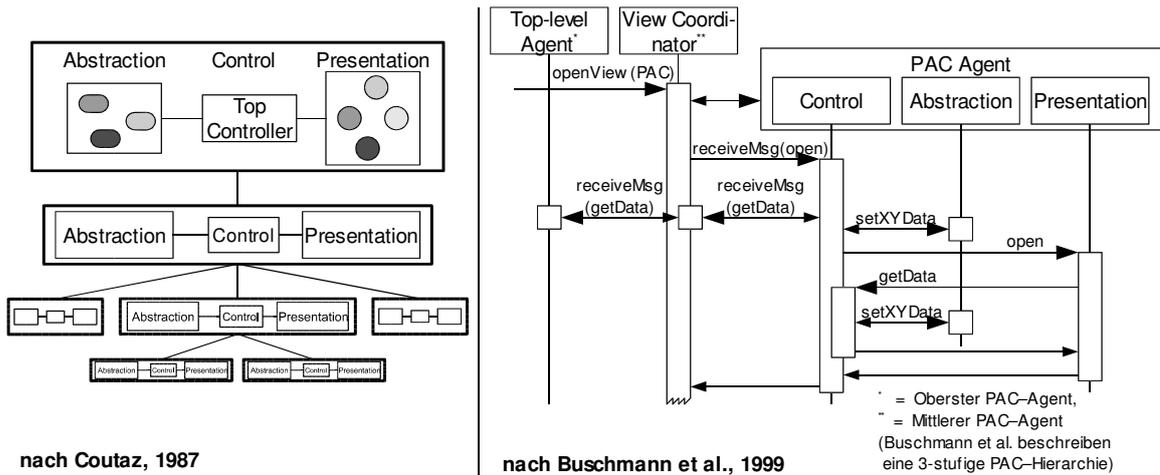


Abbildung 4.16: Presentation–Application–Control in der Literatur

4.2.2) bestimmt. Presentation– und Control–Komponente realisieren konzeptionell eine mehrstufige Abbildung zwischen den informationstragenden Formen an der Benutzungsschnittstelle und der internen Codierung zweckbezogener Ein- und Ausgaben des Kerns im Sinne von Abschnitt 3.4.3.5. Zusammen bilden sie das Interaktionswerk des Systems. Im einfachsten Falle realisiert die Presentation–Komponente einen Formadapter. Die Control–Komponente dient zur anwendungsspezifischen Steuerung von Presentation– und Abstraction–Komponente. Hierdurch ergibt sich die Struktur eines Steuerkreises. Presentation– und Abstraction–Komponente bilden dabei das Operationswerk. Dieser steuerkreisartige Aufbau wird unabhängig davon realisiert, ob die Identifikation zweckbezogener Werte ein- oder mehrschrittig erfolgt. Jede Eingabe und Ausgabe des System wird durch die Control–Komponente, abhängig einem dem durch sie verwalteten Dialogzustand, veranlasst. Eine unmittelbare — quasi automatische — Aktualisierung der Presentation–Komponente, wie bei MVC, gibt es nicht.

Unklar bleibt die in [Cou87] beschriebene Zuständigkeit von Controls für die Verwaltung benutzerunterstützender Informationen wie Hilfstexte oder Erklärungen. Die Einheit aus einer Presentation–, einer Control– und einer Abstraction–Komponente wird als PAC–Agent bezeichnet.

Eine höhere Auflösung des Modells zeigt, dass eine Presentation–Komponente aus mehreren Interaktionselementen bestehen kann, die untereinander nicht gekoppelt sind und nebenläufig zueinander agieren können. Jedes dieser Interaktionselemente ist entweder seinerseits als PAC–Agent realisiert oder als sogenanntes elementares Interaktionsobjekt, das auf einem GUI–Framework basiert, welches als gegebenes vorausgesetzt wird. Typische Beispiele für elementare Interaktionsobjekte sind Knöpfe, Menüs oder Textfelder. Das gesamte System erhält die Struktur eines hierarchischen Steuerkreises. Beispielsweise könnte die Realisierung eines interaktiven Schrittfolgenadapters durch einen untergeordneten PAC–Agenten erfolgen. Die Application–Komponente solcher PAC–Agenten dient dann zur Verwaltung des operationellen Anteils des Dialogzustandes, wie sie beispielsweise für einen Formularausschnitt zur schrittweisen Erfassung von Adresdaten durch ein Bestellsystem erforderlich ist. Eine

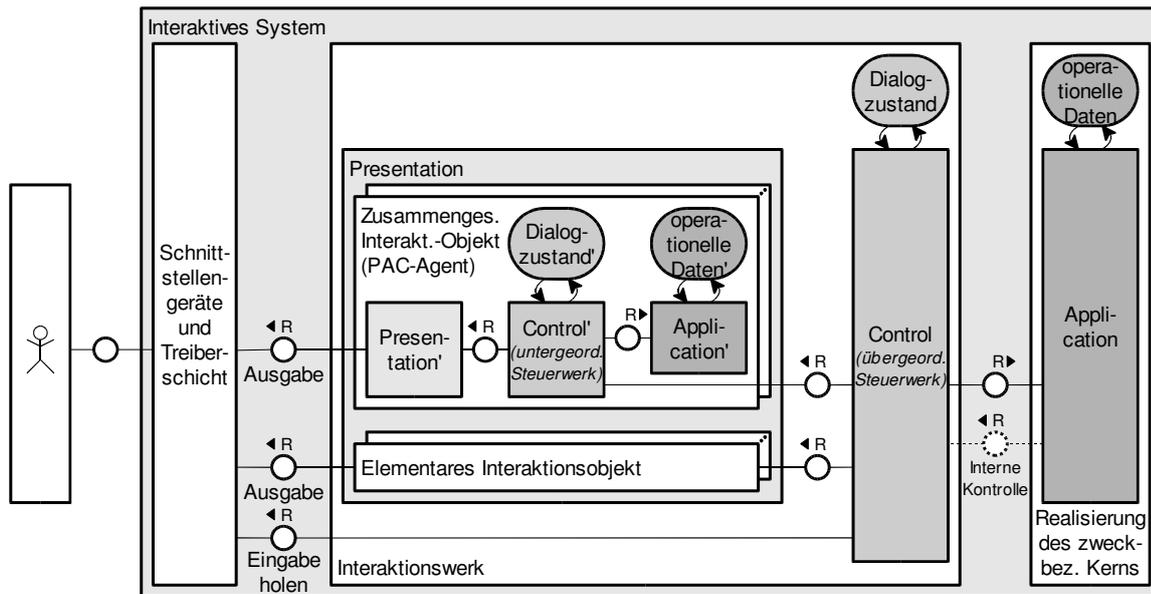


Abbildung 4.17: PAC-basierte Aufbaustruktur eines interaktiven Systems

Verfeinerung von Control oder Application ist mit der Erhöhung der Auflösung nicht verbunden.

Die Behandlung von Eingaben wirft in Verbindung mit der Nutzung elementarer Interaktionsobjekte auf der Basis beliebiger GUI-Frameworks die Frage der Koppelung zwischen Control-Komponenten und elementaren Interaktionskomponenten auf. In [Cou87] wird hierzu am Rande erwähnt, dass es Aufgabe der obersten Control-Komponente ist, alle elementaren Eingabeereignisse zu erfassen und gegebenenfalls an eine entsprechende Control-Komponente zu delegieren. Eine solche zentrale Ereignisbehandlungsschleife ist der Kern vieler interaktiver Systeme. Spätere Veröffentlichungen vernachlässigen diesen für das Systemverständnis wichtigen Aspekt vollständig.

Im Hinblick auf die Entwicklung interaktiver Systeme unter Verwendung eines vorgegebenen GUI-Frameworks erscheint PAC als ein Konzept, welches das Denken von Systemarchitekten und Entwicklern praxisnah lenken kann. Als wenig verständnisfördernd muss die Art der Darstellung in der Literatur bezeichnet werden. Vor dem Hintergrund der mit PAC verbundenen aufbauleitenden Vorstellung von einem interaktiven System erscheint die Beschränkung auf einen monolithischen zweckbezogenen Kern unverständlich. Die alternative Vorstellung von mehreren Application-Komponenten zur Realisierung des zweckbezogenen Kerns, deren zentrale Steuerung durch ein übergeordnetes Control rein technisch (und nicht durch den Systemzweck) begründet wäre, ist ein denkbarer Ausweg.

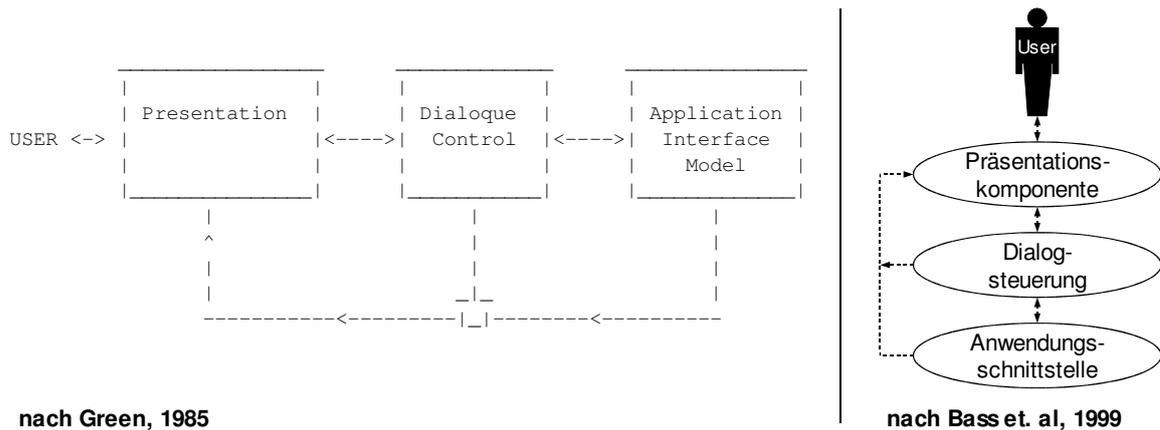


Abbildung 4.18: Das Seeheim-Modell in der Literatur

4.3.3 Seeheim-Modell

Als Seeheim-Modell wird im Allgemeinen das in [Gre85] beschriebene Modell eines sogenannten User Interface Management Systems (UIMS) angesehen⁷. Ein UIMS ist eine besondere Form von Trägersystem zur Realisierung des Interaktionswerkes eines interaktiven Systems, dessen Struktur die arbeitsteilige Systementwicklung durch Benutzungsschnittstellendesigner und Anwendungsentwickler fördert. Das Seeheim-Modell ist somit zur Ebene realisierungsbezogener Modelle zu rechnen.

Zweck der ursprünglichen Darstellung war — ausgehend von einer aufbaubasierten Vorstellung — die Erläuterung der prinzipiellen Funktionsweise eines UIMS. [Gre85] beschreibt kein fertiges Modell, sondern er identifiziert funktionale Zuständigkeiten und Zusammenhänge. In gleichem Maße weist er bewusst auf offene Fragen und abzuwägende Realisierungsentscheidungen hin. Die Beschreibung ist als Gerüst eines Systemmodells zu verstehen, das einen bestimmten Wissensstand dokumentiert und einen Bezug zu ungeklärten Fragen herstellt. In späteren Veröffentlichungen [BC92, Sta96, BCK98] wird das Seeheim-Modell trotz der recht ausführlichen Beschreibung im ursprünglichen Text im Wesentlichen auf die in [Gre85] abgebildete dreiteilige Blockstruktur reduziert (vgl. Abbildung 4.18). Diese verkürzte Darstellung dient häufig nur als historischer Ausgangspunkt für eine Erklärung von Architekturstilen wie MVC oder PAC.

Die Abbildung des Seeheim-Modells zeigt drei Komponenten eines interaktiven Systems. Aufgabe der Präsentationskomponente ist die Bereitstellung elementarer universell einsetzbarer Interaktionselemente. Aufgabe der Dialogsteuerung ist die Aktivierung bzw. Deaktivierung sowie die Koordinierung der Interaktionselemente zur Realisierung von Interaktionsadaptern höherer Semantik. Die Kopplung dieser zentralen

⁷Der Name leitet sich von der Ortschaft Seeheim bei Darmstadt in Deutschland ab. Auf der dortigen Tagung zum Thema „User Interface Management Systems“ im Jahre 1983 wurde von M. Green als Ergebnis der Arbeitsgruppe „Dialogspezifikationswerkzeuge“ das mittlerweile als Seeheim-Modell bekannte Modell vorgestellt. Andere im Tagungsband [Pfa85] beschriebene Modelle, wie die von Strubbe, Enderle oder Airchinnigh sollten daneben jedoch nicht vergessen werden.

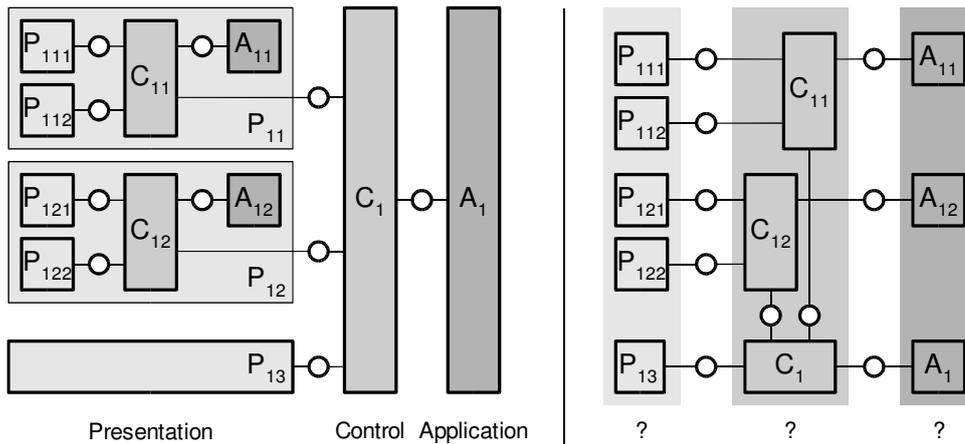


Abbildung 4.19: PAC, keine Realisierung des Seeheim-Modells

Komponente mit dem zweckbezogenen Kern ist mithilfe einer dritten Komponente, dem sogenannten Application Interface Model, realisiert.

Die grafische Darstellung in [Gre85] sowie spätere Veröffentlichungen [BCK98, Sta96] suggerieren dabei eine Identität von Application Interface Model und der Realisierung des zweckbezogenen Kerns. Folgt man diesem Ansatz, lässt sich, wie Abbildung 4.19 zeigt, scheinbar jedes PAC-Modell als Verfeinerung des Seeheim-Modells ansehen.

Folgt man hingegen der verbalen Beschreibung aus [Gre85], so ergibt sich die in Abbildung 4.20 gezeigte Aufbaustruktur. Es ist die Struktur eines Trägersystems zur Realisierung eines Interaktionswerkes, dessen Programmierung mit der Erstellung von Beschreibungseinheiten unterschiedlichen Typs verbunden ist. Diese Einheiten dienen der Umschreibung von Systemstrukturen auf der Ebene interaktionsbezogener Modelle.

Die Präsentationskomponente hat Zugriff auf ein sogenanntes Dictionary, das eine Beschreibung des Repertoires elementarer Interaktionselemente enthält, die unter Verwendung der Schnittstellengeräte des Systems realisierbar sind. Nach Green legt ein solcher Typ die Abbildung zwischen Eingabe- bzw. Ausgabetokens und Symbolen an der Benutzungsschnittstelle fest. Die Form dieser Symbole bzw. der Interaktionselemente wird durch einen Schnittstellendesigner abstrakt beschrieben („high-level description“). Die Struktur der Benutzungsschnittstelle als Komposition verschiedener Interaktionselemente in unterschiedlichen Zuständen ergibt sich zu einem bestimmten Zeitpunkt aus der aktuellen Konfiguration der Präsentationskomponente.

Die Dialogsteuerung führt die Abbildung zwischen zweckbezogenen Ein- und Ausgabentoken und elementaren Eingabe- und Ausgabentoken durch und realisiert so den Kern eines oder mehrerer Interaktionsadapter. Je nach Anwendung kann sie die Rolle eines übergeordneten Steuerwerkes übernehmen oder im Falle der sogenannten internen Kontrolle durch den zweckbezogenen Kern eine untergeordnete operationelle Rolle übernehmen. Ihr Verhalten ist nach Green durch eine Grammatik, einen Zustandsgraphen oder ein sogenanntes Ereignis-Behandler-Modell (engl. event model) beschreibbar.

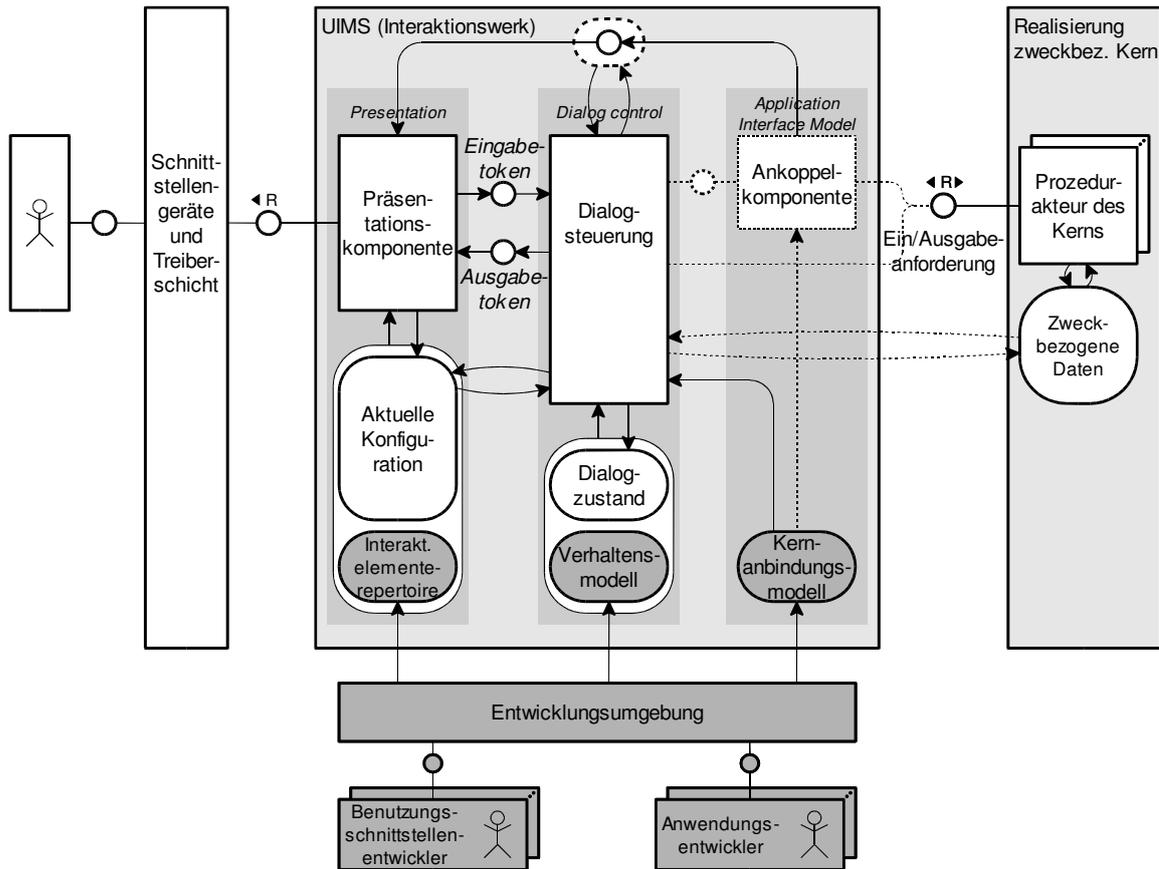


Abbildung 4.20: Aufbaustruktur eines UIMS-basierten Systems nach Seeheim

Der zweckbezogene Kern ist durch eine Anzahl von Akteuren realisiert, deren Verhalten durch programmierte Prozeduren beschrieben wird. Diese Akteure besitzen Zugriff auf Datenstrukturen, welche eine Codierung der zweckbezogenen Daten darstellen. Der Zugriff auf den Kern des System wird durch das sogenannte Application Interface Model unterstützt. Dessen zentraler Bestandteil ist eine Beschreibung des Wissens, welches die Dialogsteuerung für ihren Zugriff auf die Komponenten des Kerns benötigt und das hier als Kernanbindungsmodell bezeichnet werden soll. Als Bestandteile einer solchen Beschreibung nennt Green unter anderem die Signaturen von Prozeduren, die zur Realisierung des zweckbezogenen Kerns dienen, Vor- und Nachbedingungen für deren Aufruf sowie Bezeichner von direkt zugänglichen Datenstrukturen des Kerns.

Zusätzlich wird dem Application Interface Model eine aktive Rolle zugeschrieben. In Abbildung 4.20 äußert sich dies in der Existenz einer sogenannten Ankoppelkomponente. So soll es dieser Komponente möglich sein, über einen Kanal, der von der Dialogsteuerung eingerichtet werden kann, direkt mit der Präsentationskomponente zu kommunizieren. Zur Visualisierung von Informationsflüssen mit einer hohen Datentransferrate, die keiner zusätzlichen Behandlung durch die Dialogsteuerung bedürfen (wie beispielsweise zur Darstellung von Videos) kann dies zweckmäßig sein. In diesem Zusammenhang bleibt unklar, inwieweit die „von außen“ nutzbaren Zugriffsprozedu-

ren des Kerns nicht möglicherweise als Bestandteil des Application Interface Model anzusehen sind. Weiterhin wird vorgeschlagen, dass die Komponente in der Lage sein soll, die Aufgaben eines Interaktionsassistenten zu übernehmen.

Green beschreibt mit dem Application Interface Model eine Komponente, die in den bis dahin existierenden UIM-Systemen nicht als eigenständige Komponente vorkam, und stellt deren Notwendigkeit fest. Seine Beschreibung ist keineswegs als verbindliche Spezifikation interpretierbar, sondern sie ist vielmehr als Gerüst für zukünftige Entwicklungen zu verstehen.

Von Green wird das Modell eines Interaktionswerkes beschrieben, welches aus drei Komponenten besteht und im Zusammenspiel mit dem Kern und der Geräte- und Treiberschicht betrachtet wird. In diesem Sinne ähnelt das Seeheim-Modell dem späteren Arch-Slinky-Modell [Wor92], in dem fünf vergleichbare Ebenen eines interaktiven Systems unterschieden werden. Ein direkter Zusammenhang zu PAC-AMODEUS [Cou96] folgt daraus, wenn man das Verhaltensmodell der Dialogsteuerung durch ein Ereignis-Behandler-Modell in Form einer PAC-Struktur spezifiziert.

Es ist offensichtlich, dass in späteren Veröffentlichungen das Seeheim-Modell auf die vereinfachte, grafische Blockstruktur der Abbildung von Green reduziert wird. Der Gedanke eines UIMS als Trägersystem für ein Interaktionswerk kommt darin nicht zum Ausdruck. Gravierender ist der häufig erkennbare Fehler, die von Green explizit als „logisches Modell eines UIMS“ bezeichnete Struktur mit dem gesamten interaktiven System gleichzusetzen. Der Autor vermutet als Ursache, die tragende Rolle von Grafik bei der Vermittlung von Systemvorstellungen. Eine einprägsame Grafik wird hier gleichsam zum Kondensat einer falschen Systemvorstellung. Man kann Green nicht vorwerfen, dass er das Modell verbal nicht verständlich beschrieben habe — im Gegenteil. Allerdings zeigt seine Grafik nur einen Ausschnitt, eine Komponente, eines interaktiven Systems und dies in stark vereinfachter Form. Es ist unwahrscheinlich, dass ein Verzicht auf diese Grafik, die Verständlichkeit der ursprünglichen Beschreibung des Seeheim-Modelles erhöht hätte. Die hohe Dominanz und Einprägsamkeit von Grafiken lässt vielmehr vermuten, dass ein Verzicht auf diese Grafik den heutigen Bekanntheitsgrad des Seeheim-Modells deutlich gemindert hätte. Nach Ansicht des Autors sollte daher das Wissen um die Gefahren fehlerhaft interpretierbarer, bestenfalls nutzloser Grafiken [Pet95] zu einer sorgfältigeren Gestaltung von Systemplänen führen.

Kapitel 5

Fazit und Ausblick

Die in dieser Arbeit entwickelte und konsequent in der Aufbauvorstellung verwurzelte Begriffswelt zur Beschreibung interaktiver Systeme ist geeignet, bestehende Beschreibungs- und Entwicklungsansätze sinnvoll zu ergänzen. Mit ihrer Hilfe ist es möglich, eine Verbindung zwischen der Vielzahl unterschiedlicher Beschreibungsansätze herzustellen. Zusätzlich erlaubt sie, neue Konzepte interaktiver Systeme einzuordnen.

Es konnte gezeigt werden, dass den meisten der untersuchten Beschreibungsansätze eine Aufbauvorstellung zugrundeliegt, von der ausgehend das Systemverhalten zu beschreiben ist oder beschrieben wurde. Allerdings handelt es sich auf der Ebene zweckbezogener und interaktionsbezogener Modelle zumeist um einen Trivialaufbau aus Benutzer und interaktivem System. Die Darstellung von Aufbaustrukturen erfolgt zumeist nur implizit. Auf der Ebene realisierungsbezogener Modelle hingegen spiegeln alle dem Autor bekannten Beschreibungsansätze eine innere Aufbaustruktur eines interaktiven Systems wider. Unklarheiten in der Abgrenzung von Aufbau und Verhalten, Beschreibung und System oder in der Abgrenzung von Modellebenen erschweren in vielen Fällen das Verständnis. Die vorgefundenen grafischen Modelldarstellungen sind im Hinblick auf die Vermittlung einer angemessenen Systemvorstellung als unzureichend zu betrachten. Defizite in Struktur und Layout der Darstellungen müssen als Ursache für Kommunikationsprobleme angesehen werden.

Die vorgestellte Begriffswelt erlaubt eine verständliche und kompakte Darstellung aller untersuchten Beschreibungsansätze. Die Zusammenhänge zwischen angrenzenden Modellebenen werden erklärt. Die Findung und explizite Darstellung von Systemstrukturen mit den Mitteln der Fundamental Modelling Concepts bildet dafür eine unverzichtbare Basis.

Primärer Zweck der Begriffswelt ist es, ein gedankliches Gerüst bereitzustellen, das die Vorstellung von Personen leitet, die an der Entwicklung eines interaktiven Systems beteiligt sind. Die grafische Darstellung kann ein wichtiges Hilfsmittel zur Vermittlung gefundener Systemstrukturen sein. Sie sollte jedoch nicht als Kern dieser Arbeit verstanden werden. Nach Ansicht des Autors ist es grundsätzlich hilfreich, die für das Verständnis einer Betrachtungsebene entscheidenden Grundstrukturen darzustellen. Stets sind dabei jedoch Nutzen und Aufwand der Darstellung gegeneinander abzuwägen.

Eine praktische Anwendung der Begriffswelt und Darstellungsweise macht nach Ansicht des Autors folgende Auswirkungen wahrscheinlich:

- In den frühen Phasen der Findung zweckbezogener und interaktionsbezogener Modelle lässt sich in der Kommunikation von Systementwicklern mit Kunden, zukünftigen Anwendern und Benutzern schneller als bisher eine gemeinsame Systemvorstellung entwickeln.
- Die Vermittlung einführender Systemmodelle durch Bedienungsanleitungen führt effektiver und effizienter zu einem Systemverständnis und so zu einer besseren Nutzbarkeit interaktiver Systeme.
- Basierend auf den Konzepten von Interaktionsadaptern und Interaktionsassistenten lässt sich die Idee einer benutzergerechten Anpassung von Benutzungsschnittstellen durch Austausch, Hinzunahme oder Entfernen von Komponenten kommunizieren und realisieren. Die Unterscheidung von Interaktionsadaptern und Assistenten könnte sowohl die Gestaltung übersichtlicher Benutzungsschnittstellen als auch die Verbesserung einer individuellen Benutzerunterstützung fördern.
- Auf Ebene realisierungsbezogener Modelle kann eine geeignete Dokumentation die Vermittlung der häufig nur gedachten Brücke zwischen interaktions- und realisierungsbezogenen Modellen und damit die Nutzung von Realisierungsplattformen (wie etwa JSP-Servern) vereinfachen.

Um den Nutzen dieser Vorschläge für die Praxis objektiv beurteilen zu können, sind weitere Arbeiten erforderlich. Denkbare Forschungsgebiete sind:

- Studien zur Ermittlung von Aufwand und Nutzen bei der Entwicklung interaktiver Systeme im Hinblick auf eine konkrete Einbettung in den Entwicklungsprozess
- Studien zur Erlernbarkeit und Akzeptanz der Systemmodelle bei Kunden, Anwendern und Benutzern

Literaturverzeichnis

- [Azu97] AZUMA, Ronald T.: *A survey of augmented reality*. 1997. – <http://citeseer.ist.psu.edu/article/azuma97survey.html>
- [Bal00] BALZERT, Helmut: *Lehrbuch der Software-Technik*. 2. Spektrum Akademischer Verlag, Heidelberg, 2000. – ISBN 3827400422
- [Bar93] BARNARD, Philip J.: Modelling Users, Systems and Design Spaces (Esprit Basic Research Action 3066). In: *HCI '93*, Elsevier, 1993, S. 331–342
- [BC92] BASS, Len ; COUTAZ, Joëlle: *Developing Software for the User Interface*. 2. Addison-Wesley Publishing Company, 1992. – ISBN 0–201–51046–4
- [BCK98] BASS, Len ; CLEMENTS, Paul ; KAZMAN, Rick: *Software Architecture in Practice*. Addison-Wesley, 1998. – ISBN 0–201–19930–0
- [BK82] BORUFKA, H. G. ; KUHLMANN, Herbert W.: Dialogue Cells: a Method for Defining Interactions. In: *IEEE-CGA 2* (1982), July, Nr. 5, S. 25–27, 29–30, 32–33
- [BMR⁺99] BUSCHMANN, Frank ; MEUNIER, Regine ; ROHNERT, Hans ; SOMMERLAD, Peter ; STAL, Michael: *Pattern-oriented Software Architecture - A System of Patterns*. John Wiley & Sons, 1999. – ISBN 0–471–95869–7
- [BMW94] BIGGERSTAFF, Ted J. ; MITBANDER, Bharat G. ; WEBSTER, Dallas E.: Program Understanding and the Concept Assignment Problem. In: *Communications of the ACM* 37 (1994), May, Nr. 5, S. 72–83
- [BRJ99] BOOCH, Grady ; RUMBAUGH, James ; JACOBSON, Ivar: *The unified modeling language user guide*. Addison Wesley, 1999. – ISBN 0–201–57168–4
- [Buc94] SIMON BUCKINGSHAM, Nick Hammond Annette A. (Hrsg.). *AMODEUS-2 HCI Modelling and Design Approaches: Executive Summaries and Worked Examples*. 1994
- [Bun98] BUNGERT, Andreas: *Beschreibung programmierter Systeme mittels Hierarchien intuitiv verständlicher Modelle*. Shaker Verlag, Juni 1998. – ISBN 3–8265–3911–7
- [Bur92] BURBECK, Steve: *Applications Programming in Smalltalk-80: How to use Model-View-Controller (MVC)*. 1992. – <http://st-www.cs.uiuc.edu/users/smarch/st-docs/mvc.html>

- [CCN97] CALVARY, Gaelle ; COUTAZ, Joëlle ; NIGAY, Laurance: From Single-User Architectural Design to PAC*: a Generic Software Architecture Model for CSCW. In: *CHI*, 1997, S. 242–249
- [CMN80] CARD, Stuart K. ; MORAN, Thomas P. ; NEWELL, Allen: The Keystroke-Level Model for User Performance Time with Interactive Systems. In: *Communications of the ACM* 23 (1980), July, Nr. 7, S. 396–410
- [Cou87] COUTAZ, Joëlle: PAC: An Object Oriented Model for Dialog Design. In: H.-J. BULLINGER, B. S. (Hrsg.): *Human-Computer Interaction: INTERACT'87*. Amsterdam : North-Holland, 1987, S. 431–436
- [Cou96] COUTAZ, Joëlle: PAC-Based Software Architecture Modelling for Interactive Systems. In: *Softwaretechnik-Trends* 16 (1996), September, Nr. 4, S. 4–11
- [CY91] COAD, Peter ; YOURDON, Edward: *Object Oriented Analysis*. 2. Yourdon Press, 1991. – ISBN 0–13–629981–4
- [DeM78] DEMARCO, Tom: *Structured analysis and system specification*. Yourdon Press, 1978. – ISBN 0–13–854380–1
- [Dew95] DEWAN, Prasun: Multiuser architectures. In: BASS, Leonard J. (Hrsg.) ; UNGER, Claus (Hrsg.): *Engineering for Human-Computer Interaction, Proceedings of the IFIP TC2/WG2.7 Working Conference on Engineering for Human-Computer Interaction*, Chapman and Hall, August 1995, S. 247–270
- [DIN93] DIN: DIN66341 - Grafische Datenverarbeitung - Referenzmodell / Deutsches Institut für Normung e.V. 1993. – Deutsche Norm - identisch mit ISO/IEC 11072:1992
- [Dob00] DOBELLE, William H.: Artificial Vision for the Blind by Connecting a Television Camera to the Brain. In: *ASAIO Journal* (2000), S. 3–9
- [Dro74] DROSDOWSKI, Günther (Hrsg.): *Der Duden: in 10 Bd.; d. Standardwerk zur dt. Sprache, Das Fremdwörterbuch*. Bd. 5. 3. Dudenverlag, 1974. – ISBN 3–411–00915–2
- [Dro89] DROSDOWSKI, Günther (Hrsg.): *Der Duden: in 10 Bd.; d. Standardwerk zur dt. Sprache, Das Herkunftswörterbuch, Etymologie der deutschen Sprache*. Bd. 7. 2. Dudenverlag, 1989. – ISBN 3–411–20907–0
- [Dzi83] DZIDA, Wolfgang: Das IFIP-Modell für Benutzerschnittstellen. In: *Office Management* 31 (1983), S. 6–8
- [FD82] FOLEY, James D. ; VAN DAM, Andries: *Fundamentals of interactive computer graphics*. 1. Addison Wesley, Reading, Massachusetts, 1982. – ISBN 0–201–14468–9
- [Gal98] GALES, Frank: *Ein Beitrag zur Begriffswelt der Schaffung programmierter Systeme*. Shaker Verlag, 1998. – ISBN 3–8265–3925–7

- [GG96] GENTNER, Don ; GRUDIN, Jonathan: Design Models for Computer-Human-Interfaces. In: *IEEE Computer* 29 (1996), June, Nr. 6, S. 28–35
- [Gla02] GLASS, Robert L.: The Naturalness of Object Orientation: Beating a Dead Horse? In: *IEEE Software* 19 (2002), May/June, Nr. 3, S. 104, 103
- [Gol84] GOLDBERG, Adele: *Smalltalk-80, the interactive programming environment*. Addison Wesley, 1984. – ISBN 0–201–11372–4
- [Gre85] GREEN, Mark: Report on Dialogue Specification Tools. In: PFAFF, Guenther E. (Hrsg.): *User Interface Management Systems*, Springer, Berlin, 1985. – ISBN 0–387–13803–X, S. 9–20
- [Guz01] GUZDIAL, Mark: *Squeak - Object-Oriented Design with Multimedia Applications*. Prentice Hall, 2001. – ISBN 0–13–028028–3
- [Har88] HAREL, David: On Visual Formalisms. In: *Communications of the ACM* 31 (1988), May, Nr. 5, S. 514–530
- [Har01] VAN HARMELEN, Mark (Hrsg.): *Object modeling and user interface design*. Addison Wesley, 2001. – ISBN 0–201–65789–9
- [Hat98] HATTON, Les: Does OO Sync with How We Think? In: *IEEE Software* 15 (1998), May/June, Nr. 3, S. 46–54
- [HHN86] HUTCHINS, Edwin L. ; HOLLAN, James D. ; NORMAN, Donald A.: Direct Manipulation Interfaces. In: DONALD A. NORMAN, Stephen W. D. (Hrsg.): *User Centered System Design*. Hillsdale, New Jersey : Lawrence Erlbaum Associates, 1986. – ISBN 0–89859–782–9, S. 87–124
- [ICS03] ICSE2003 (Hrsg.): *ICSE'03 International Conference on Software Engineering, Bridging the Gaps Between Software Engineering and Human-Computer Interaction*. 2003
- [JK96] JOHN, Bonnie E. ; KIERAS, David E.: The GOMS Family of User Interface Analysis Techniques: Comparison and Contrast. In: *ACM Transactions on Computer-Human Interaction* 3 (1996), December, Nr. 4, S. 320–351
- [Joh92] JOHNSON, Peter: *Human computer interaction: psychology, task analysis and software engineering*. McGraw-Hill International (UK) Limited., 1992. – ISBN 0–07–707235–9
- [JWMP93] JOHNSON, Peter ; WILSON, Stephanie ; MARKOPOULOS, Panos ; PYCOCK, James: ADEPT - Advanced Design Environment for Prototyping with Task Models. In: STACY ASHLUND, A. Henderson E. Hollnagel T. W. (Hrsg.): *Proceedings of ACM Conf. Human Factors in Computing Systems, INTERCHI*, ACM Press, 1993. – ISBN 0897915755, S. 24–29
- [KA92] KIRWAN, Barry ; AINSWORTH, Les: *A Guide to Task Analysis*. Taylor and Francis, 1992. – ISBN 0–7484–0057–5

- [KAB01] KRUCHTEN, Philippe ; AHLQVIST, Stefan ; BYLUND, Stefan: User Interface Design in the Rational Unified Process. In: VAN HARMELEN, Mark (Hrsg.): *Object modeling and user interface design*. Addison Wesley, 2001. – ISBN 0–201–65789–9, S. 161–196
- [Kam85] KAMRAN, Abid: Issues pertaining to the Design of a User Interface Management System. In: PFAFF, Guenther E. (Hrsg.): *User Interface Management Systems*, Springer, Berlin, 1985. – ISBN 0–387–13803–X, S. 9–20
- [Kel03] KELLER, Frank: Über die Rolle von Architekturbeschreibungen im Software-Entwicklungsprozess / Hasso-Plattner Institute for IT-Systems Engineering. 2003. – PhD. Theses
- [Kle99] KLEIS, Wolfram: *Konzepte zur verständlichen Beschreibung objektorientierter Frameworks*. Shaker Verlag, 1999. – ISBN 3–8265–6754–4
- [KP88] KRASNER, Glenn E. ; POPE, Stephen P.: A Description of the Model-View-Controller User Interface Paradigm in the Smalltalk-80 system. In: *Journal of object-oriented programming* 1 (1988), August/September, Nr. 3, S. 26–49
- [MA01] MCGINNES, Simon ; AMOS, Johnny: Accelerated Business Concept Modeling. In: VAN HARMELEN, Mark (Hrsg.): *Object modeling and user interface design*. Addison Wesley, 2001. – ISBN 0–201–65789–9, S. 3–36
- [MC02] MICROSOFT-CORPORATION: *Platform SDK Documentation - Win32 API*. 2002. – <http://msdn.microsoft.com/library/default.asp>
- [Mil56] MILLER, George A.: The magical number seven, plus or minus two: Some limits on our capacity of processing information. In: *The psychological review* 63 (1956), S. 81–97
- [Mor38] MORRIS, Charles W.: *Foundations of the theory of signs*. Chicago, Ill. : The University of Chicago Press, 1938
- [Mor81] MORAN, Thomas P.: The Command Language Grammar: a representation for the user interface of interactive computer systems. In: *International journal of man machine studies* 15 (1981), S. 3–50
- [Moy96] MOYNIHAN, Tony: An Experimental Comparison of Object-Orientation and Functional- Decomposition as Paradigms for Communicating System Funtionality to Users. In: *Journal of Systems and Software* 33 (1996), May, Nr. 2, S. 163 – 169
- [Nor01] NORMAN, Donald A.: *The Design of Everyday Things*. MIT Press, 2001. – ISBN 0–262–64037–6
- [Par69] PARNAS, David L.: On the use of transition diagram in the design of a user interface for an interactive computer system. In: *Proceedings of the 24th ACM Conference*, 1969, S. 379–385

- [Par72] PARNAS, David L.: On the criteria to be used in decomposing systems into modules. In: *Communications of the ACM* 15 (1972), December, Nr. 12, S. 1053–1058. – ISSN 0001–0782
- [Pet95] PETRE, Marian: Why looking isn't always seeing: Readership Skills and Graphical Programming. 36 (1995), Nr. 6, S. 33–44. – ISSN 0001–0782
- [Pfa85] PFAFF, Guenther E. (Hrsg.): *User Interface Management Systems*. Springer, Berlin, 1985. – ISBN 0–387–13803–X
- [PK03] PAECH, B. ; KOHLER, K.: Usability Engineering integrated with Requirements Engineering. In: *Proceedings of the ICSE 2003, Bridging the GAPS between Software Engineering and Human-Computer Interaction*, 2003, S. 36–40
- [Ras00] RASKIN, Jef: *The Humane Interface*. Addison Wesley, 2000. – ISBN 0–201–37937–6
- [RNT95] RICHARD N. TAYLOR, Joëlle C. (Hrsg.): *Software Engineering and Human-Computer Interaction, ICSE '94 Workshop on SE-HCI: Joint Research Issues, Sorrento, Italy, May 16-17, 1994, Proceedings*. Springer, Berlin, 1995 (Lecture Notes in Computer Science). – ISBN 3–540–59008–0
- [Sal95] DANIEL SALBER, Laurence N. (Hrsg.). *The System Modelling Glossary*. 1995
- [Shn82] SHNEIDERMAN, Ben: Multiparty grammars and related features for defining interactive systems. In: *IEEE Transactions on systems, man and cybernetics* 12 (1982), March/April, Nr. 2, S. 148–154
- [Shn98] SHNEIDERMAN, Ben: *Designing the user interface*. Addison-Wesley, 1998. – ISBN 0–201–69497–0
- [SM91] SUTCLIFFE, A. G. ; MCDERMOTT, M.: Integrating methods of human-computer interface design with structured systems development. In: *International Journal of Man-Machine Studies* 34 (1991), Nr. 5, S. 631–655. – ISSN 0020–7373
- [Sta96] STARY, Christian: *Interaktive Systeme - Software-Entwicklung und Software-Ergonomie*. 2. Vieweg Verlag, 1996. – ISBN 3–528–15384–9
- [Sta00] STARY, Christian: TADEUS, Seamless Development of Task-Based and User-Oriented Interfaces. In: *IEEE Transactions on Systems, Man and Cybernetics* 30 (2000), September, Nr. 5, S. 509–525
- [Str85] STRUBBE, Hugo J.: Components of Interactive Applications. In: PFAFF, Guenther E. (Hrsg.): *User Interface Management Systems*, Springer, Berlin, 1985. – ISBN 0–387–13803–X, S. 3–8
- [Tab00] TABELING, Peter: *Der Modellhierarchieansatz zur Beschreibung nebenläufiger, verteilter und transaktionsverarbeitender Systeme*. Shaker Verlag, 2000. – ISBN 3–8265–7989–5

- [Thi00] THISSEN, Frank: *Screen Design Handbuch*. Springer-Verlag, 2000. – ISBN 3-540-64804-6
- [Tid99] TIDWELL, Jennifer: *COMMON GROUND: A Pattern Language for Human-Computer Interface Design*. 1999. – http://www.mit.edu/~jtidwell/common_ground.html
- [Was85] WASSERMAN, Anthony I.: Extending State Transition Diagrams for the Specification of Human-Computer Interaction. In: *IEEE Transactions on Software Engineering* 11 (1985), August, Nr. 8, S. 699–713
- [Wen70] WENDT, Siegfried: Eine Methode zum Entwurf komplexer Schaltwerke unter Verwendung spezieller Ablaufdiagramme. In: *Digital Processes* 5 (1970), Nr. 3-4, S. 213–222
- [Wen80] WENDT, Siegfried: On the Partitioning of Computing Systems into Communicating Agencies. In: *Informatik-Fachberichte GI-NTG Fachtagung Struktur und Betrieb von Rechensystemen*, Springer-Verlag, Maerz 1980, S. 194–204
- [Wen91] WENDT, Siegfried: *Nichtphysikalische Grundlagen der Informationstechnik*. Springer-Verlag, Berlin, 1991. – ISBN 3-540-54452-6
- [Wen01] WENDT, Siegfried: Ein grundlegender Begriffsrahmen für das Wissensmanagement im Software-Engineering. In: *Proceedings of the KnowTech (Dresden, 2001)*, 2001. – <http://www.community-of-knowledge.de>
- [Whi87] WHITEFIELD, A.: Models in Human Computer Interaction: a classification with special reference to their uses in design. In: H.-J. BULLINGER, B. S. (Hrsg.): *Human-Computer Interaction: INTERACT'87*. Amsterdam : North-Holland, 1987, S. 57–63
- [Wor92] WORKSHOP, CHI'91 UIMS Tool D.: A Metamodel for the Runtime Architecture of an Interactive System. In: *ACM SIGCHI Bulletin* 24 (1992), January, Nr. 1, S. 32–37
- [You83] YOUNG, Richard M.: Surrogates and Mapping: To kinds of conceptual models for interactive devices. In: DEDRE GENTNER, Albert L. S. (Hrsg.): *Mental Models*. Hillsdale, NJ : Lawrence Erlbaum Associates, 1983, S. 35–52

Index

Übergangintervall, 21

Abbildung, 13

Abstraktion, 11

adapterinhärente Form, 56

Akteur, 26

Anschauungsmodell, 17

Anwender, 52

Arbeitsteilung, 70

Arch-Slinky-Modell, 106

Architektur, 69

Aufbaudiagramm, 25

Aufgabenmodell, 86, 89

Auflösung, 10

Aussage, 14

Ausschnitt, 10

Bedienwissen, 50

Beispiel

Autorensystem, 43, 44, 47, 68

Bibliothekssystem, 83

eMail-System, 93

Flugreservierungssystem, 90

Rubik-Würfel, 87

Warmwasserversorgung, 19

Wegeplanungssystem, 36

Benennen, 13

Benutzer, 30

Benutzerfähigkeiten, 50

Benutzungsschnittstelle, 31

Benutzungsschnittstellendesigner, 37

Beobachtungsort, 10

Beschreibung, 14

Betrachtungsebenenwechsel, 20

Computer Graphics Reference Model,
CGRM, 72

Data Flow Diagram, DFD, 83

Didaktik, 16

Domänenmodell, 32

Eigenschaften, 8

Empfänger, 12

Entity-Relationship-Diagramm, 25

Entwickler, 37

Entwicklungsumgebung, 75

Entwurfsentscheidung, 20

Ereignis, 9

Ereignisfolgengeflecht, 21

Ereigniskommunikation, 14

ergebnisorientiertes System, 40

Ersatzmodell, 20

externe Kontrolle, 73

Fachwissen, 50

Flow of events storyboard, 92

Formadapter, 55

Fundamental Modeling Concepts, FMC,
25

Gestaltungsmuster, 60

Grammatik, 81

IFIP-Modell, 60

Information, 21

Informationeller Kern, 70

Interaktionsadapter, 52

Interaktionsassistent, 63

interaktionsbezogenes Modell, 33, 48

Interaktionselement, 60

Interaktionsprozess, 30

Interaktionssystem, 30

Interaktionswerk, 71

Interaktives System, 30

interne Kontrolle, 73

Interpretation, 14

Interpretationskette, 15

Interpretationsvereinbarung, 14

Kanal, 26

- Kanalmultiplex, 66
- Klasse, 11
- Kognitive Fähigkeiten, 50
- Kommunikation, 12
- konzeptionelles Modell, 32

- Mapping, 32
- materiell-energetischer Zweck, 44
- Model-View-Controller, 94
- Modellsynthese, 40
- Modellsystem, 13
- Modellverfeinerung, 45
- Multiagent Model, 72
- Multiparty-Grammatik, 81
- MVC-Triade, 95

- Objekt-orientierte Analyse, 89
- Observer-Pattern, 100
- operationeller Zustand, 26
- Ortsmultiplex, 66

- PAC-AMODEUS, 106
- Petri-Netz, 25
- Physische Fähigkeiten, 50
- Piktogramm, 15
- Präsentieren, 13
- Presentation-Application-Control, PAC, 100
- Problembereichsmodell, 89
- Programmierung, 22
- Prozesse, 9
- prozessorientiertes System, 40

- Rational Unified Process, RUP, 92
- realisierungsbezogenes Modell, 33, 68
- Realisierungsmodell, 20
- Rollenmultiplex, 22
- Rollensystem, 22

- SA/SD, 83
- Schnittstelle, 9
- Schnittstellengerät, 70
- Schnittstellengestaltung, 65
- Schnittstellenvariable, 9
- Schrittfolgenadapter, 55
- Seeheim-Modell, 103
- sekundärer Kanal, 63
- Sender, 12
- Smalltalk-80-System, 95
- Speicher, 26
- Sprache, 14
- Steuerkreis, 96
- Steuerzustand, 26
- Structured Design, 84
- Structured English, 83
- Surrogate, 32
- Symbol, 13
- System, 6
 - dynamisches System, 6
 - gerichtetes System, 22
 - informationelles System, 21
 - Interaktionssystem, 30
 - interaktives System, 30
 - programmierbares System, 22
- Systemarchitekt, 37
- Systemkomponente, 10
- Systemmodell, 10

- Task analysis, 86
- Terminal-System, 79
- Trägersystemmodell, 22
- Transitionsdiagramm, 79
- Treiber, 71
- Typ, 11

- Umschreiben, 14
- Use Case Storyboard, 92
- User Interface Management System, UIMS, 103

- virtuelle Objekte, 62

- Wechselwirkung, 9
- Wert, 9
- wertdiskret, 21
- Wertidentifikation, 52

- Zeigen, 12
- zeitdiskret, 21
- Zeitmultiplex, 66
- zweckbezogener Kanal, Realisierung, 56
- zweckbezogener Kern, Realisierung, 71
- zweckbezogenes Modell, 19, 33, 39