

Detlef Zühlke

Nutzergerechte Entwicklung von Mensch-Maschine- Systemen

Useware-Engineering für technische Systeme

2., neu bearbeitete Auflage



Springer

VDI

Nutzergerechte Entwicklung von Mensch-Maschine-Systemen

Detlef Zühlke

Nutzergerechte Entwicklung von Mensch-Maschine-Systemen

Useware-Engineering für technische Systeme

2., neu bearb. Auflage



Springer

Prof. Dr.-Ing. Detlef Zühlke
Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI)
Trippstadter Straße 122
67663 Kaiserslautern
Deutschland
Detlef.Zuehlke@dfki.de

ISBN 978-3-642-22073-9 e-ISBN 978-3-642-22074-6
DOI 10.1007/978-3-642-22074-6
Springer Heidelberg Dordrecht London New York

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

© Springer-Verlag Berlin Heidelberg 2004, 2012

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funksendung, der Mikroverfilmung oder der Vervielfältigung auf anderen Wegen und der Speicherung in Datenverarbeitungsanlagen, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Eine Vervielfältigung dieses Werkes oder von Teilen dieses Werkes ist auch im Einzelfall nur in den Grenzen der gesetzlichen Bestimmungen des Urheberrechtsgesetzes der Bundesrepublik Deutschland vom 9. September 1965 in der jeweils geltenden Fassung zulässig. Sie ist grundsätzlich vergütungspflichtig. Zu widerhandlungen unterliegen den Strafbestimmungen des Urheberrechtsgesetzes.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Einbandentwurf: WMXDesign GmbH, Heidelberg

Gedruckt auf säurefreiem Papier

Springer ist Teil der Fachverlagsgruppe Springer Science+Business Media (www.springer.com)

Vorwort

Wir brauchen einfache und bequeme Sachen, die dem gesunden Menschenverstand einleuchten. Denken Sie (Ingenieure) es nicht so sehr von der Technik, sondern denken Sie es immer von dem, der dann voller Hoffnung in den Laden geht und zu Hause plötzlich mutterseelenallein mit der Gebrauchsanweisung steht.

Bundeskanzlerin Angela Merkel auf der CeBIT 2006

Bundeskanzlerin Merkel hat mit dieser Mahnung an die Adresse der Ingenieure sicher vielen Menschen aus der Seele gesprochen. Die technischen Neuerungen nehmen mit immer kürzer werdenden Innovationszyklen zu, aber man kann sich des Eindrückes nicht erwehren, dass der Mensch in dieser Entwicklung schlichtweg vergessen wird. Wer kann heute noch seinen DVD-Recorder in allen seinen Funktionen bedienen oder wer kann alle Funktionen seines Handys nutzen, ohne die buchdicken Bedienungsanleitungen zu konsultieren? Und hiermit stehen wir sicher erst am Anfang einer schnell fortschreitenden „Funktionalitis“. Intelligente Kühl-schränke, Smart Homes, Internet im Auto, Avatare und andere sog. Errungenschaften des Fortschritts stehen uns noch bevor und werden sicher das Problem weiter verschärfen. Dieser Trend ist in allen Lebensbereichen – von alltäglichen Geräten im Haushalt bis hin zu komplexen industriellen Systemen – zu verzeichnen.

Die Basis hierfür bereitet die enorme Leistungsfähigkeit moderner Mikroelektronik, verbunden mit den Fortschritten in der Kommunikationstechnik. Damit werden neue Geräteneigenschaften bei vertretbarem Entwicklungsaufwand machbar, die zu einer steigenden Funktionalität und zunehmenden Komplexität dieser Geräte führen. Sowohl die privaten Nutzer als auch die Facharbeiter im industriellen Bereich fühlen sich von der zunehmenden Informationsflut mehr und mehr überfordert. Der Gestaltung menschengerechter Systeme und Produkte wird daher in der zukünftigen Technikgestaltung eine Schlüsselrolle zukommen und auch zunehmend zum Marketingaspekt werden.

In den vergangenen 20 Jahren hat das Zentrum für Mensch-Maschine-Interaktion an der Technischen Universität Kaiserslautern eine Vielzahl von Industrie-Projekten zur nutzergerechten Gestaltung technischer Systemen durchgeführt. Dabei konnten wir nicht nur sehr viele Erfahrungen bei der Analyse, Gestaltung und Evaluation von Mensch-Maschine-Systemen sammeln, sondern mussten auch feststel-

len, dass den meisten Entwicklern das Wissen fehlt, wie man MMS entwickelt. Deswegen veranstalten wir bereits seit vielen Jahren Schulungsseminare, in denen wir Entwicklern das nötige Basiswissen zur Gestaltung nutzergerechter Bediensysteme vermitteln.

Das vorliegende Buch ist das Ergebnis dieser umfangreichen industriellen Erfahrungen. Es soll vor allem Entwicklern das notwendige Basiswissen praxisgerecht vermitteln und ein Leitfaden bei der Entwicklung nutzergerechter Systeme sein. Es ist also aus der Praxis für die Praxis entstanden. Dabei ist mir bewusst, dass die anfangs bereits beschriebene Geschwindigkeit des technischen Fortschritts zu immer neuen Interaktionstechniken führen wird. Das iPhone und das iPad machen dies sehr deutlich, zeigen aber auch, dass die Einfachheit der Bedienung ein wichtiges Element für den Verkaufserfolg ist.

Bei der Erarbeitung dieses Buches haben mich viele meiner ehemaligen und aktuellen Mitarbeiterinnen und Mitarbeiter unterstützt. Hierzu zählen Dr.-Ing. Geritt Meixner, Jun.-Prof. Dr.-Ing. Kerstin Röse, Dr.-Ing. Alexander Bödcher, Dr.-Ing. Ssamuela Mukasa, Dr.-Ing. Holger Oortmann, Dr.-Ing. Achim Reuther, Dr.-Ing. Gerd Schneider, Dr.-Ing. Lutz Krauss, Dipl.-Ing. Ines Heck, denen ich an dieser Stelle meinen Dank sagen möchte.

Kaiserslautern
im September 2011

Detlef Zühlke

Inhalt

1 Einführung.....	1
2 Der Mensch und seine Fähigkeiten.....	5
2.1 Informationsaufnahme	6
2.1.1 Das visuelle System	8
2.1.2 Das auditive System.....	11
2.1.3 Das haptische System.....	14
2.2 Wahrnehmung und Kognition	15
2.2.1 Wahrnehmung	18
2.2.2 Kognition und Handeln.....	22
2.3 Menschliches Versagen	28
2.3.1 Fehlertaxonomie.....	29
2.3.2 Fehlerursachen in komplexen technischen Systemen.....	30
2.3.3 Psychosoziale und emotionale Fehlerursachen.....	33
3 Vorgehensweise bei der Useware-Entwicklung	35
3.1 Vorgehensweise	35
3.1.1 Projektteam	35
3.1.2 Der Entwicklungsprozess.....	36
3.1.3 Die Bedeutung eines Moderators.....	38
3.2 Analyse.....	39
3.2.1 Der Nutzer, ein unbekanntes Wesen?.....	40
3.2.2 Wie gehe ich konkret vor?.....	41
3.2.3 Wen befrage ich?	43
3.2.4 Welche Methoden sind geeignet?.....	46
3.2.5 Wie sehen die Ergebnisse aus?.....	48
3.3 Strukturgestaltung	50
3.3.1 Beschreibung einer grundlegenden Benutzungsstruktur.....	51
3.3.2 Erweiterung des Benutzungsmodells	56
3.3.3 Verknüpfung von Benutzungs- und Funktionsmodell.....	59
3.3.4 Berücksichtigung weiterer Attribute	60
3.3.5 Ergebnis der Strukturgestaltung	62
3.3.6 Hilfsmittel für die Strukturgestaltung	64

3.4	Gestaltung	65
3.4.1	Vorgehensweise	66
3.4.2	Plattformspezifikation	67
3.4.3	Plattformspezifisches Benutzungsmodell	75
3.4.4	Navigationskonzept	76
3.4.5	Strukturierung von Informationen	79
3.4.6	Klassifizierung von Informationen	80
3.4.7	Dialoge	83
3.4.8	Statisches Bildschirmlayout	86
3.4.9	Dialogelemente	89
3.4.10	Feingestaltung	90
3.4.11	Erstellung eines Style-Guides	103
3.5	Realisierung	105
3.6	Evaluation	106
3.6.1	Evaluationsverfahren	107
3.6.2	Evaluationsmethoden	109
4	Useware-Systeme für internationale Märkte	123
4.1	Interkulturelles Design	124
4.2	Interkulturelles Useware-Engineering	127
4.2.1	Interkultureller Entwicklungsprozess	127
4.2.2	Useware-Design für den chinesischen Markt	131
4.2.3	Das Projekt INTOPS 2	132
4.3	Kulturorientierte Gestaltung für den chinesischen Markt	134
4.3.1	Designaspekte	134
4.3.2	Hinweise für die Funktions-Ebene	135
4.3.3	Hinweise für die Oberflächen-Ebene	136
4.3.4	Hinweise für die Interaktions-Ebene	137
4.4	Probleme kulturspezifischen Designs	139
5	Hilfsmittel bei der Useware-Entwicklung	143
5.1	Entwicklungswerkzeuge	144
5.1.1	Universelle Entwicklungswerkzeuge	144
5.1.2	Prozessvisualisierungswerkzeuge	147
5.1.3	Domänenspezifische Entwicklungswerkzeuge am Beispiel Automotive	147
5.1.4	Rapid-Prototyping-Werkzeuge	149
5.2	UML – Unified Modeling Language	151
5.2.1	UML-Anwendungsbeispiel	152
5.2.2	Anwendungsfalldiagramm	153
5.2.3	Klassendiagramm	153
5.2.4	Aktivitätsdiagramm	155
5.2.5	Zustandsdiagramme	157
5.3	Model-Driven Architecture (MDA)	157
5.4	Modellbasierte Useware-Entwicklung	159

5.4.1	Kernmodelle	161
5.4.2	Modellbasierte Architektur	163
5.4.3	Notationen und Tools für die Analyse	163
5.4.4	Notationen und Tools für die Strukturgestaltung	167
5.4.5	Notationen und Tools für die Gestaltung	169
5.4.6	Zusammenfassung und Ausblick	174
6	Grundlegende Prinzipien und Interaktionstechniken	177
6.1	Normen und Richtlinien	177
6.1.1	DIN EN ISO 9241	178
6.1.2	VDI 3850	181
6.1.3	VDI 3699	183
6.2	Gestaltungsprinzipien für Bediensysteme	184
6.2.1	Grundsätze der Dialoggestaltung	184
6.2.2	Weitere Prinzipien	187
6.2.3	Gestaltgesetze	190
6.3	Interaktionsgestaltung	195
6.3.1	Interaktionsformen	195
6.3.2	Interaktionselemente	201
6.3.3	Interaktionstechnologien	209
7	Zukünftige Interaktionstechniken	227
7.1	Multimodale Interaktion	227
7.2	Mobilität durch Miniaturisierung	229
7.3	Neue Interaktionstechniken	230
7.4	Displaytechnologien	231
7.4.1	Flexible Displays	231
7.4.2	3D Monitore (Autostereoskopische Displays)	231
7.4.3	Virtuelle Realität (VR)	234
7.4.4	Augmented Reality (AR)	235
7.5	Netzwerktechnologien	236
7.6	Agentensysteme	236
7.7	Emotionale Systemgestaltung	238
Literatur	241	
Sachverzeichnis	247	

Kapitel 1

Einführung

Bis in die 70er Jahre bestanden Gerätesteuerungen vorwiegend aus Hardware; neue Funktionen konnten nur durch immer komplexere Hardwareentwicklungen realisiert werden. Mit der Erfindung des Mikroprozessors Mitte der 70er-Jahre war es dann erstmals möglich, teure Hardwareentwicklungen durch standardisierte Mikroprozessor-Hardware plus „billiger“ – da reproduzierbarer – Software zu ersetzen. Doch die Euphorie hielt nicht lange an. Schon Mitte der 80er-Jahre sprach man von der Softwarekrise, da die Softwareentwicklung einen Umfang annahm, der ebenso zu enormen Entwicklungs- und vor allem Wartungskosten führte. Auch hier half die Standardisierung – sei es durch die offiziellen Standards wie ISO, IEEE usw. oder durch die immer stärker aufkommenden Firmen-Standards wie die PC-Technik oder Microsoft-WINDOWS – die Probleme in den Griff zu bekommen. So werden heute viele technische Systeme im Konsum- wie auch im industriellen Bereich weitest möglich auf einer PC-Plattform aufgebaut und mit Standardsoftware bestückt, die man preiswert und erprobt auf dem Markt kaufen kann. Damit werden die Systeme jedoch immer ähnlicher, so dass die notwendige Produktdifferenzierung mehr und mehr im Bereich der Bedienoberflächengestaltung stattfindet.

Man könnte die heutige Situation vereinfachend wie folgt umschreiben: man kaufe eine standardisierte Hardwareplattform wie etwa die PC-Technik, ergänze sie durch die absolut notwendige gerätespezifische Hardware, man kaufe dazu die ebenso standardisierte Software (WINDOWS, Gerätetreiber, Netzwerksoftware) und entwickle nur noch die Bedienoberfläche selbst. Diese sicher etwas übertrieben gezeichnete Situation ist aber durchaus realitätsnah. So erklärte ein großer deutscher Elektrokonzern vor kurzem, dass in industriellen Entwicklungsvorhaben bis zu 70 % der Softwarekosten in die Bediensystementwicklung fließen.

In Analogie zur Softwarekrise wäre es heute angebracht, von einer Bedienkrise zu sprechen. Untersuchungen zeigen, dass viele Menschen nicht mehr in der Lage sind, komplexe Technik in allen ihren Funktionen zu bedienen (Zühlke 2005). Mag dies im Bereich der Konsumgüter wie Videorecorder oder Handys noch hingenommen werden (...dafür bin ich halt zu dumm...), so ist es bei Investitionsgütern nicht tolerierbar. Eine Untersuchung im Rahmen eines großen BMBF-Projektes (HÜM-NOS) in den 90er-Jahren machte deutlich, dass den Maschinenbedienern meist nur 50 % des Funktionsumfangs ihrer Maschine bekannt war und sie darüber hinaus

im Schnitt 2–3 Wochen Schulung benötigten. Die Bediensysteme waren von ihren Entwicklern i. w. funktionsorientiert aufgebaut worden, während der Nutzer aufgabenorientiert denkt und handelt.

Die besagte Softwarekrise wurde aber vor allem dadurch entschärft, dass sich eine neue Disziplin etablierte: das Software-Engineering, das in den folgenden Jahren die anfangs sehr spontane und unsystematische Softwareentwicklung durch neue Entwicklungsmethoden in einen ingenieurmäßigen Entwicklungsprozess überführt hat. Mittlerweile ist die Zeit reif, auch im Bereich der Bediensystemgestaltung diesen Weg zu gehen. So wurde schon 1998 der Begriff **USEWARE** als Sammelbegriff eingeführt für alle Hard- und Softwarekomponenten, die der Benutzung dienen. Mit dem Begriff *Useware* verbindet sich eine Fokussierung der Technikgestaltung auf menschliche Fähigkeiten und Bedürfnisse. Den Nutzer mit seinen Fähigkeiten und Grenzen zu begreifen und die Technik daran auszurichten, ist die einzige Erfolg versprechende Methode bei der Gestaltung zukünftiger technischer Produkte und Systeme. Mit dem eigenständigen Begriff *Useware* soll darüber hinaus deutlich gemacht werden, dass neben den traditionellen Entwicklungsfeldern Hard- und Software die *Useware* einen mindestens ebenso bedeutenden Platz einnimmt. Und so wie wir selbstverständlich von Hardware- oder Software-Engineering sprechen, so sollten wir das Useware-Engineering als die Lehre von der ingenieurmäßigen Gestaltung der Useware etablieren.

Die Spannweite des Useware-Engineering ist dabei bei weitem größer als bei den beiden anderen Disziplinen. Braucht man bei der Hardware vor allem Elektrotechniker und bei der Software Informatiker, so ist das Kompetenzfeld beim Useware-Engineering sehr viel breiter angelegt. Da die Useware die Hard- und Software der Bediensysteme umfasst, benötigt man sicher Kompetenzen im Electrical- und Software-Engineering. Darüber hinaus zeigt sich in den Projekten immer wieder, dass die Arbeits- und Kognitionspsychologie eine sehr wichtige Kompetenz darstellt, die vor allem in den frühen Projektphasen der Nutzungsanalyse sehr wichtig ist. Weitere Kompetenzen können aus dem Bereich der Arbeitsmedizin sowie der Berufspädagogik erforderlich werden. Und selbstverständlich braucht man die Kompetenz im jeweiligen Anwendungsfeld, z. B. im Maschinen- und Anlagenbau. Daraus lässt sich schließen, dass das Useware-Engineering ein sehr interdisziplinäres Tätigkeitsfeld ist, welches nur im Team unterschiedlicher Kompetenzen erfolgreich bearbeitet werden kann.

Das vorliegende Buch soll Entwicklern von Useware-Systemen – und dies sind vor allem Ingenieure – das notwendige Basiswissen zur Verfügung stellen, mit dem sie komplexe technische Systeme aufgabenorientiert und nutzergerecht entwickeln können. Das Buch ist wie folgt aufgebaut.

Während Ingenieure in ihrer Ausbildung sehr viel über die Eigenschaften technischer Systeme lernen, ist das Wissen über die Fähigkeiten und vor allem die Grenzen des Menschen als dem zweiten Kommunikationspartner in einem Mensch-Maschine-System weitgehend nicht vorhanden. Dementsprechend wird in Kap. 2 eine kurze und zielgerichtete Einführung in die Informationsaufnahme und -verarbeitung des Menschen gegeben. Es ist unabdingbar, die wichtigsten Fähigkeiten und

auch Grenzen des kognitiven Systems Mensch zu kennen. An dieser Stelle soll auch das Thema menschliches Versagen angesprochen und versachlicht werden.

In Kap. 3 wird den Lesern die Vorgehensweise bei der Entwicklung von Useware-Systemen detailliert anhand vieler praxisnaher Beispiele vorgestellt. Beginnend mit der Projektorganisation, über die sehr wichtige Analysephase sowie die Strukturgestaltung und Gestaltung bis hin zur Realisierung wird den Entwicklern das nötige praktische Wissen vermittelt.

In vielen industriellen Entwicklungsprojekten wird den Unternehmen spätestens in der Analysephase klar, dass ihre Produkte für den Weltmarkt bestimmt sind. So mit müssen für eine nutzergerechte Entwicklung natürlich repräsentative Nutzer aus allen Märkten befragt werden. In vielen Projekten wurden mittlerweile solche internationalen Nutzerbefragungen durchgeführt. Dabei zeigen sich vor allem in den bedeutenden außereuropäischen Märkten wie den USA oder Fernost starke kulturbedingte Einflüsse auf die Maschinen- und Anlagenbedienung. In Kap. 4 sollen solche typischen Besonderheiten angesprochen und eine Vorgehensweise für eine kulturspezifische Useware-Gestaltung präsentiert werden.

Das Kap. 5 widmet sich den Hilfsmitteln der Useware-Entwicklung. Beginnend mit der Einführung zu verschiedenartigen Entwicklungswerkzeugen (universelle und prozessvisualisierende Entwicklungswerkzeuge) werden domänenspezifische Entwicklungswerkzeuge in der Automotive-Domäne sowie Rapid-Prototyping-Werkzeuge vorgestellt. In weiteren Unterkapiteln wird neben einer kurzen Einführung der Unified Modeling Language (UML) insbesondere tiefergehend auf das aktuell wichtige Thema der modellbasierten Useware-Entwicklung eingegangen.

Das Kap. 6 stellt eine Art Referenz für Entwickler von Maschinenbediensystemen dar. Hier werden grundlegende Gestaltungsmerkmale, wichtige Normen, Standards und Richtlinien, sowie Interaktionsformen, -elemente und -technologien vorgestellt und ihre Anwendungsfelder beschrieben.

Die schnellen Innovationszyklen in der Mikroelektronik und den Softwaretechnologien werden uns in den nächsten Jahren immer neue Techniken bieten, die starke Auswirkungen auf die Useware-Entwicklung haben werden. Seien dies die autostereoskopischen Bildschirme, die uns realitätsnahe 3D-Systeme ohne Hilfsmittel wie Shutterbrillen o. ä. darstellen lassen oder die große Palette der multimodalen Interaktionstechniken, die Sprach-, Gesten- und Mimikerkennung beinhalten. In Kap. 7 soll ein Ausblick auf diese neuen Technologien und ihren Einfluss auf die Useware-Gestaltung gegeben werden.

Kapitel 2

Der Mensch und seine Fähigkeiten

Mensch und Maschine sind Kommunikationspartner, die sich in ihren Eigenschaften und Fähigkeiten wesentlich unterscheiden. Während die technische Ausführung von Maschinen hinsichtlich Funktionen, Bedienkonzepten, Automatisierungsgrad etc. vom Konstrukteur in weiten Grenzen variabel gestaltet werden kann, ist der Mensch als Anwender und somit Kommunikationspartner der Maschine durch weitgehend festliegende Fähigkeiten und somit auch Grenzen gekennzeichnet. Andererseits müssen wir die Eigenschaften und das Verhalten der Maschine über ihre Lebensdauer als konstant ansehen, während der Mensch in der Lage ist, sich permanent auf neue Anforderungen einzustellen. Der Mensch ist also beschränkt in seiner Wahrnehmung und seiner Informationsverarbeitung, dem hingegen wiederum sehr flexibel in der Lage, neue unbekannte Problemstellungen zu lösen.

In den letzten 20 Jahren wurden immer wieder Versuche unternommen, den Menschen nicht nur in seinen physischen Eigenschaften sondern auch in seinen kognitiven Eigenschaften zu modellieren. Zwar lassen sich für viele Fähigkeiten quantitative Grenzwerte ermitteln, doch sind sie nicht als Fixum anzusehen, sondern teilweise in weiten Bereichen variabel. Das menschliche Gehirn ist bestens darauf ausgelegt, neue Vorgehensweisen zu erlernen und die Informationsmengen auf eine der folgenden Verarbeitung angepasstes Maß zu reduzieren. Darüber hinaus wird die kognitive Leistung des Menschen stark von physischen und physiologischen Randbedingungen beeinflusst. Stress, Unwohlsein, persönliche Probleme oder Alkohol beeinflussen diese Grenzwerte meist negativ. Hingegen wirken Erfolgserlebnisse, Glücksgefühl oder etwa Ausgeruhtheit meist positiv. Diese Einflüsse lassen sich als „Black Box“ des menschlichen Verhaltens beschreiben, da es für den Außenstehenden schwer zu erkennen ist, welchen Einflüssen der jeweilig betrachtete Mensch unterliegt.

Maschinenentwickler neigen nicht selten dazu, die Fähigkeiten des Menschen zu überschätzen und dabei stillschweigend davon auszugehen, dass sich der flexible Mensch an die unflexible Maschine anzupassen hat. Die Folge davon sind lange Schulungszeiten, Stress am Arbeitsplatz oder Angst vor Fehlern. Hier gilt es, ein neues Verständnis bei Entwicklern zu schaffen, in dem der Mensch mit seinen Fähigkeiten im Mittelpunkt steht und die Maschine konsequent auf seine Fähigkeiten ausgerichtet wird. Diese Vorgehensweise wird auch als *Human-Centered-Design* bezeichnet.

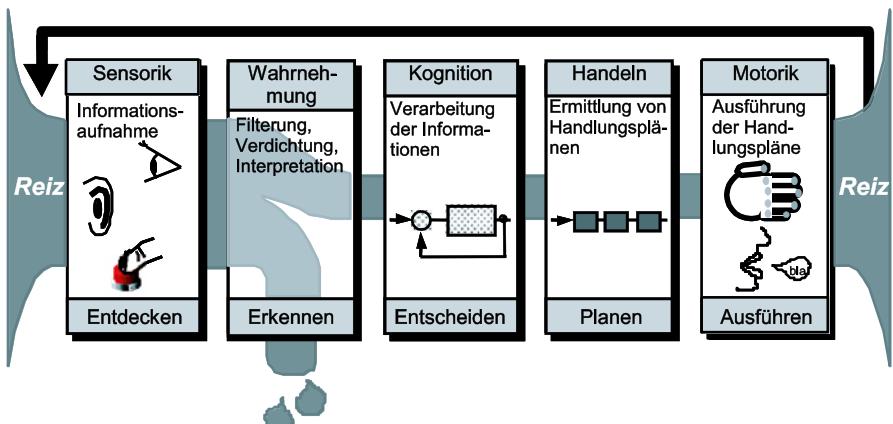


Abb. 2.1 Informationsaufnahme und -verarbeitung

Zur Anpassung von Bediensystemen an den Menschen als Bediener sind zunächst einmal grundlegende Kenntnisse über dessen Informationsaufnahme und -verarbeitung notwendig. Dazu gehören (Herczeg 1994)

- die **Sensorik**, d. h. beim Menschen die Eigenschaften der Sinne Licht, Schall, Druck, Temperatur, Geruch und Geschmack zur Informationsaufnahme,
- die **Wahrnehmung**, d. h. die Verdichtung und Interpretation der über die Sensorik aufgenommenen Informationen (Erkennung symbolischer Elemente und struktureller Muster),
- die **Kognition**, d. h. die zielgerichtete Verarbeitung der verdichteten und interpretierten Informationen (eigentliche Denkleistung mit Problemlösungsstrategien),
- das **Handeln**, d. h. die Umsetzung der während der Kognition ermittelten Ergebnisse in Handlungspläne und
- die **Motorik**, d. h. die Ausführung der Handlungspläne durch die Akteure wie Hand, Fuß, Auge oder auch die Stimme.

Die einzelnen Phasen bei der menschlichen Informationsaufnahme und -verarbeitung zeigt Abb. 2.1 vereinfacht im Überblick. Im Folgenden werden diese detaillierter beschrieben. Dabei findet eine Beschränkung auf die Vorgänge und Eigenschaften statt, die bei der Gestaltung der Schnittstelle zwischen Mensch und Maschine zu beachten sind.

2.1 Informationsaufnahme

Die Informationsaufnahme und -verarbeitung des Menschen ist ein faszinierendes Thema. Allein die Art und Weise der Datenaufnahme und die Datenmengen, die der Mensch durch seine sensomotorischen Fähigkeiten aufnehmen kann, sind ein

Tab. 2.1 Menschliche Sinne und zugeordnete Informationsflüsse

Sinn	Organ	Sinnesfläche cm ²	Zahl der Rezeptoren	Zahl der Nervenfasern	Informationsfluss in bit/s
Licht	Auge	23,5	2.130.106	2.106	10.000.000
Schall	Ohr	20,08	215.000	2.000	50.000
Druck	Haut	20.000	500.000	10.000	200.000
Temperatur	Haut	20.000	200.000	10.000	2.000
Geruch	Nase	22,5	20.106	2.000	100
Geschmack	Mund	50	2.000	2.000	10

Phänomen für sich. Es lässt sich errechnen, dass der Mensch ca. 10 Mbit/s, das sind 80 % seiner insgesamt aufgenommenen Informationen, über den visuellen Kanal d. h. seine Augen aufnimmt. Dieser immense Informationsstrom wird jedoch in einer ersten Phase der Informationsverarbeitung im Gehirn bereits sehr stark reduziert. Die externen Reize des sensorischen Systems treffen dabei auf abgegrenzte Bereiche des Gehirns, in denen neuronale Netze eine Mustererkennung durchführen. Die eingehenden Informationen werden einer Datenanalyse unterzogen und auf ihre Relevanz hinsichtlich des aktuellen Handlungsziels des Nutzers geprüft. Information, die relevant sein könnte, wird zur weiteren Verarbeitung an das Kurzzeitgedächtnis weiter geleitet, unwichtige Information wird ausgeblendet. Dieser Prozess wird unterbewusst durch das Gehirn gesteuert. Durch Versuche lässt sich herausfinden, dass nach der Datenvorverarbeitung noch ca. 10 bit/s im Mittel, 16 bit/s im Spitzenwert vom Menschen verarbeitet werden können. Diese Werte variieren abhängig vom Alter und auch vom Gemüts- und Bewusstseinszustand. Der Maximalwert wird ungefähr zwischen dem 20. und 35. Lebensjahr bei ausgelenktem Gemütszustand erreicht (Dinges 1982).

Diese recht scharfen Messwerte könnten einen dazu verleiten, die Güte eines Mensch-Maschine-Systems informationstheoretisch rechnerisch zu ermitteln, indem man die Informationsmengen bei einer bestimmten Bedienhandlung ermittelt und sie den oben beschriebenen Grenzwerten unter Berücksichtigung des Bedieneralters gegenüberstellt. Leider funktioniert das nicht. Wie im Kap. 2.2.1 noch deutlich werden wird, lässt sich die Wahrnehmung des Menschen nicht mit informationstheoretischen Methoden und Messgrößen wie bit/s ausreichend beschreiben. Stattdessen werden Informationen situationsabhängig zu Bedeutungen verdichtet.

Hieraus wird deutlich, dass es einerseits keinen Sinn macht, die Qualität einer Mensch-Maschine-Schnittstelle also z. B. die Gestaltung einer Anzeige rein mathematisch zu beschreiben, andererseits aber das Wissen um die vom Menschen aufnehmbaren Informationsmengen und die Prinzipien der Informationsverarbeitung notwendiges Basiswissen für jeden Entwickler sein sollte.

Zur Informationsaufnahme nutzt der Mensch seine Sinnesorgane, um die physikalischen Größen Licht, Schall, Druck, Temperatur, Geruch und Geschmack zu erfassen. Tabelle 2.1 zeigt die verschiedenen Sinne und die maximalen durch sie aufnehmbaren Informationsmenge (Bauer 1981).

Tab. 2.2 Physikalische und entsprechende physiologische Größen

Physikalische Größe	Einheit	Physiologische Größe
Lichtstärke	cd	Helligkeit
Wellenlänge	nm	Farbe (Farbton)
Spektrale Zusammensetzung	–	Sättigung
Intensität und Wellenlänge	–	Kontrast

Für die weitere Betrachtung sind allerdings weniger die reinen Sensordaten von Interesse sondern eher die Wahrnehmung der darin enthaltenen Informationen. Man spricht dann von den Sinnessystemen. Für den Bereich der Mensch-Maschine-Systeme sind folgende Sinne die wichtigsten:

- Das visuelle System (Gesichtssinn)
- Das auditive System (Gehörsinn)
- Das haptische System (Tastsinn)
- Das Vestibularsystem (Gleichgewichtssinn)
- Das olfaktorische System (Geruchssinn)

Davon sollen die drei erstgenannten aufgrund ihrer elementaren Bedeutung für die weiteren Betrachtungen ausführlicher betrachtet werden.

Weiterhin gelten im Folgenden alle Werte für den gesunden Menschen. In der Praxis muss jedoch auch von behinderten Nutzern ausgegangen werden, die insbesondere in der Informationsaufnahme andere Grenzwerte haben. So ist bei vielen, vor allem männlichen Nutzern eine rot-grün Sehschwäche bis hin zu völliger Farbenblindheit festzustellen, die zwangsläufig Auswirkungen auf die Gestaltung von Bediensystemen haben muss.

2.1.1 Das visuelle System

Zur Ableitung von Anforderungen an Bediensysteme ist das visuelle System das mit Abstand wichtigste. Wie die Außenwelt tatsächlich existiert und was wahrgenommen wird, ist nicht dasselbe. Das Auge setzt dazu die physikalischen Größen in physiologische um. Tabelle 2.2 zeigt den Zusammenhang zwischen den physikalisch messbaren Größen und den dazugehörigen physiologischen Größen.

Licht als elektromagnetische Strahlung wird vom Auge von einer Wellenlänge $\lambda=380$ nm (violett) bis zu $\lambda=760$ nm (rot) (Bullinger 1994) erfasst. Die Adaption von dunkel auf hell erfolgt in wenigen Sekunden, von hell auf dunkel benötigt das Auge hingegen einige Minuten.

Mit sinkender Beleuchtungsstärke nimmt auch die Fähigkeit der Adaption ab (Herczeg 1994). Um Anstrengungen des Auges und damit Ermüdung zu minimieren, ist daher darauf zu achten, dass möglichst wenige Adoptionsvorgänge bei der Arbeit an Bediensystemen durchgeführt werden müssen. Dies erfordert eine intensive, gleichmäßige Arbeitsplatzbeleuchtung und eine Angleichung der Helligkeiten der Arbeitsmittel. Wird z. B. häufig die Blickrichtung von einer Papiervorlage auf den Bildschirm gewechselt, ist ein heller Bildschirmhintergrund mit dunkler Schrift (Positivdarstellung) zu bevorzugen. Aus gleicher Grund sind generell hell gefärbt-

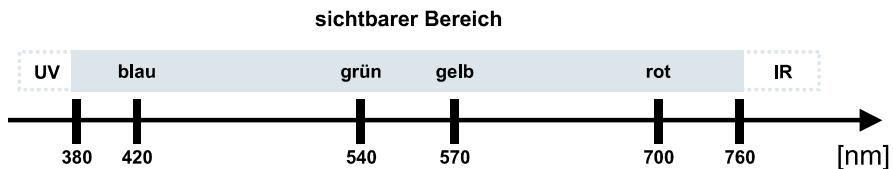


Abb. 2.2 Wellenlängen des sichtbaren Lichts

te Bedienelemente zu bevorzugen. Allerdings bestehen im produktionstechnischen Umfeld Blend- und Verschmutzungsprobleme. Hierfür hat sich als Bildschirmhintergrundfarbe ein helles Grau sehr bewährt.

Eine weitere wichtige Eigenschaft des Auges ist die Akkommodation, die die Fokussierung des Auges auf die betrachteten Gegenstände bewirkt. So können unterschiedlich weit entfernte Gegenstände scharf gesehen werden. Hierbei wird unter Sehschärfe die Fähigkeit verstanden, zwei nebeneinander liegende Punkte von einem bestimmten Abstand noch als getrennt wahrzunehmen. Der Bereich des scharfen Sehens beträgt weniger als 1° (Snyder 1988) (bei einer Sehentfernung von 500 mm sind dies umgerechnet nur 7,85 mm) und ist von der Beleuchtungsstärke abhängig. Die maximale Sehschärfe liegt bei einer Beleuchtungsstärke von 100 cd/m². Bei kleineren Beleuchtungsstärken nimmt die Sehschärfe ungleich mehr ab als bei größeren. Um Ermüdung des Auges durch Akkommodationsvorgänge zu vermeiden, sollten die Ebenen von Bildschirm, Tastatur und sonstigen Bedienelementen in etwa gleichem Abstand vor den Augen liegen.

Allerdings ist die Fähigkeit, Objekte in verschiedenen Abständen scharf sehen zu können, altersabhängig. Der sog. Nahpunkt, d. h. der Punkt mit der geringsten Entfernung, an der noch scharf gesehen werden kann, steigt ab einem Alter von 50 Jahren auf einen Wert größer als 500 mm (Herczeg 1994) (normale Arbeitsentfernung bei der Arbeit mit Bildschirmen), d. h. dass Bildschirmarbeit von dieser Personengruppe oftmals nur mit einer Sehhilfe durchgeführt werden kann. Die sehr beliebten Bifokal- und Gleitsichtbrillen sind hierfür i. A. ungeeignet, da sie eine unnatürliche und ungesunde Kopfhaltung im Lesebereich erfordern (sog. Geierhals).

Nach der Linse passiert das Licht den Glaskörper des Auges und trifft auf die Netzhaut (Retina), wo die äußeren physikalischen Reize mit Hilfe der Rezeptoren (Sehzellen) in innere physiologische umgewandelt werden. Bei diesen Rezeptoren werden zwei verschiedene Arten unterschieden, die aufgrund ihrer Eigenschaften verschiedene Funktionen besitzen:

- Zäpfchen sind für die Farbempfindung zuständig,
- Stäbchen dienen der Helligkeitsempfindung.

Die Fähigkeit, verschiedene Farben wahrnehmen zu können, erfolgt durch den Sehfarbstoff Rhodopsin, der in den Rezeptoren mit unterschiedlicher Konzentration enthalten ist und so jeweils Licht nur bestimmter Wellenlänge absorbiert. Die Maxima der Absorption des farbigen Lichts durch die Zäpfchen liegen bei $\lambda=420$ nm (blau), $\lambda=500$ nm (blau-grün, cyan), $\lambda=540$ nm (grün), $\lambda=570$ nm (gelb) und $\lambda=700$ nm (rot) (s. Abb. 2.2). Zur Bedeutungscodierung bei der Gestaltung von Bediensystemen eignen sich diese Farben daher besonders.

In der optischen Achse der Linse liegt auf der Netzhaut in der Netzhautgrube der gelbe Fleck (Fovea centralis), wo beim Menschen sehr dicht gedrängt nur Zäpfchen (ca. 300.000) stehen. Da die Zäpfchen allerdings aufgrund der geringeren Sehfarbstoffmenge weniger lichtempfindlich als die Stäbchen sind, können Farben bei schlechten Lichtverhältnissen (mangelnder Lichtstärke) nicht mehr erkannt werden. Die Zäpfchen sind daher für das Farbsehen als Tagsehsystem verantwortlich. Ausgehend von diesem gelben Fleck als Zentrum nimmt die Anzahl der Zäpfchen rapide ab. Gleichzeitig nimmt die Anzahl der Stäbchen zu. Da die Stäbchen eine höhere Sehfarbstoffmenge aufweisen, die sogar noch bei geringen Lichtstärken ansprechen, sind sie für das Dämmerungssehen (Helligkeitsempfindung) verantwortlich, ohne jedoch aufgrund der geringen Sehfarbstoffkonzentration Farben unterscheiden zu können. Daher ist in der Dunkelheit die Sehfähigkeit und vor allem die Farbsehfähigkeit eingeschränkt.

Aber auch die spektrale Empfindlichkeit der Stäbchen und Zäpfchen ist von der Beleuchtungsstärke abhängig (auch diese Eigenschaft fällt unter den Oberbegriff der Adaption). Beim helladaptierten Auge liegt die maximale Hellempfindlichkeit zwischen der Farbe Grün und Gelb, beim dunkeladaptierten Auge bei kleineren Wellenlänge, d. h. zwischen Blau und Grün (Bullinger 1994).

Durch die Trägheit der ablaufenden Prozesse in den Sehzellen (Rezeptoren) können nicht beliebig viele Einzelbilder pro definierter Zeiteinheit auch als Einzelbilder wahrgenommen werden. Daher verschmelzen ab einer bestimmten Bildwechselfrequenz zwei aufeinander folgende Bilder (Flimmern), d. h. das Erkennen der Einzelbilder ist nicht mehr möglich; damit wird dem Betrachter ein kontinuierliches Bild vorgetäuscht. Diese Frequenz liegt beim dunkeladaptierten Auge bei ca. 20 Hz, beim helladaptierten Auge dagegen bei ca. 70 Hz. Daher ist bei Bildschirmen für ermüdungsarmes Arbeiten beim Einsatz der Positivdarstellung (helladaptiertes Auge durch dunkle Schrift auf hellem Untergrund) eine Bildwiederholfrequenz von mindestens 70 Hz wünschenswert. Da bei der Verwendung von Halbbildern (Interlacing) nur jede 2. Zeile eines Bildes abwechselnd pro Zeiteinheit verändert wird, entstehen bei verschiedenen Mustern unerwünschte Effekte. Daher ist der Wechsel des gesamten Bildes günstiger zu bewerten. Durch ihre andere Arbeitsweise bieten LCD-Bildschirme auch bei niedrigeren Bildfrequenzen i. A. bessere Bilder als Röhrenbildschirme.

Die Farbe der gleichzeitig zu betrachtenden Objekte hat auch Einfluss auf die Sehschärfe. Durch die chromatische Abberation (Lichtbrechung abhängig von der Wellenlänge des Lichtes und damit von der Farbe) werden beim helladaptierten Auge grüne bis gelbe Objekte genau auf der Netzhaut abgebildet (aufgrund der höheren Empfindlichkeit des Auges für den Spektralbereich zwischen 500 und 580 nm), blaue Objekte vor und rote Objekte hinter der Netzhaut scharf abgebildet. Hieraus folgt, dass beispielsweise rote Zeichen auf blauem Grund oder umgekehrt eine permanente Umfokussierung des Auges erfordern und somit zu starker Ermüdung führen.

Für die Netzhautperipherie gilt, dass die Farbempfindlichkeit dort für die Farben Grün und Rot gering, für Blau und Gelb dagegen hoch ist. Daher sollten Grün und

Rot nicht für wichtige Objekte am Rande des Sehfeldes, d. h. auch nicht am Rande des Bildschirmes, verwendet werden.

Das Sehfeld des menschlichen Auges ist in der horizontalen und vertikalen Ebene verschieden ausgeprägt. Eine Auswertung ergibt einen Optimalbereich (zur Minimierung der Kopfbewegungen) von $+/- 15^\circ$ in horizontaler Ebene und zwischen 0 und 30° (nach unten) in vertikaler Ebene (Herczeg 1994).

Die Farbempfindlichkeit ist auch von der Ausdehnung des Objektes abhängig. Dies drückt sich in der Abhängigkeit der Farbempfindlichkeit von der Ortsfrequenz, d. h. von der Anzahl der Farbwechsel bzgl. des Betrachtungswinkels, aus. Das Maximum der Empfindlichkeit beim Schwarz-Weiß-Sehen liegt bei einer Ortsfrequenz von ca. 5 Farbwechseln pro Grad. Das entspricht z. B. der Breite eines Objektes von 1,5 mm bei einer optimalen Betrachtungsdistanz von 500 mm. Beim Farbsehen liegt das Maximum der Empfindlichkeit bei geringeren Ortsfrequenzen von ca. 0,25 1/Grad (abhängig von den Farben).

Dadurch werden große farbige Objekte besser wahrgenommen als kleinere. Farben werden deshalb besser eingesetzt, um globale Markierungen zu erkennen (keine einzelnen Buchstaben), während mittels Schwarz und Weiß auch feine Strukturen (einzelne Buchstaben, Schraffuren) gut erkannt werden können. Damit sollte bei der Gestaltung von Bediensystemen Farbe als Gestaltungsmittel nur sehr sparsam eingesetzt werden. Hier gilt die Empfehlung, bei der Gestaltung zunächst auf den Einsatz von Farben zu verzichten, diese dann erst später zur Verstärkung der gewählten Codierung hinzuzufügen. Die gleichzeitige Verwendung von sehr vielen Farben zur Informationscodierung ist aus physiologischer Sicht zu vermeiden.

Eine erste Verarbeitung des durch die Rezeptoren in ein physiologisches Signal umgewandelten Lichts findet bereits in der Netzhaut statt. Eine große Zahl neuronaler Elemente verbindet die Rezeptoren mit den Ganglienzellen. Die Ganglienzellen sind über den Sehnerv (Nervus opticus) mit dem Gehirn verbunden. Durch diese Schaltzellen findet eine erste Informationsverdichtung statt (die Informationen von 130 Mio. Sehzellen werden an lediglich 1 Mio. Ganglienzellen weitergeleitet). Verwertbare Aussagen über diese Informationsverdichtung liegen definitiv nicht vor. Es kann allerdings davon ausgegangen werden, dass bereits eine einfache Auswertung von Kantenrichtungen, einfachen Mustern, Winkeln etc. vorgenommen wird. Somit können hier schon Mehrdeutigkeiten und optische Täuschungen entstehen.

Tabelle 2.3 zeigt als Zusammenfassung die Eigenschaften des menschlichen Auges bei der Informationsaufnahme mit ihren ableitbaren Anforderungen an ein Bediensystem.

2.1.2 ***Das auditive System***

Das auditive System ist das zweitwichtigste Sinnessystem des Menschen in der Arbeitswelt. Das Ohr setzt bzgl. des Hörvermögens physikalische Größen in physiologische um. Daneben ist das Ohr auch für wichtige Grundfunktionen des Gleichgewichtssinnes zuständig, was hier aber nicht weiter vertieft werden soll.

Tab. 2.3 Eigenschaften des Auges und abgeleitete Anforderung an die Gestaltung von Bediensystemen

Eigenschaften des Auges	Anforderung an das Bediensystem
Mangelnde Adaption bei geringer Beleuchtungsstärke	Intensive Arbeitsplatzbeleuchtung
Ermüdung bei Adoptionsvorgängen	Angleichung der Helligkeiten der Arbeitsmittel (Positivdarstellung, helle Bedienelemente)
Sehschärfe abhängig von der Beleuchtungsstärke	Beleuchtungsstärke von 100 cd/m ² für Bildschirme
Ermüdung bei Akkomodationsvorgängen	Ebenen von Bildschirm und Bedienelementen in etwa gleichem Abstand vor den Augen
Große Zeitdauer von hell-dunkel Adaptionen	Hell-dunkel Adaptionen durch helle Bildschirme und gute Arbeitsplatzbeleuchtung vermeiden (d. h. nur mit helladaptierten Auge arbeiten); Positivdarstellung (dunkle Schrift auf hellem Grund)
Absorptionsmaxima des Lichts bei $\lambda=420$ nm, $\lambda=500$ nm, $\lambda=540$ nm, $\lambda=570$ nm und $\lambda=700$ nm	Zur Bedeutungscodierung Verwendung der Farben Blau, Cyan, Grün, Gelb, Magenta und Rot
Chromatische Abberation	Sparsamer Umgang mit Farben, gleichzeitige Verwendung von Farben außerhalb des grün-gelb-Bereiches vermeiden bei Anforderung des scharfen Sehens
Erkennen von Einzelbildern (Flimmern)	Bildwiederholfrequenz von mindestens 70 Hz bei Positivdarstellung
Farbempfindlichkeit der Netzhautperipherie gering für die Farben Grün und Rot	Wichtige Objekte am Bildschirmrand nicht in grüner oder roter Farbe darstellen
Farbempfindlichkeit abhängig von der Ortsfrequenz	Nur größere Objekte mit Farbe versehen
Unterschiedliches Sehfeld in horizontaler und vertikaler Ebene	Bedienelemente und Anzeigeelemente so anbringen, dass sie sich im Bereich von $+/- 15^\circ$ in horizontaler Ebene und zwischen 0 und 30° (nach unten) in vertikaler Ebene bezogen auf die normale, horizontale Sehachse befinden

Tabelle 2.4 stellt den Zusammenhang zwischen den physikalisch messbaren Größen und den dazugehörigen physiologischen Größen dar.

Der Mensch kann über das Ohr Schallfrequenzen zwischen ca. 16 Hz und 20.000 Hz aufnehmen (Mörike 1989), wobei die obere Grenze sehr vom Alter und der bereits erfolgten Belastung des Ohres abhängt. Sie kann in Extremfällen bis auf 5.000 Hz fallen. Das Maß für die Lautstärke eines Signals ist der Schalldruck. Er wird in Pa bzw. μ Pa gemessen. Der hörbare Bereich umfasst ca. $10^6 \mu$ Pa. Da diese Größe sehr unhandlich im Gebrauch ist, verwendet man lieber den Schalldruckpegel, der das logarithmische Verhältnis zwischen dem realen Schalldruck und dem Effektivwert des Schalldrucks der Hörschwelle bei 1 KHz $P_o = 20 \mu$ Pa darstellt. Somit ergibt sich für den Schalldruckpegel

$$L_p = 20 \log P/P_o \text{ dB}$$

Tab. 2.4 Physikalische und entsprechende physiologische Größen

Physikalische Größe	Einheit	Physiologische Größe
Schalldruck	Pa	Lautstärke
Schallfrequenz	Hz	Tonhöhe
Niedrige Schallfrequenz		Tiefer Ton
Hohe Schallfrequenz		Hoher Ton

Die bisher betrachteten Schallgrößen sind physikalische Größen. Sie sind also objektiv vorhanden und somit messbar. Die Lautstärke, mit der ein Mensch eine Schallstärke subjektiv empfindet, ist hingegen eine physiologische Größe. Sie wird in *phon* angegeben und hat die Herleitung

$$L_n = 20 \log P_{1000}/P_0 \text{ phon}$$

mit P_{1000} als dem Schalldruck eines subjektiv gleich laut empfundenen 1.000 Hz Tones, d. h. bei einem 1.000 Hz-Ton sind L_p und L_n gleich (Abb. 2.3).

Dieses Diagramm gilt prinzipiell nur für sinusförmige Töne. Wenn es sich hingegen wie in den meistens technischen Anwendungen um Frequenzgemische handelt, so benötigt man ein praktikableres Bewertungssystem. Hierzu wurde der bewertete Schallpegel eingeführt, bei dem von der IEC eine der physiologischen Wahrnehmung angepasste Filterkurve auf die gemessenen Schallpegel angewandt wird. Für das menschliche Gehör gilt die Filterkurve A und somit erhält die Einheit des Schalldrucks die Bezeichnung dBA oder dB (A). Mit diesem Bewertungssystem liegt die Hörschwelle bei 4 dBA, die Schmerzgrenze bei 120 dBA.

Die Hörschwelle, d. h. ab welchem Schalldruck ein Ton hörbar ist, ist frequenzabhängig. Diese Frequenzabhängigkeit ist bei niedrigen Frequenzen (tiefen Tönen) stark ausgeprägt, während sie zwischen 1.000 und 5.000 Hz kaum in Erscheinung tritt (Bullinger 1994). In diesem Frequenzbereich liegt die Hörgrenze fast konstant bei den bereits genannten 20 µPa. Auch die sog. Schmerzgrenze (120 dBA; Presslufthammergeräusch) ist frequenzabhängig. Sie liegt in im Bereich von 1.000 bis 5.000 Hz zwischen 2 und 20 Pa.

Eine normale Unterhaltung mehrerer Menschen bewegt sich in einem Frequenzbereich von ca. 125 bis 6.000 Hz und einem Schallpegel zwischen ca. 50 und 60 dBA. Messungen in Werkhallen zeigen häufig Schallpegel am Arbeitsplatz zwischen 70 und 90 dBA.

Tabelle 2.5 zeigt als Zusammenfassung die Eigenschaften des menschlichen Ohres mit ihren ableitbaren Anforderungen an ein Bediensystem.

Die Übertragung der von den Sinnesorganen in physiologische Signale umgewandelten physikalischen Größen in das Gehirn erfolgt durch Nervenzellen, sog. Neuronen. Aus den Eigenschaften dieses Transports können keine definitiven Anforderungen an die Kommunikation zwischen Mensch und Maschine abgeleitet werden.

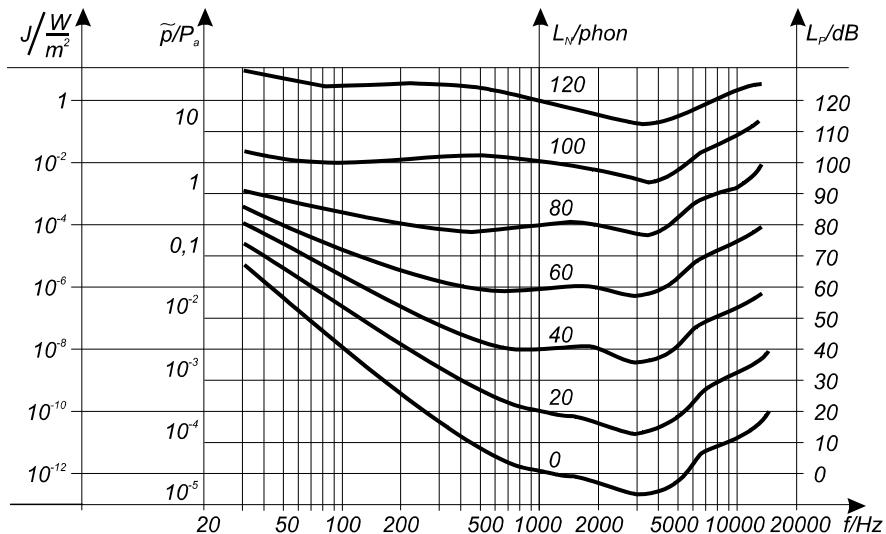


Abb. 2.3 Schallempfindlichkeit des menschlichen Gehörs (Kuchling 2001)

Tab. 2.5 Eigenschaften des Ohres und abgeleitete Anforderung an die Gestaltung von Bediensystemen

Eigenschaften des Ohres	Anforderung an das Bediensystem
Aufnehmbare Schallfrequenz zwischen 16 und 20.000 Hz, bzw. 5.000 Hz im Alter	Töne nur im Bereich von 250 bis 4.000 Hz einsetzen
Hörgrenze (4 dBA) frequenzabhängig	Töne im Bereich zwischen 1.000 und 4.000 Hz einsetzen; hier liegt die größte Empfindlichkeit des Gehörs
Hörbereich zwischen 4 dBA und (Hörgrenze und 120 dBA (Schmerzgrenze)	Systeme auf Lautstärkepegel zwischen 30 und 100 dBA auslegen

2.1.3 Das haptische System

Für die Interaktion mit technischen Systemen hat der Tastsinn des Menschen ebenso eine sehr große Bedeutung. Er setzt sich aus den zwei Wahrnehmungskomponenten Exterozeption und Propriozeption zusammen.

Die **Exterozeption** oder taktile Wahrnehmung erfolgt durch spezielle Sensortypen in der Haut (Cutis). Weil sie auf mechanische Verformung der Haut ansprechen, werden sie **Mechanorezeptoren** oder mechanorezeptive Fasern genannt (Goldstein 1997). Sie sind in der Lage, die **Sinnesmodalitäten** Druck und Berührung in bzw. auf der Haut wahrzunehmen. Neben diesen Tastkörperchen verfügt die Haut noch über eine Reihe weiterer Rezeptoren, die sich auf die Wahrnehmung von Temperatur (Thermorezeptoren) oder Schmerz (Nozizeptoren) spezialisiert haben, auf die hier nicht weiter eingegangen wird, da sie sich nach bisherigem Erkenntnisstand nicht bzw. nur sehr bedingt für die Mensch-Maschine-Interaktion eignen (Kurze 1998).

Die **Propriozeption** (Tiefensensibilität) betrifft den Schwere- und den Kraftsinn sowie die für Bewegungen zuständige Kinästhetik. Die **Propriozeption** ist kein einheitlicher Sinn, sondern integriert die Sinneseindrücke, die durch Reizung von Muskel-, Sehnen- und Gelenkmechanorezeptoren zustande kommen. Diese Rezeptoren ermöglichen es, sowohl die Stellung und Bewegung der Gliedmaßen, als auch die in den Muskeln aufgebrachten Kräfte zur Bewegung oder Fixierung des Körpers wahrzunehmen. Einen Beitrag zur Tiefensensibilität liefern ebenfalls die Hautrezeptoren der Exterozeption, welche bei Gelenkbewegungen durch Stauchung oder Dehnung erregt werden können. Insgesamt erreicht die Propriozeption eine hohe Sensibilität. So nimmt der Mensch bei den Schultergelenken noch Drehungen bis zu $0,2^\circ$ und Geschwindigkeiten bis herab zu $0,3^\circ/\text{s}$ wahr. Beim Fingergelenk sind es 1 und $12^\circ/\text{s}$. Der Kraftsinn erlaubt es dem Menschen, Gewichte und Kräfte in beiden Händen mit ca. 3 % Genauigkeit zu vergleichen (Völz 1999; Rühmann und Schmidtke 1992).

Den Oberbegriff für die exterozeptive und propriozeptive Wahrnehmung bildet die **haptische Wahrnehmung** oder auch als **Tastsinn** bezeichnet (Katz 1969; Matlin 1991). Ein wirkliches Erasten der Umwelt setzt beim Menschen die aktive Bewegung von Händen und Fingern voraus. Das heißt, es findet eine Bewegung statt (kinästhetische Reizung), bei der z. B. Oberflächen von Gegenständen durch Dehnung und Bewegung der Haut (taktile Reizung) erkannt werden. Das einfache Ablegen einer Fingerkuppe auf eine Oberfläche bewirkt beim Menschen noch keine Wahrnehmung einer Oberflächenstruktur. Erst wenn der Finger über die Oberfläche gleitet, kann festgestellt werden, ob sie z. B. glatt oder rau ist. Somit ist das „Erasten“ bzw. die haptische Wahrnehmung nur durch die Kombination von kinästhetischen und taktilen Informationen möglich (Kurze 1998; Zwiesler 1998).

Propriozeptive und exterozeptive Meldungen werden nicht oder selten bewusst wahrgenommen und sehr schnell unbewusst reflektorisch-motorisch im zentralen Nervensystem des menschlichen Körpers verarbeitet (Heuer 1987). Ohne bewusste Wahrnehmung bleiben vor allem die propriozeptiven Informationen aus Muskeln und Sehnen, während Gelenkrezeptoren für den Lagesinn zwar vorwiegend reflektorisch verarbeitet werden, aber bei Bedarf auch zu bewussten Wahrnehmungen beitragen (Gauer 1972). Die Regulationsvorgänge der unteren Ebenen sind zum Teil nicht einmal bewusstseinsfähig, d. h. das Bemühen um bewusste Erfassung stellt vielfach sogar selbst eine Störung der Bewegungsführung dar. Erst wenn Störungen auftreten, welche die Flexibilität der Unterprogramme übersteigen, sind Programmsprünge zu höheren Ebenen und später auch bewusste Eingriffe notwendig (Hacker 1978).

2.2 Wahrnehmung und Kognition

Für die menschliche Wahrnehmung und Informationsverarbeitung existieren eine ganze Reihe unterschiedlicher Betrachtungskonzepte (Johannsen 1993):

- das aufgaben- und tätigkeitsorientierte,
- das interaktionsorientierte,

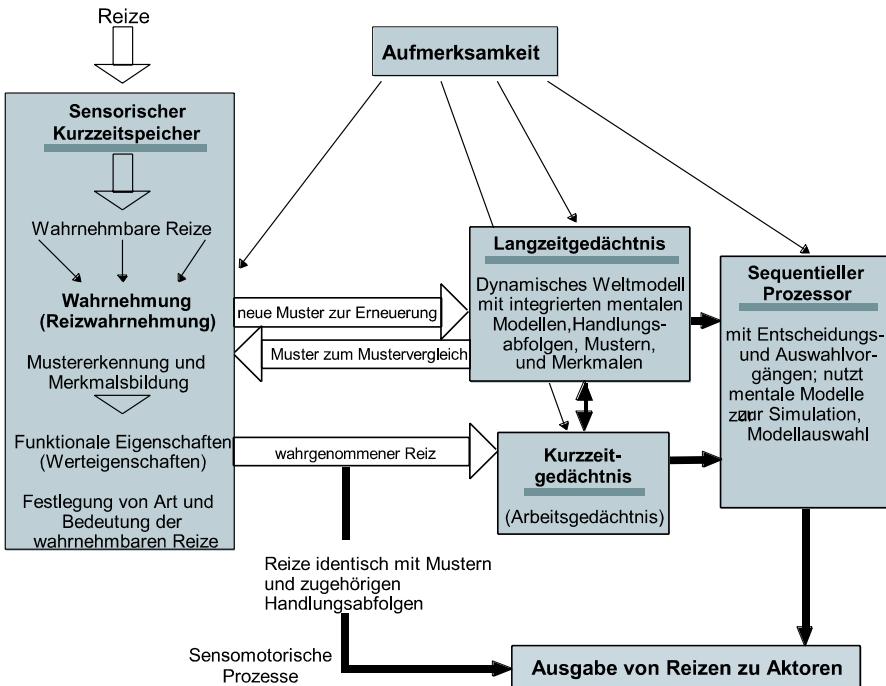


Abb. 2.4 Wahrnehmung und Informationsverarbeitung

- das verhaltensorientierte,
- das kognitionorientierte,
- das ressourcenorientierte sowie
- das strukturorientierte Konzept

Die umfassende Erläuterung aller Vorgänge der Wahrnehmung und Informationsverarbeitung mit Hilfe eines Konzeptes ist allerdings nicht möglich. Im Folgenden werden die hinsichtlich der Durchführung von Arbeitstätigkeiten und der kognitiven Verarbeitungsprozesse wichtigsten Konzepte erläutert.

Die in Abb. 2.1 dargestellten einzelnen Phasen der menschlichen Informationsaufnahme und -verarbeitung können weder deutlich voneinander getrennt noch unabhängig voneinander betrachtet werden (Muthig 1990). In Abb. 2.4 werden daher im Vergleich zu Abb. 2.1 die Phasen der Wahrnehmung, Kognition und des Handelns detaillierter sowie mit entsprechenden Abhängigkeiten dargestellt.

Reize, die durch die Sinnesorgane aufgenommen werden, gelangen in den als sensorischen Puffer (zwischen Sinnesorgane und Kurzzeitgedächtnis) arbeitenden sensorischen Kurzzeitspeicher (auch als Ultrakurzzeitgedächtnis bezeichnet). Informationen werden hier nur solange gespeichert (interindividuell verschieden, visuelle Informationen ca. 0,1–0,5 s; akustische dagegen etwas länger), wie sie für aktuelle mentale Tätigkeiten benötigt werden. Daher muss für eine ausreichende

Anzeigedauer von mehr als 0,5 s gesorgt werden, damit nochmals Reize aufgenommen werden können, falls bisher keine Wahrnehmung erfolgt ist. Aufgabe des sensorischen Kurzzeitspeichers ist die Bereithaltung der sensorischen Informationen für die Wahrnehmungsvorgänge der Mustererkennung und Merkmalsbildung. Es werden also keinesfalls alle auf die Sinnesorgane treffenden Reize auch wahrgenommen. Erst durch den Prozess der Wahrnehmung werden z. B. aus aufgenommenen Lichtreizen durch Mustervergleich mit im Gedächtnis abgelegten Mustern, Merkmalen und Modellen erst Formen und Informationen abgeleitet. Damit muss im sensorischen Kurzzeitspeicher ein genaues und vollständiges Bild von der Umwelt für die aktuelle Problemstellung vorhanden sein (Johannsen 1993).

Zur Definition einer Antwort auf die durch die Wahrnehmung identifizierten Informationen sind zentralnervöse Vorgänge verantwortlich, die zwischen möglichen Alternativen entscheiden. Dazu ist ein ständiger Informationsaustausch mit dem Kurzzeitgedächtnis (dem Arbeitsgedächtnis) notwendig, in dem die wahrgenommenen Informationen kurzzeitig (einige Sekunden bis maximal wenige Minuten) abgelegt werden können und das bzgl. der Kapazität der verschiedenen Gedächtnis- oder Speicherstufen den Engpass darstellt (ca. 5 bis 9 Chunks, s. Kap. 2.2.1). Nur durch ständige Wiederholung können Informationen dort über längere Zeit gehalten werden. Wichtige Informationen sind also länger anzuzeigen, damit sie im Kurzzeitgedächtnis auch über längere Zeit vorliegen. Dies hat auch Einfluss auf Informationseingabesysteme: z. B. muss bei der Eingabe einer längeren Auftragsnummer, die nicht im Langzeitgedächtnis gespeichert ist, häufiger nachgeschaut werden. Sowohl Speicherung als auch Zugriff erfolgen beim Kurzzeitgedächtnis wesentlich schneller als beim Langzeitgedächtnis, in dem Informationen länger andauernd festgehalten werden können. Informationen können vom Kurzzeitgedächtnis in das Langzeitgedächtnis wechseln und umgekehrt, d. h. die beiden sind miteinander verbunden. Dabei fällt das Einprägen von Informationen (mit Lernaufwand und erhaltenden Wiederholungen verbunden) in das Langzeitgedächtnis häufig leichter als das Erinnern, d. h. der Entnahme von Informationen. Nach heutigem Kenntnistanstand besitzt das Langzeitgedächtnis offenbar eine unbegrenzte Kapazität, das Problem der Nutzung dieser gigantischen Informationsmengen liegt vielmehr im Zugriff darauf. Wird der Bediener einer Maschine beispielsweise mit einem sehr selten auftretenden Fehler konfrontiert, so kann er sich in der Anfangsphase häufig nicht an eine solche Situation erinnern. Nach einiger Zeit des Nachdenkens und Sammelns weiterer Informationen „dämmert“ es ihm allmählich, um dann plötzlich die Problemlösung von damals klar vor Augen zu haben. Informationen, auch in Form von Vorstellungen (mentalnen Modellen, s. Kap. 2.2.2), sind sehr lange existent, sie gehen lediglich durch Abstrahierung teilweise verloren. Gespeicherte Informationen bleiben vor allem dann erhalten, wenn eine logische Verknüpfung zu anderen Informationen oder zum bewussten Handeln besteht, d. h. z. B. wenn sie für Aktionen benötigt werden. Damit existieren miteinander verknüpfbare, innere Repräsentationen von Informationen (z. B. durch Klassenbildung über gemeinsame Eigenschaften oder durch Strukturierung), die durch Mappings (Verbindungen) mit der Realität aufgefrischt werden. Dies bedeutet, dass durch die Bildung von Informationsklassen sowie die Darstellung der Zugehörigkeit von Elementen zu diesen

Informationsklassen die Wahrnehmung und Speicherung der Informationen durch leichtere Musterbildung und danach erleichterten Mustervergleich ermöglicht wird.

Die eigentliche Verarbeitungsleistung der wahrgenommenen Informationen wird durch den sequentiellen Prozessor erbracht. Er hat Zugriff auf Kurz- und Langzeitgedächtnis und nutzt damit die dort abgelegten mentalen Modelle als die interne Vorstellung über äußere Systemzusammenhänge (z. B. über die Funktionsweise technischer Einrichtungen), um diese zur Definition von Antworten durch eine interne Simulation (in Form einer kognitiven Vorwegnahme) mit Hilfe vieler sequentieller Einzeloperationen einzusetzen. Erst wenn die Ergebnisse dieser internen Simulation zufriedenstellend sind, werden die entsprechenden Lösungen in Form von motorischen Antworten an die Akteure weitergeleitet.

Allerdings sind zur Generierung von Antworten auf durch Prozesse der Wahrnehmung identifizierte Informationen nicht generell bewusste Entscheidungs- und Auswahlvorgänge notwendig. Bei den sog. sensomotorischen Prozessen können zu den wahrgenommenen Informationen quasi automatisiert motorische Antworten definiert werden, da diese direkt mit den Mustern im Langzeitgedächtnis abgelegt sind und durch den Vorgang der Wahrnehmung beim Mustervergleich aktiviert werden. Diese Prozesse sind bei hochgeübten Personen und damit auch bei Maschinenbedienern beim Verrichten von Routinetätigkeiten anzutreffen (Hacker 1986).

Die Ausführung der Antworten erfolgt durch das niedere Zentralnervensystem, wo verschiedene motorische Programme (eingeübte Bewegungsfolgen) abgespeichert sind, die dann abgerufen und zu den Akteuren geleitet werden.

Die Verteilung von Prioritäten für die beschriebenen Vorgänge und damit die übergeordnete Steuerung erfolgt durch die Aufmerksamkeit, die der Mensch individuell auf Wahrnehmung, Entscheidung und Antwortauswahl durch den sequentiellen Prozessor, das Arbeitsgedächtnis und die Antwortausführung legen kann. Diese Verteilung der Aufmerksamkeit kann zu menschlichem Fehlverhalten führen, indem z. B. durch mangelnde Aufmerksamkeit für die Wahrnehmung gewisse Sachverhalte im Kurzzeitgedächtnis verloren gehen. Sie werden beispielsweise nicht mehr aufgefrischt; z. B. das Übersehen einer wichtigen Zustandsanzeige, weil die Aufmerksamkeit auf das Einrichten eines neuen Werkzeugs gerichtet ist.

Im Folgenden werden die beschriebenen Teilprozesse bzgl. der Durchführung von Arbeitstätigkeiten und der kognitiven Verarbeitungsprozesse bei der Bedienung von Maschinen und Anlagen detaillierter beschrieben.

2.2.1 Wahrnehmung

Nicht alle auf die Sinnesorgane treffenden Reize werden vom Menschen auch tatsächlich wahrgenommen und damit zur weiteren Verarbeitung verfügbar gemacht. Die Wahrnehmung selbst ist ein komplexer Vorgang von Informationsaufnahme, Empfindung und integrativer Verarbeitung von Umwelt und Körperreizen (Pschyrembel 1990). Ziel der Wahrnehmung ist die Festlegung der Art und Bedeutung der aufgenommenen Reize, verbunden mit einer Klassifikation in neue, alte, wichtige

Tab. 2.6 Relation zwischen Chunkanzahl und Darstellungsform

Chunkform	Zahl	Wort	digital	Buchstabenfolge
Darstellungsform	12	Zwölf	1100	Flöwz
Chunkanzahl	1 (für die Zahl ein Chunk)	1 (für die Zahl als Wort ein Chunk)	4 (für jede Stelle der digitalen Darstellung ein Chunk)	5 (für jeden Buchsta- ben ein Chunk, da die Bedeutung aus dieser Kombina- tion nicht ersicht- lich wird)

und unwichtige, mit Hilfe von Filterung sowie die Weiter- und Neuentwicklung des im Langzeitgedächtnis abgelegten Weltmodells. Dieses Weltmodell enthält sowohl mentale Modelle, Handlungsabfolgen für sensomotorische Prozesse als auch Muster und deren spezielle Merkmale. Es stellt die Basis für erfolgreiches und individuelles Entscheiden und Verhalten dar. Die Filterung erfolgt durch Merkmalsbildung und Mustererkennung im Vergleich mit bereits im Langzeitgedächtnis abgelegten Mustern (s. Abb. 2.4). Diese Filterung ist alleine schon aufgrund der großen Differenz im Informationsfluss zwischen der Informationsaufnahme durch die Sinnesorgane von bis zu 10 Mbit/s (s. Tab. 2.1) und der maximalen Informationsverarbeitungskapazität des Gehirnes von 16 bit/s.

Dies bedeutet, dass z. B. Linien, die das Auge erkennt, erst nach dem Prozess der Wahrnehmung z. B. als Buchstaben durch Mustervergleich mit intern abgelegten und erlernten Mustern (Schablonen) verfügbar sind. Diese intern abgelegten Muster können auch mehr Information tragen ohne das Gehirn mehr zu belasten. Sie werden als **Chunks** (Bündel, Klumpen, Gedächtnisbrocken) bezeichnet. Die Erzeugung (das Erlernen) von Chunks und deren Wiedererkennung wird dann unterstützt, wenn

- Informationen nach logischen, zeitlichen, hierarchischen und ästhetischen Gesichtspunkten angeordnet sind und
- bei den aufzunehmenden Informationen bereits eine Übereinstimmung mit Gedächtnisinhalten und durch Übung erworbene Fähigkeiten existiert.

Daher ist die gewählte Darstellungsform für Informationen auf die vorhandenen Gedächtnisinhalte und bisher erworbenen Fähigkeiten abzustimmen, um die Anzahl der dafür zur Verfügung stehenden Chunks so gering wie möglich zu halten (s. Tab. 2.6).

Durch Lernen bzw. Training kann die Chunkanzahl und damit die Belastung des Gedächtnisses i. A. vermindert werden. Das Gehirn verlagert dabei die Erkennung der Bedeutung von der bewusstseinspflichtigen Ebene in die Ebene der Merkmalerkennung. Ein Vergleich der Lesezeiten der folgenden Sätze zeigt den Unterschied sehr deutlich.

Satz 1: Wie spät ist es?
 Satz 2: Etipäte? wssis

Obwohl wir in beiden Sätzen die gleichen Zeichen lesen, also auch die gleichen Rohdaten über die Augen aufnehmen, brauchen wir für den sinnlosen Text ein Vielfaches der Zeit des sinnvollen Satzes. Unsere Lernprozesse in früher Jugend haben dem Gehirn antrainiert, mehrere Buchstaben als Wort und im Zusammenhang als Bedeutung zu identifizieren. Eine andere in der Technik beliebte Methode zur Reduzierung der Informationsmenge stellen Icons bzw. Piktogramme dar. Sie können (wenn sie richtig ausgelegt sind) komplexe erlernte Bedeutungen bei minimalem Erkennungsaufwand ermöglichen.

Bei der Gestaltung von Bediensystemen ist daher die Chunkbildung zu fördern und zu erleichtern durch

- besondere Lernhilfen (z. B. durch Übereinstimmungen zwischen Realität und Abbildung – Mapping),
- einheitliche Zeichen und Begriffe für den gesamten Nutzerkreis (leichtere Erlernbarkeit) und
- eindeutige Wahrnehmbarkeit und Einschränkungen von Entscheidungs- und Handlungsmöglichkeiten des Bedieners.

Dazu ist es unbedingt erforderlich, dass sich der Entwickler des Bediensystems in die Welt des Nutzers hineinversetzt und aufbauend und unter Berücksichtigung dieser Welt die Gestaltung vornimmt.

Bei der Merkmalsbildung selbst werden aus sensorischen Reizen Merkmale extrahiert, um den Vergleich mit abgelegten Mustern in Bezug auf einen Bedeutungszusammenhang oder Kontext zu ermöglichen. Speziell in Situationen hoher Belastung unter Zeitdruck ist die Gefahr sehr groß, dass aufgenommene Reize in vorhandene Muster eingepresst werden, obwohl nicht alle Merkmale übereinstimmen. Daher können tatsächliche Reizbegebenheiten verfälscht werden. Eine eindeutige Zuordnung von physisch messbaren Größen und den psychisch wahrgenommenen Strukturen (d. h. der subjektiven Wahrnehmung) ist also nicht möglich; dies ist an sog. optischen Täuschungen nachzuvollziehen.

Bei den sensomotorischen Prozessen werden bereits mit wahrgenommenen Informationen vollständige Muster vorprogrammierter Handlungsfolgen abgearbeitet. Diese Muster werden vom Menschen kontext- und aufgabenbezogen erlernt. Eine aufgabenorientierte Gestaltung von Bediensystemen unterstützt somit das Erlernen der Handlungsmuster. So automatisiert sind diese Prozesse nicht mehr bewusstseinspflichtig; damit ist eine höhere Verarbeitungsgeschwindigkeit möglich (Johannsen 1993).

Sind solche Muster nicht vorhanden (z. B. bei erstmalig durchzuführenden Aufgaben oder bei ungewohnten Situationen), muss eine bewusste Informationsverarbeitung (Kognition) einsetzen, die alle zur Verfügung stehenden Informationen unabhängig von deren Nutzen, der zu diesem Zeitpunkt noch unbekannt ist, zunächst aufnimmt.

Um den Prozess der Wahrnehmung bei der Bedienung von Anlagen und Maschinen zu erleichtern und zu beschleunigen, ist dafür zu sorgen, dass zum einen entsprechende Muster und Merkmale schnell erlernbar und zum anderen schnell

wieder erkennbar sein müssen. Dies hat durch Orientierung an den Aufgaben, die der Mensch mit Hilfe des Bediensystems durchführt, zu erfolgen (da die Muster ebenfalls aufgabenbezogen erlernt werden). Hierzu gehört die deutliche Gruppierung zusammengehöriger Bedienelemente und Informationen, die über gemeinsame Merkmale verfügen (z. B. gleiche Informationsklasse oder gleiche Aufgabe) sowie die gleichzeitige Darstellung und damit Wahrnehmbarkeit der Informationen, die auch parallel benötigt werden.

Die Reize der einzelnen Sinne werden nach aktuellem Kenntnisstand parallel verarbeitet und dabei entsprechende Muster und Merkmale abgeleitet. Für die Folgerung von Bedeutungen werden dann jedoch in der Phase der Wahrnehmung die Merkmale mehrerer Sinne miteinander abgeglichen und integrierend ausgewertet. Ein Schalthebel wird visuell erkannt, haptisch erfasst und zugehörige Schaltgeräusche akustisch aufgenommen. Es lässt sich durch Versuche nachweisen, dass der Mensch zu deutlich geringeren Fehlerraten in der Bedienung technischer Geräte kommt, wenn sich ergänzende und teilweise redundante Reize über mehrere Sinneskanäle aufgenommen werden. Leider ist in den letzten Jahren bei vielen technischen Systemen der Trend zu verzeichnen, die Reize auf den visuellen und allenfalls noch minimal auf den auditiven Kanal („Beep“) zu reduzieren und somit weite Bereiche der menschlichen Wahrnehmung ungenutzt zu lassen.

Ein besonders interessantes Gebiet der Wahrnehmung des Menschen ist die Wahrnehmung räumlicher Informationen, die insbesondere in technischen Systemen eine wichtige Bedeutung hat (s. Kap. 7.4.2 autostereoskopische Bildschirme). Auf den ersten Blick verbindet man räumliche Wahrnehmung mit räumlichem Sehen und damit wiederum mit der Auswertung zweier – durch den Augenabstand – leicht versetzter aufgenommener Bilder. Doch die Realität sieht hier bei weitem komplizierter aus. Da wir auch mit einem abgedeckten Auge noch Aussagen zu räumlichen Gegebenheiten machen können wie z. B. *Dieser Stuhl steht hinter dem Tisch* muss das Gehirn offenbar über weitere Fähigkeiten zur Wahrnehmung von räumlichen Informationen verfügen.

Und in der Tat, es lässt sich wissenschaftlich zeigen, dass der Mensch allein über das visuelle System zwölf verschiedene Techniken nutzt, um Tiefe zu erkennen. Diese sind zu unterteilen in:

- Die **okulomotorischen** Tiefenkriterien, die die Stellung der Augen und die Anspannung der Augenmuskulatur auswerten. Diese Kriterien funktionieren vor allem bei sehr nahen Objekten im Abstand kleiner 1,5–3 m.
- Die **monokularen** Tiefenkriterien, die auch mit einem Auge funktionieren und beispielsweise Informationen über die Verdeckung oder Überschneidung von Objekten oder über die Helligkeits- und Schattenverteilung auswerten. Hierbei fließt vor allem die Erfahrung mit in die Auswertung ein. So geht das menschliche Gehirn stets von einer Lichtquelle aus, die von oben die Szene beleuchtet (Analogie zum Sonnenlicht) und erwartet dementsprechende Helligkeitsverteilungen und Schatten. Helle Objekte werden als näher angesehen als dunkle und bei Objekten mit ähnlicher Dimension (z. B. Autos) wird das kleiner abgebildete als weiter entfernt vermutet als das größer abgebildete.

- Die **bewegungsinduzierten** Tiefenkriterien, die zusätzliche Informationen über relative Bewegungen zwischen den Objekten und dem Betrachter auswerten. Sich schnell am Betrachter vorbeibewegenden Objekte werden als näher empfunden als sich langsam bewegende, zumal auch hier die typische Geschwindigkeit vergleichbarer Objekte aus der Erfahrung mit in die Erkennung einfließt.
- Die **Querdisparation**, die die zwei unterschiedlichen Bilder der beiden Augen auswertet, ist die offensichtlichste Technik der Tiefenerkennung, funktioniert aber nur im Nahbereich bis ca. 20 m.

Räumliche Wahrnehmung entsteht daher vor allem im Gehirn und nicht in den Augen. Erst die integrative Auswertung der oben genannten Kriterien unter Nutzung von Erfahrungswerten liefert den räumlichen Eindruck. Und dabei fließen natürlich auch entsprechende auditive Informationen mit ein, denn auch die Lautstärke von bekannten Schallquellen liefert Informationen über deren Entfernung.

Fasst man alle diese Erkenntnisse zusammen, so wird deutlich, dass die technische Nachbildung des räumlichen Sehens weitaus schwieriger ist, als nur zwei getrennte Bilder in unsere beiden Augen zu lenken. Letzteres kann man mit autostereoskopischen Bildschirmen oder Shutter-Glasses recht einfach bewerkstelligen, ein nutzbringender räumlicher Bildeindruck wird aber nur dann entstehen, wenn auch die anderen Tiefekriterien so berücksichtigt werden, dass sie der menschlichen Erfahrung entsprechen.

2.2.2 Kognition und Handeln

Nach erfolgter Wahrnehmung der Informationen sind diese in der Phase der Kognition bzw. des Erkennens weiter zu verarbeiten. Dies umfasst die Prozesse Verstehen, Lernen, Problemlösen, Entscheiden (Charwat 1994) und Planen. Sie werden je nach Komplexität und dementsprechend je nach ihrer kognitiven Inanspruchnahme, die individuell unterschiedlich eingeschätzt wird, auf verschiedenen Ebenen, den sog. Regulationsebenen menschlichen Verhaltens, durchgeführt.

2.2.2.1 Regulationsebenen menschlichen Verhaltens

Wenn der Mensch handelt, so lässt sich sein Verhalten nach dem Grad der mentalen Beanspruchung strukturieren. Nach dem dänischen Psychologen Rasmussen kann man dieses Verhalten in drei Ebenen einteilen (s. Abb. 2.5) (Rasmussen 1986).

Die **sensomotorische Verhaltensebene** ist geprägt durch erlernte und stark automatisierte sensomotorische Verhaltensweisen, bei denen unbewusste Programme ablaufen. Dazu gehören u. a. auch sensomotorische Prozesse (s. Abb. 2.4). Die Tätigkeiten, die auf dieser Ebene durchgeführt werden, zeichnen sich aus durch nicht bewusstseinspflichtige Abbilder und Bewegungsentwürfe und erfordern keine willentliche Aufmerksamkeit oder Steuerung.

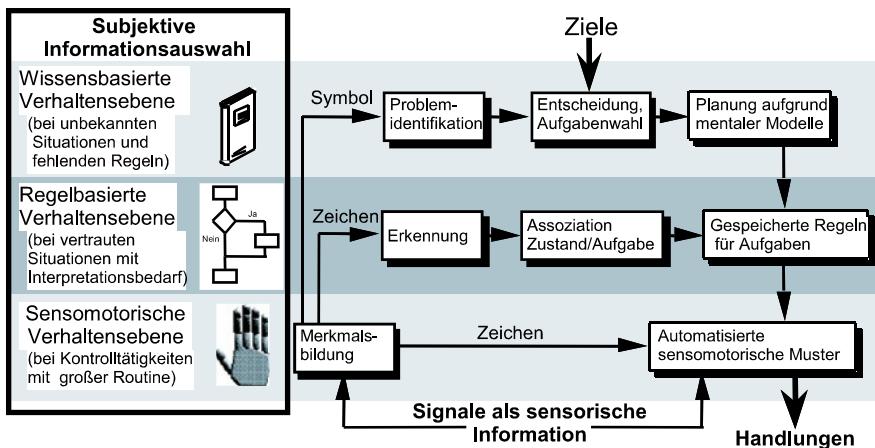


Abb. 2.5 Regulationsebenen menschlichen Verhaltens

Sie laufen somit im Unterbewusstsein als Routinetätigkeiten ab. Auf dieser Ebene werden Kontrolltätigkeiten in normalen Betriebssituationen (z. B. Überwachung von Prozess- und Maschinendaten wie der Maschinen- und Werkzeugzustände) sowie einfache Problemlösungstätigkeiten durchgeführt, die häufig auftreten und dadurch routiniert bearbeitet werden. Auch das intuitive Drücken des Not-Aus-Schalters nach einem Knall wird auf dieser Ebene abgearbeitet.

Tätigkeiten, die auf der **regelbasierten Verhaltensebene** ablaufen, sind gekennzeichnet durch erlernte Handlungsschemata, d. h. für die durch die Merkmalsextraktion gebildeten Zeichen (Merkmale) ist zunächst eine Verbindung zu ähnlichen, bekannten Aufgaben oder Zuständen herzustellen, für die bereits erlernte oder aus Instruktionen abgeleitete Regeln existieren. Damit ergeben sich die Eingangsinformationen aus einer (im Vergleich zur sensomotorischen Ebene) qualitativ anspruchsvolleren Informationsinterpretation und Reduktion. Zustände eines Systems werden so identifiziert und durch sukzessiven Vergleich mit gespeicherten Regeln (*wenn Zustand=..., dann führe aus ...*) mit auszuführenden Handlungen verknüpft. Diese Handlungen können dann durchaus der unteren sensomotorischen Ebene zugeordnet sein.

Sind keine geeigneten Regeln vorhanden oder ist der Zustand bisher noch nie wahrgenommen worden, muss auf der höchsten kognitiven Ebene, der **wissenbasierten Verhaltensebene**, eine entsprechende Regel entwickelt werden. Dazu werden die Informationen, die aus dem Umfeld aufgenommen werden, auf dieser Ebene als komplexe funktionelle und strukturelle Zusammenhänge in Form von Symbolen wahrgenommen (da der Zustand bisher unbekannt ist, können keine Zeichen aus der Merkmalsbildung extrahiert werden). Auf dieser Basis findet eine Identifikation des vorliegenden Zustandes statt, der nach bestimmten Zielgrößen, geordnet nach Prioritäten, bewertet wird. Daraus ergeben sich Entscheidungen für die zu lösende Aufgabe. Zu deren Bearbeitung werden die im Langzeitgedächtnis

abgelegten mentalen Modelle als eine innere Vorstellung über die funktionalen Eigenschaften und Zusammenhänge des technischen Systems herangezogen. Dies führt zu Planungsaktivitäten, die eine Auswahl aus bereits vorhandenen, zu modifizierender Alternativen oder die Neugeneration von Alternativen zum Ziel haben, um Regeln zu bilden und anschließend mit Hilfe entsprechender Handlungsentwürfe auszuführen. Hierzu ist eine intellektuelle Analyse zur Entwicklung von Plänen und Strategien (Hacker 1986) notwendig.

Entscheidungs- und Problemlösetätigkeiten im industriellen Umfeld werden hauptsächlich aufgrund von regelbasiertem Verhalten oder, wenn dieses nicht zum Ziel führt, aufgrund von wissensbasiertem Verhalten ausgeführt. In besonders kritischen Situationen wird vom Menschen ein angemessenes regel- und wissensbasiertes Verhalten erwartet. Allerdings kann beobachtet werden, dass in diesen Situationen mit hohem Informationsbedarf und auch großem Informationsangebot seitens des Bediensystems sowie dem Stress, dem der Bediener dabei nicht selten ausgesetzt ist, das menschliche Verhalten stereotyp wird und auf die unteren Verhaltensebenen – im Extremfall bis hinunter auf die sensomotorische Ebene – zurückfällt (DeKeyser 1986). Um dies zu verhindern, ist eine an den Aufgaben des Menschen in diesen Situationen angepasste Useware-Gestaltung zu fordern, indem genau die Informationen angezeigt werden, die zu dieser Situation benötigt werden. Bei hochkomplexen technischen Systemen wie Flugzeugen oder Kraftwerken werden die Bediener in regelmäßigen Simulatortrainings auf „Ready-to-use“-Regeln trainiert. Ziel ist es dabei, dass der Mensch auch in schwierigen Fehlerfällen seine Tätigkeiten auf der regelbasierten und sensomotorischen Verhaltensebene durchführen kann und damit schneller und mit geringerer Beanspruchung reagieren kann. Allerdings können nicht alle eventuell auftretenden Situationen trainiert werden; dies würde deren Kenntnis voraussetzen, was zumindest in komplexen Systemen nicht machbar ist. Daher müssen ebenso die höheren Regulationsebenen z. B. durch Training von Problemlösegrundstrategien gefördert werden (Blau 1987).

2.2.2.2 Mentale Modelle

Besonders für Prozesse, die auf der wissensbasierten Verhaltensebene ablaufen, ist das dynamische Weltbild mit vielen verschiedenen aufgabenbezogenen Modellen der Umwelt die Basis der mentalen Informationsverarbeitung. Diese Modelle werden als **mentale Modelle** bezeichnet. Sie sind somit persönliche Abbilder in Form einer inneren Vorstellung von Objekten, deren Funktion, vom dynamischen Verhalten technischer Systemen und des dazugehörigen Prozesses und von Verhaltensmustern und Handlungen der für einzelne Arbeitssituationen interessierenden Umwelt. Es gibt daher nicht **das** mentale Modell von einem Objekt oder einer Arbeitssituation.

Mentale Modelle sind nach (Norman 1983)

- *unvollkommen* – es werden immer nur eine begrenzte Anzahl von Eingangs- und Ausgangsgrößen berücksichtigt, da sie wie jedes Modell lediglich eine vereinfachte Darstellung der Realität sind,
- *unbeständig* – d. h. mentale Modelle können vergessen oder verändert werden,

- *nicht voneinander abgrenzbar*, d. h. ähnliche Modelle können sich vermischen,
- *unökonomisch*, d. h. unbenötigte Modelle bleiben trotzdem erhalten, und
- *werden nur dann eingesetzt*, wenn keine sensomotorischen Prozesse für die aktuelle Aufgaben- oder Problemstellungen vorliegen.

Die Modelle werden durch mentale Tätigkeiten wie Wahrnehmen, Schlussfolgern, Lernen, Entscheiden, Entwicklung von Kreativität, Bewerten und Weitergabe von Gedanken gebildet und sind damit von individueller Art. Aufgrund unterschiedlicher, persönlicher Vorstellungen und des Blickwinkels können sich dadurch unterschiedliche mentale Modelle über die Funktion einer technischen Einrichtung oder eines Prozesses ergeben. Da der Entwickler eines technischen Systems eine vertiefte Vorstellung über dessen Funktionsweise hat (**Entwicklermodell**), besitzt er auch ein anderes mentales Modell des gleichen Systems als ein Bediener (**Nutzermodell**), dem sich das System nur über das Bediensystem darstellt. Daher ist eine Angleichung der beiden Vorstellungen eine Grundvoraussetzung zur Gestaltung nutzerorientierter Bediensysteme. Eine Möglichkeit besteht darin, den Bediener bei der Bildung seines mentalen Modells zu unterstützen, indem die Funktionalität des technischen Systems am Bediensystem offensichtlich wird (Transparenz). Da der Entwickler eines Systems dieses Wissen über die Funktionalität bereits besitzt, stellt er andere Anforderungen an die darzustellenden Informationen und Bedienfunktionen durch das Bediensystem. Bei einer idealen Entwicklung ist das Nutzermodell mindestens ein Teil des Entwicklermodells, im Idealfall stimmen beide überein.

In der Praxis ist die Situation leider noch komplizierter, denn häufig werden komplexe technische Systeme von mehreren Entwicklern bearbeitet und auf der anderen Seite von mehreren Nutzern genutzt. Damit wird es mehrere wenn auch teils überlappende Entwickler- wie Nutzermodelle geben. Der Instandhalter einer Maschine benötigt häufig andere Kenntnisse über die Funktionsweise als der normale Bediener. Damit verfügen beide, Instandhalter und Bediener, auch über zumindest teilweise verschiedene mentale Modelle.

Bei der grundlegenden mentalen Aufgabe des Planens werden die mentalen Modelle genutzt, um durch eine interne Simulation mit Hilfe des sequentiellen Prozessors Erwartungen zu definieren, die dann anschließend mit den tatsächlich wahrgenommenen Informationen verglichen werden (Erwartungskonformität). Um diesen Vergleich zu ermöglichen, sind Rückmeldungen (Feedback) über Systemzustände und Bedienereingriffe unbedingt zu realisieren. Wird bei dem Vergleich eine Differenz festgestellt, werden weitere Tätigkeiten zunächst unterbrochen. Diese Differenz kann sich entweder aufgrund fehlerhafter Modelle oder aufgrund eines fehlerhaften Systems ergeben. Speziell in Situationen unter Zeitdruck und damit Stress neigt der Mensch dazu, von fehlerhaften Modellen und oder sogar von fehlerhaft angezeigten Informationen auszugehen. Die Generierung mentaler Modelle mit hoher Übereinstimmung zwischen Vorstellung und Realität wird durch Transparenz des Maschinen- und Prozessverhaltens gefördert. Trotzdem können Situationen mit mangelnder Übereinstimmung nie ganz ausgeschlossen werden.

Die Grundlage für eine weitgehende Übereinstimmung von Entwickler- und Nutzermodell stellt die Analyse des Nutzers mit seinen Arbeitsaufgaben dar. Da diese Analyse nicht nur aufwändig sondern obendrein mit dem klassischen Inge-

nieuwissen nicht machbar ist, wird sie in vielen Entwicklungsprozessen auch nicht durchgeführt. Die Entwickler glauben genug über den Nutzer zu wissen oder befragen bestenfalls Vertriebs- oder Schulungsmitarbeiter über die Wünsche des Kunden.

Es kann nicht intensiv genug betont werden, dass eine nutzergerechte Useware-Gestaltung ohne eine Nutzeranalyse nicht machbar ist!

2.2.2.3 Feedback

Die Handlungsausführung erfolgt auf den höheren Regulationsebenen des regelbasierten und vor allem des wissensbasierten Verhaltens nur dann, wenn der Mensch ein ausreichendes Feedback über das Systemverhalten erhält. Der Mensch führt einen Handlungsschritt durch und beobachtet das daraus resultierende Systemverhalten. Entspricht es seiner Erwartung, so fährt er mit der Handlung fort, entspricht es nicht seiner Erwartung, so wird er die Handlung überdenken, modifizieren oder verwerfen. Feedback ist also ein essentielles Merkmal gut gestalteter technischer Systeme.

Feedback kann dem Nutzer über ein oder besser mehrere Sinnesmodalitäten gegeben werden. Im einfachsten Fall zeigt der Prozess unmittelbar das gewünschte Verhalten und gibt so implizites Feedback. In vielen technischen Systemen benötigt der Nutzer allerdings Feedback über nicht oder nicht sofort offensichtliche Systemzustände. Wenn der Nutzer einer Werkzeugmaschine ein neues NC-Programm vom zentralen DNC-Server abruft, so kann der Vorgang durchaus einige Zeit dauern, ohne dass der Nutzer den Fortschritt seiner Handlung geschweige denn die Zielerreichung erkennen kann. Der Nutzer wird zunächst auf ein Feedback warten und nach kurzer Zeit entweder Handlungsalternativen prüfen oder schlichtweg die gleiche Bedienhandlung ein zweites Mal ausführen.

Strukturiert man den Feedbackbedarf, so lassen sich drei Ebenen unterscheiden:

- Das **Handlungsfeedback** gibt dem Nutzer eine sofortige Rückmeldung darüber, dass das technische System seine Handlung erkannt hat. Dies kann z. B. bei Tasten durch den sog. Knackfroscheneffekt taktile-auditiv erfolgen als auch z. B. auditiv durch einen kurzen Piepton.
- Das **Statusfeedback** gibt dem Nutzer eine Rückmeldung über den Grad der Zielerreichung. Insbesondere bei Systemreaktionen, die nicht offensichtlich sind und länger als 2 s dauern, sollte dem Nutzer ein Feedback z. B. in Form einer Sanduhr oder besser noch eines Verlaufsbalkens gegeben werden, aus dessen Darstellungsform er den weiteren Zeitbedarf ersehen kann. Im Beispiel des NC-Programms könnte man ihm ein Fenster einblenden mit dem Text „*NC-Programm GRX075 wird geladen*“ zusammen mit einer Balkenanzeige, die von 0 auf 100 % hochläuft.
- Das **Zielfeedback** meldet das Erreichen des Handlungziels an den Nutzer zurück und schließt damit den aktuellen Handlungsschritt ab. Im Beispiel der NC-Maschine würde ihm also angezeigt „*NC-Programm GRX075 wurde geladen*“.

Auf der obersten Regulationsebene des wissensbasierten Verhaltens sind Handlungen ohne entsprechendes Feedback überhaupt nicht mehr vorstellbar. Hier wird der Mensch Strategien zur Problemlösung anwenden, die einen permanenten Abgleich zwischen Handlung und daraus abgeleitetem Feedback erfordern. Nur aus dem Zusammenspiel von Handlung und Feedback ist der Mensch in der Lage, ein unbekanntes System oder einen unbekannten Zustand zu erkunden und sich ein Handlungsmodell zu erarbeiten. Diese Vorgehensweise setzt natürlich die Fehler-toleranz des technischen Systems voraus (Trial-and-Error Prinzip). Das System muss unbedeutendere Handlungsfehler tolerieren und den Nutzer vor irreversiblen schwerwiegenden Handlungsfolgen warnen.

Feedback ist jedoch nicht nur als Folge einer Bedienhandlung wichtig, sondern auch als eigenständige Information der Maschine über ihren Status. Insbesondere in hochautomatisierten Anlagen muss und möchte der Mensch über die wesentlichen Vorgänge informiert werden. Wenn die Anlagensteuerung einen Vorgang anstößt, so sollte dies dem Nutzer mitgeteilt werden.

2.2.2.4 Aufmerksamkeit

Der Mensch ist im Allgemeinen nicht in der Lage, einen hohen Grad an Aufmerksamkeit über einen längeren Zeitraum aufrecht zu erhalten. Bei Anlagen- bzw. Maschinenbedienern wird sich insbesondere im täglich wiederkehrenden Routinebetrieb ein langsam absinkender Aufmerksamkeitspegel einstellen. Und je seltener Bedienhandlungen werden, desto stärker wird der Mensch die notwendigen Schritte verlernen. Der Zugriff auf seine dafür notwendigen mentalen Modelle im Langzeitgedächtnis wird erheblich verlangsamt werden und die mentalen Modelle selbst werden mit der Zeit auf ein Mindestmaß verkümmern. Kommt es dann nach langer Zeit der Routine zu schwerwiegenden Fehlern, so wird der Mensch unter Stress geraten, was seine nunmehr geforderte mentale Leistung weiter reduziert.

Es ist daher in allen technischen Systemen angeraten, den Menschen in das Automatisierungskonzept so zu integrieren, dass er stets mental gefordert wird und jederzeit seine Handlungsmodelle abrufen kann („*Keep the operator in the loop!*“). Zusätzlich kann es – zumindest in Systemen mit einem hohen Gefährdungspotential – sinnvoll sein, den Bediener in regelmäßigen Zeitabständen auf wichtige Handlungsabläufe zu trainieren. Ein gutes Beispiel hierfür sind die Simulatortrainings der Piloten, in denen ihnen regelbasiertes Verhalten auch in äußerst selten vorkommenden Ausnahmesituationen antrainiert wird.

Zu einem gut gemachten System gehört daher auch eine Aufmerksamkeitssteuerung, die einigen Grundregeln folgt.

- Der Nutzer wird auch in Routinesituationen mit einem sinnvollen Maß an Informationen versorgt.
- Der Nutzer wird so in das Bedienkonzept integriert, dass er in regelmäßigen Abständen Aufgaben zu erledigen hat. Diese Aufgaben sollten kognitiv so anspruchsvoll sein, dass der Mensch die Sinnhaftigkeit seines Handelns akzeptiert.

- Treten Fehler in der Anlage auf, so sollte die Aufmerksamkeit – falls technisch akzeptabel – stufenweise gesteigert werden.
- Treten Fehler und Folgefehler in sehr kurzer Zeit auf, so sollte das technische System eine Prioritätensetzung und Filterung dieser Fehler mit dem Ziel vornehmen, den Menschen nur mit den aktuell wichtigsten Fehlern zu konfrontieren.
- Die Aufmerksamkeitsstufen sollten fest mit bestimmten Darstellungsattributen über mindestens zwei Sinnesmodalitäten (z. B. visuell und auditiv) verknüpft sein z. B. in der Form:
 - 1. Stufe: Hinweismeldung BLAU+1 Beep
 - 2. Stufe: Warnmeldung GELB+3 Beeps
 - 3. Stufe: Alarmmeldung ROT+Sirenenton

2.3 Menschliches Versagen

Nach schweren Unfällen erfährt man häufig aus den Medien, dass es sich um menschliches Versagen gehandelt hat. Die dem zugrunde liegende Logik ist so einfach wie sie falsch ist. Das technische System hat nach den Erkenntnissen weitgehend einwandfrei funktioniert, also hat der Mensch falsch gehandelt. Bei näherem Hinsehen entpuppen sich viele solche Unfälle als ein Versagen der Mensch-Maschine-Kommunikation. Seien es die Atomunfälle in Three-Mile-Island (1979) oder Tschernobyl (1986) oder Flugzeugabstürze wie der am Odilienberg in Straßburg (1992), meistens hat der Mensch die Meldungen des technischen Systems falsch erkannt, interpretiert und daraus falsche Handlungen abgeleitet. Es ist jedoch völlig falsch, ihm damit die alleinige Schuld an diesen Desastern zuzuweisen. In allen genannten Fällen waren die Mensch-Maschine-Schnittstellen so gestaltet, dass sie wesentliche Eigenschaften und Grenzen der menschlichen Informationsaufnahme und -verarbeitung nicht ausreichend berücksichtigt haben.

In der CIM-Euphorie der 80er-Jahre wurde der Mensch gerne als Fehlerquelle in automatisierten Prozessen angesehen und damit das Ziel begründet, den Menschen durch noch mehr Automatisierung zu ersetzen. Nach der Ernüchterung in den 90-er Jahren ausgelöst durch die Lean-Production-Philosophie wurde der Mensch dann wieder als der wichtigste Produktionsfaktor angesehen und die Ausrichtung der Technikgestaltung an seinen Fähigkeiten und Grenzen akzeptiert. Da die menschliche Handlungsweise auf Zielvorgabe, Planung, Handlung und Feedback über die Handlung und damit Zielerreichung basiert, müssen sich technische Systeme daran orientieren und Fehler des Menschen zur Systemexploration tolerieren.

Im Allgemeinen wird ein Fehler als die Abweichung eines Istwertes von einem vorgegebenen Sollwert bezeichnet. Innerhalb der Mensch-Maschine-Systeme werden Fehler im Zusammenhang mit Arbeitsaufgaben, d. h. mit zielorientiertem Handeln beschrieben. Somit geht es hier mehr um Fehlhandlungen. Da innerhalb komplexer Mensch-Maschine-Systeme ein Handlungsfehler schwerwiegende Folgen, z. B. technische Havarien oder Unfälle, auslösen kann, ist die Fehlervermeidung bzw. Fehlertoleranz ein wichtiges Ziel in allen technischen Systemen.

Tab. 2.7 Fehlertaxonomie

Kognitive Verhaltensebene	Fehler während des Schritts im Handlungsprozess		
	Ziele/ Planung	Gedächtnis	Rückmeldung
Wissensbasierte Verhaltensebene	<i>Wissensfehler</i> Fehler aufgrund von unvollständigem oder fehlerhaftem Wissen; Diskrepanz zwischen mentalem Modell und Realität. Bsp.: falsche Werkstoff-/Schneidstoffkombination ausgewählt		
Intellektuelle Verhaltensebene	<i>Denkfehler</i> Aufstellung falscher oder unrealistischer Ziele, Treffen falscher Entscheidungen. Bsp.: Fehlerhafte Planung der Bearbeitungsfolge beim Drehen einer Welle	<i>Merk-/Vergessensfehler</i> Fehlerhafte oder unvollständige Speicherung im Gedächtnis. Bsp.: Einschalten des Kühlmittels vergessen; Werkzeugmagazin nicht aufgefüllt	<i>Urteilsfehler</i> Fehlerhafte Interpretation von Systemreaktionen als Folge eigener Handlungen. Bsp.: Wegen falsch eingestellter Schneidparameter beginnt die NC-Maschine zu rattern; defekter Antriebsriemen wird aber als Ursache angenommen
Regelbasierte Verhaltensebene	<i>Gewohnheitsfehler</i> Anwendung eines falschen Handlungsmusters Bsp.: Anschaffung und Installation eines neuen Werkzeugs erspart Arbeitsgänge, die aber aus Gewohnheit ausgeführt werden	<i>Unterlassensfehler</i> Vorhandene Handlungsschemata gelangen nicht zur Ausführung, werden übersprungen oder zu spät aktiviert. Bsp.: Einstellen der Drehzahl vergessen. Informationsträchtige Umweltsignale als Rückmeldung auf eine eigene Handlung werden übersehen oder verwechselt	<i>Erkennensfehler</i> Bsp.: Übersehen einer Warnlampe
Sensomotorische Verhaltensebene	<i>Bewegungsfehler</i> bei weitgehend automatisierter Handlung (weder bewusstseinspflichtig noch -fähig). Bsp.: Beim Ausschalten versehentlich den falschen Taster betätigt		

2.3.1 Fehlertaxonomie

Wenn Fehler durch entsprechende Maßnahmen vermieden werden sollen, so muss man zunächst Fehler systematisch nach Ursachen- und Maßnahmenebenen klassifizieren.

Eine sehr differenzierte Klassifikation, die sog. handlungstheoretische Fehlertaxonomie (Zapf 1989), nimmt eine Einteilung der Fehlhandlungen bzgl. der Regulationsebene und dem Schritt im Handlungsprozess vor. Tab. 2.7 zeigt diese

Taxonomie, in der im Vergleich zu Abb. 2.5 zwischen der regelbasierten und der wissensbasierten Verhaltensebene eine weitere, die intellektuelle Verhaltensebene eingefügt ist. Auf dieser intellektuellen Verhaltensebene werden im Unterschied zur regelbasierten Verhaltensebene, wo die entsprechenden Regeln bereits bekannt sind, Handlungspläne und Handlungen mit einem Neuigkeitscharakter entwickelt.

Zur Gliederung des Handlungsprozesses wird für die hier dargestellte Taxonomie an dessen Anfang ein Ziel aufgestellt und ein entsprechender Plan zur Erreichung des Ziels entworfen. Dieser Plan wird anschließend ausgeführt; dies setzt allerdings dessen Speicherung im Gedächtnis so voraus, dass er jederzeit präsent ist. Bei und nach der Ausführung des Planes ist eine Rückmeldung erforderlich, ob die zuvor definierten Ziele erreicht wurden (Feedback). Die hier angegebenen Klassifikationen beziehen sich auf die Fehler bei der Bedienung technischer Systeme. Sie berücksichtigen daher nicht die Fähigkeit des Menschen, sog. Entwurfsfehler des Entwicklers, d. h. systematische Fehler, bei ausgeprägter wissens- und regelbasierter Verhaltensebene auszugleichen. Dies spricht u. a. für eine Mensch-Maschine-Kooperation und gegen eine Vollautomatisierung.

Als Beurteilungsmaß für menschliche Fehler bei verschiedenen Varianten eines Bediensystems wird die menschliche Zuverlässigkeit R als die Wahrscheinlichkeit des Nicht-Auftretens menschlicher Fehler herangezogen.

$$R = 1 - HEP$$

Wobei HEP (human error probability) die menschliche Fehlerwahrscheinlichkeit ist, die sich aus dem Quotienten der Anzahl der fehlerhaft durchgeführten Aufgaben zur Gesamtzahl aller durchgeführten Aufgaben ergibt. Die Anzahl der fehlerhaft durchgeführten Aufgaben sollte experimentell bestimmt werden. Die experimentelle Bestimmung ist jedoch bei ausgereiften technischen Systemen und gut ausgebildeten Nutzern aufgrund der geringen Fehlerhäufigkeiten eine zeitaufwändige Prozedur, die in der Praxis daher häufig als subjektive Einschätzung aus Experten-interviews ermittelt wird.

2.3.2 Fehlerursachen in komplexen technischen Systemen

Das Handeln eines Menschen erfolgt immer eingebettet in einen Kontext und ist in der Regel zielorientiert. Beim Umgang mit technischen Systemen wird der Mensch mit drei grundlegenden Systemeigenschaften konfrontiert, die ihm – wie der Psychologe Dietrich Dörner in seinem Buch *Die Logik des Misslingens* eindrucksvoll nachgewiesen hat (Dörner 2003) – häufig Probleme bereiten. Diese drei Eigenschaften sind (Abb. 2.6):

- die **Systemkomplexität**,
- die **Systemdynamik** sowie
- die **Systemintransparenz**.

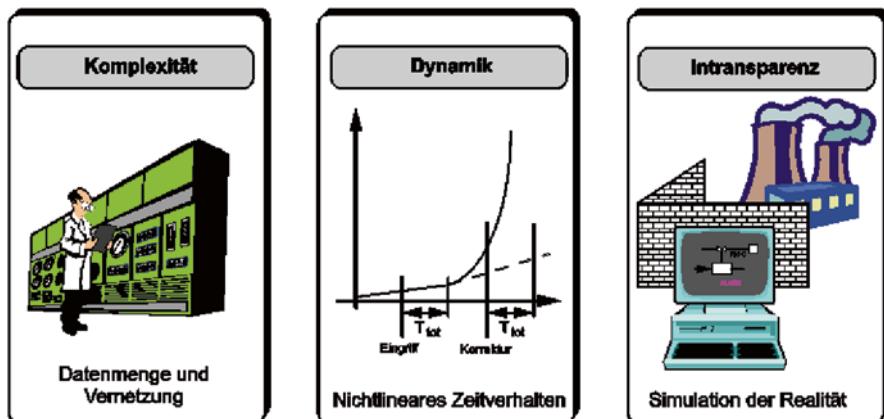


Abb. 2.6 Problemfelder menschlichen Handelns

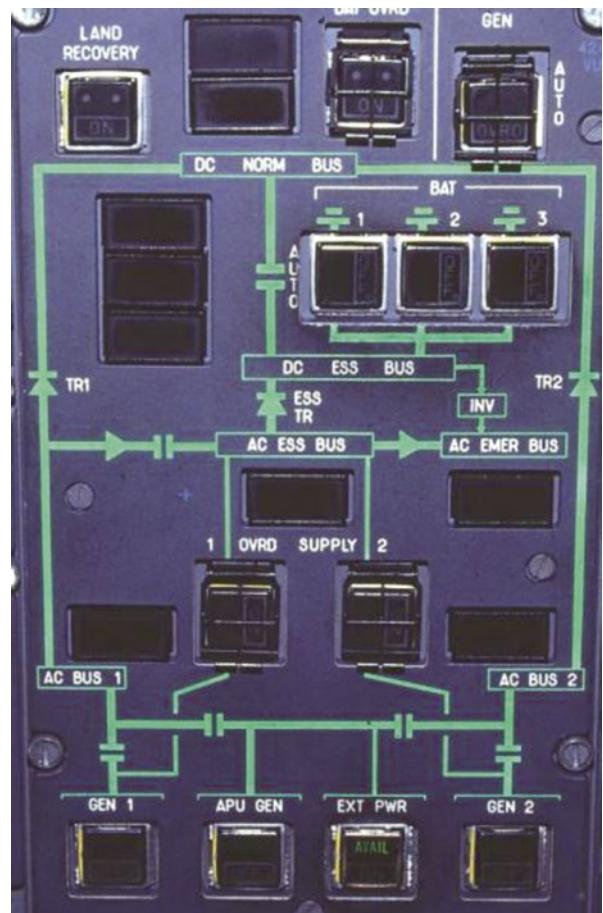
Ein **komplexes System** ist gekennzeichnet durch eine hohe Anzahl von Merkmalen sowie ihre gegenseitigen Abhängigkeiten, die vom Menschen zwar in ihrer Gesamtheit, jedoch nicht in ihren einzelnen Wechselwirkungen erkannt werden. Durch fehlendes Wissen über diese einzelnen Wechselwirkungen können falsche Schlussfolgerungen entstehen, die zu Fehlhandlungen führen, was insbesondere in kritischen Situationen schwerwiegende Folgen nach sich ziehen kann.

Die **Dynamik** von Systemen erschwert dem Menschen die Erfassung von Entwicklungstendenzen und somit das vorhersehbare Systemverhalten. Der Mensch neigt dazu, ein lineares Zeitverhalten zu antizipieren und somit auf der Basis einer Istanalyse linear zu extrapolieren. Leider zeigen sehr viele technische Systeme aber ein stark nichtlineares Verhalten, so dass eine lineare Extrapolation zu krassem Fehlern in der Situationsanalyse und damit zu Fehlhandlungen führen würde. Der Mensch benötigt daher Wissen über die dynamischen Zusammenhänge, um die richtigen Handlungen ableiten zu können.

Die **Intransparenz** ist ein typisches Phänomen moderner technischer Systeme. Konnte der Mensch in den alten mechanischen Systemen Vorgänge noch sehen, fühlen und hören, so berauben ihm die mikroelektronischen Systeme vielen seiner Sinneseindrücke. Die Systeme sind für den Menschen mehr und mehr eine Black-Box, deren Wirkungsweise er weder sieht noch hört und häufig auch nicht mehr versteht. Durch diesen Techniktrend wird der Mensch als Aktionspartner im Mensch-Maschine-System immer mehr vom Handlungsgeschehen abgekoppelt. Somit muss er auf einer unvollständigen Informationsbasis Entscheidungen treffen. Eine Zwangslage, die insbesondere bei überraschend eintretenden Zustandsänderungen des Systems zu Problemen führt.

Diese Systemeigenschaften machen deutlich, dass bei jeder menschlichen Entscheidung ein anteilmäßiger „Black-Box-Effekt“ nicht zu vermeiden ist. Der Anteil dieses Faktors der Ungewissheit steigt mit der zunehmenden Komplexität, Dynamik und Intransparenz eines Systems.

Abb. 2.7 Electrical Control Panel Airbus-A310



Für den Entwicklungsprozess kann man daraus ableiten, dass technische Systeme für den Nutzer transparent bzgl. ihres Verhaltens sein sollten. Da eine vollständige Transparenz bei komplexen mikroelektronischen Systemen allerdings kaum realistisch sein dürfte, sollten Entwickler versuchen, soviel Transparenz wie möglich in die Gestaltung der Useware zu bringen. Abbildung 2.7 zeigt das Electrical Control Panel eines Airbus A310, bei dem die Transparenz über die wesentlichen Systemzusammenhänge durch die Schalteranordnung in einem Übersichtsschaubild der elektrischen Anlage erreicht wurde.

Die Reduktion der Komplexität ist ein weiteres schwer zu realisierendes Merkmal. Entwickler neigen zu der Sicht, dass die Informationsmenge und ihre gegenseitigen Abhängigkeiten technisch erforderlich sind. Die langjährige Praxis zeigt hingegen, dass eine gute Nutzer-/Aufgabenanalyse zu einer deutlichen Reduzierung der notwendigen Informationen als auch durch bessere Strukturierung der Informationen zu einer deutlich geringeren Vernetzung führt. Da technische Störungen

häufig überraschend nach einer langen Routinephase auftreten, ist davon auszugehen, dass der Mensch das Problem zunächst auf der zeitintensiven wissensbasierten Handlungsebene angehen muss. Hierfür muss das System dem Menschen die zur Regulierung der Ausnahmesituation notwendigen Informationen situationsgerecht und mit ausreichender Handlungszeit anbieten. Dazu zählt bei komplexen Systemen die Schaffung übersichtlicher handlungsorientierter Einheiten und eine verständliche Darstellung dynamischer Systemzusammenhänge sowie eine am realen Handlungsgeschehen orientierte Systemabbildung, um eine Intransparenz des Systems sowie eine Informationsüberflutung zu vermeiden.

2.3.3 *Psychosoziale und emotionale Fehlerursachen*

Bislang wurde der Mensch mit seinen Fähigkeiten zur Aufnahme und Verarbeitung von Informationen sowie seinen kognitiven Prozessen vorgestellt. Nicht betrachtet wurden die psychosozialen und emotionalen Einflüsse auf das menschliche Verhalten. Bei den bisherigen Betrachtungen wurde davon ausgegangen, dass sich der handelnde Mensch in einem emotionalen Neutralzustand befindet. Dies ist aber in der Realität eher unwahrscheinlich. Jeder Mensch wird jeden Tag in einer anderen Stimmungslage an seinen Arbeitsplatz kommen. Und dabei werden zumindest ausgeprägte emotionale Zustände deutliche Einflüsse auf die Fehlerwahrscheinlichkeit haben. Ein Maschinenbediener, der sich privat großen Problemen gegenüber sieht, wird in seinem gesamten kognitiven Verhalten deutlich schlechtere Werte erzielen als ein Mensch in emotionaler Neutralität. Auch die in den letzten Jahren stark geförderte Teamarbeit hat Auswirkungen auf das Verhalten von Menschen in der Maschinen- und Anlagenbedienung. Während ein gut harmonierendes Team durch das gegenseitige Erkennen von Problemen Bedienfehlerraten deutlich senken kann, wird ein schlecht funktionierendes das Gegenteil bewirken. So ist das sog. „*Follow-the-Leader*“-Syndrom bekannt, welches beschreibt, dass in strikt hierarchisch beherrschten Teams Untergebene auch dann ihrem Chef folgen, wenn sie eigentlich schon von der Fehlerhaftigkeit seines Vorgehens überzeugt sind. Der Chef in einem solchen Team hingegen folgt dem „*I am the boss*“-Syndrom und wird selbst bei eigenen Zweifeln an seinen Handlungen diese nicht seinen Untergebenen zeigen wollen. Während sich diese Ebene des menschlichen Handelns einer ingenieurmäßigen Bearbeitung weitgehend entzieht und sicher auch in den meisten Systemen keine schwerwiegenden Folgen auf das Arbeitssystem haben wird, darf sie bei Arbeitssystemen mit hohem Gefährdungspotential wie Kernkraftwerken, Chemieanlagen etc. keinesfalls außer Acht gelassen werden. Hier sind Team-Management Konzepte und psychologisch begleitete Rollenspiele u. ä. wichtige Elemente der Störfallvermeidung.

Der Mensch strebt auch am Arbeitsplatz nach Ausgeglichenheit und Zufriedenheit. Dieses wird sich nicht einstellen, wenn der Mensch aufgrund einer schlechten Gestaltung seines Arbeitssystems Angst vor Fehlern hat und Stress empfindet.

Wird der Mensch durch zu viele Informationen in zu kurzer Zeit überfordert, so steigt sein Unwohlsein, es folgen daraus Stress und eine höhere Fehlerrate. Wird er

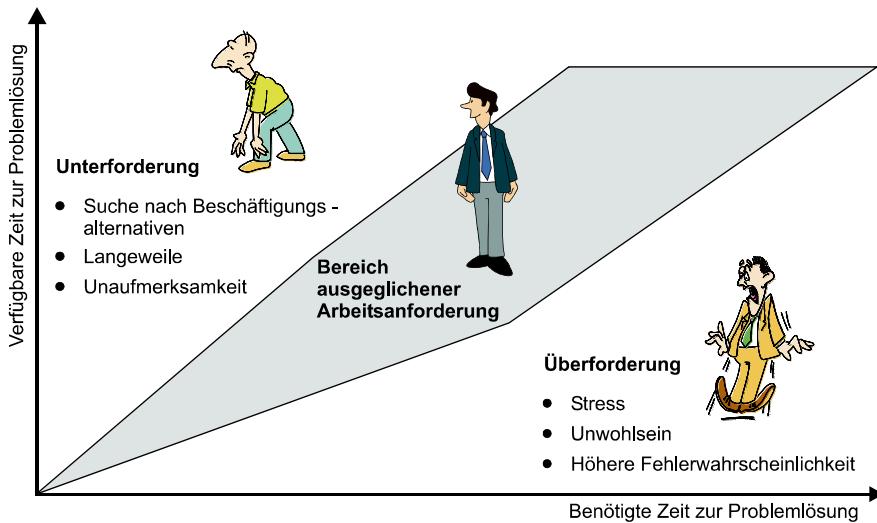


Abb. 2.8 Über-/Unterforderung des Menschen

andererseits z. B. durch die Automatik einer Maschine von zu vielen Tätigkeiten befreit, d. h. mit nur einer geringen Menge von Information pro Zeiteinheit belastet, so fühlt er sich ununterfordert (s. Abb. 2.8) und beginnt, nach Beschäftigungsalternativen zu suchen (Zeitung lesen, mit Kollegen unterhalten, schlafen etc.).

Zu einer menschengerechten Gestaltung von Mensch-Maschine-Systemen gehört deshalb auch, dass der Mensch mit einer ausreichenden und angemessenen Menge von Informationen wie auch Aufgaben versorgt wird und sich somit als einen wichtigen Teil des Prozesses sieht.

Kapitel 3

Vorgehensweise bei der Useware-Entwicklung

Die industrielle Praxis zeigt, dass die gestiegene Komplexität von Maschinen und Anlagen bislang häufig in ihrer komplizierten Bedienung Ausdruck findet. Die Bedeutung einer nutzerfreundlichen Bedienung ist daher gewachsen und wird auch weiterhin wachsen. Die Akzeptanz und Effizienz einer Maschine werden immer weniger durch den Funktionsumfang und immer mehr durch ihre Benutzbarkeit bestimmt, so dass sie zu einem wesentlichen Faktor für die Kaufentscheidung der Kunden geworden ist und einen wirtschaftlichen Faktor darstellt.

3.1 Vorgehensweise

Daher muss sich die Entwicklung von Maschinenbediensystemen an den Bedürfnissen der Nutzer orientieren und darf nicht, wie bisher, vom „flexiblen“ Menschen die Anpassung an die „starre“ Maschine fordern. In den folgenden Kapiteln wird eine systematische Vorgehensweise dargestellt, die die zukünftigen Nutzer der Maschine konsequent einbindet und es den Entwicklern ermöglicht, bereits in frühen Entwicklungsphasen die erarbeiteten Ergebnisse zu evaluieren. Diese Vorgehensweise hat sich bereits in vielen Useware-Entwicklungsprojekten bewährt und dazu beigetragen, neben der Gebrauchstauglichkeit des Maschinenbediensystems auch die Effizienz des Entwicklungsprozesses zu steigern.

3.1.1 Projektteam

Bei der Vorgehensweise ist zu beachten, dass im Umfeld der Produktionstechnik andere Rahmenbedingungen bestehen als bei der klassischen Software-Entwicklung. Useware-Projekte werden in der Regel in den technischen Entwicklungsabteilungen koordiniert, in denen vorwiegend Ingenieure arbeiten. Somit steht den Projektmitarbeitern eher technisches Know-how als Usability-Wissen zur Verfügung.

Für die Gestaltung eines Bediensystems sind aber neben der Technologie vor allem Eigenschaften und Charakteristiken der Nutzer von Bedeutung. Die Steuerungs- und Organisationsstrukturen der Bedienung müssen an deren Qualifikations- und Erfahrungsprofil sowie Aufgaben angepasst sein. Um dieser komplexen Aufgabenstellung gerecht zu werden, ist neben einer systematischen Vorgehensweise eine **interdisziplinäre Zusammensetzung** des Projektteams aus bspw. Ingenieuren, Informatikern, Designern und Psychologen nötig (Norman 1998).

Häufig sind in den vornehmlich kleinen und mittelständischen Unternehmen des Maschinen- und Anlagenbaus keine Usability-Experten vorhanden, die in die Entwicklung des Maschinenbediensystems einbezogen werden könnten. In diesen Fällen können ggf. entsprechende Kompetenzen in anderen Bereichen des Unternehmens ermitteln werden.

Wertvolle Beiträge zur Aufbereitung des Bediensystems können alle Abteilungen liefern, die in engem Kundenkontakt stehen (z. B. Service, Hotline, Anwendungslabor). Besonders hervorzuheben sind die Redaktionsabteilungen, in denen die produktbegleitenden Dokumentationen erstellt werden. Die technischen Redakteure sind durch ihre Ausbildung und Tätigkeit mit den Problemen und Eigenheiten der Nutzer vertraut und können so einen Teil der fehlenden Kompetenzen ersetzen (Cadera 2002).

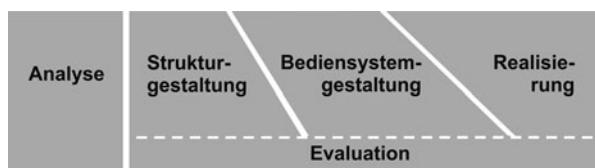
In vielen Fällen werden außerdem externe Berater in Useware-Projekte einbezogen. Dies liegt häufig darin begründet, dass „der Prophet im eigenen Hause nichts zählt“ und externe Projektmitarbeiter und Moderatoren die Durchsetzung von Projektergebnissen vereinfachen und die Effizienz des Projektablaufs verbessern.

3.1.2 Der Entwicklungsprozess

Die heutige Entwicklung gebrauchstauglicher Maschinenbediensysteme teilt sich in Hard- und Softwareentwicklung auf, wobei die Entwicklung von Software immer wichtiger geworden ist. Meist wird in der Praxis bei der Hardware auf bereits etablierte bekannte Komponenten zurückgegriffen. Der softwarebasierte Anteil der Useware am Gesamtsystem hat hingegen stetig zugenommen hat. Bei der Entwicklung von gebrauchstauglichen Maschinenbediensystemen hat sich in den letzten Jahren ein evolutionärer Prozess mit Nutzereinbindung etabliert. Im Gegensatz z. B. zum Wasserfallmodell, das häufig die Grundlage der Useware-Entwicklung bildet, sollten Nutzer an beliebiger Stelle im Entwicklungsprozess zum Testen und Bewerten der erarbeiteten Konzepte einzbezogen werden können. Dabei hat sich gezeigt, dass die Nutzerbefragungen selbst in frühen Konzeptstadien bereits gute Aussagen über die grundlegende Struktur und den Aufbau der Useware liefern. Deren Bedeutung ist sogar höher einzustufen als die der späteren grafischen Ausgestaltung (Geis und Hartwig 1998, Rauterberg 1994). Tabelle 3.1 fasst zusammen, welche Auswirkungen die konsequente Beteiligung der Nutzer für den Hersteller, den Entwickler und den Nutzer selbst hat.

Tab. 3.1 Gründe für die Nutzerbeteiligung

Nutzerbeteiligung		
Für den Hersteller	Für den Nutzer	Für den Entwickler
– Höhere Qualität des Produkts	– Verständnis für Möglichkeiten und Grenzen neuer Technologien	– Feedback über das Produkt
– Höhere Akzeptanz durch Nutzerorientierung	– Größere Identifikation bzw. höhere Akzeptanz	– Qualifizierungsmöglichkeit in Fachfragen
– Geringere Kosten durch Vermeidung von Nacharbeit/ Änderungen	– Höhere Motivation zur Benutzung	– Bessere Einsicht in die Arbeit der Nutzer – Größere Sicherheit bei der Lösungsfindung

Abb. 3.1 Useware-Entwicklungsprozess

Mit der gesteigerten Nutzerbeteiligung im Verlauf des Entwicklungsprozesses verlagert sich die bisher im Anschluss an die Entwicklung durchgeföhrte Evaluationsphase. Sie verläuft nun parallel zu den kreativen, gestalterischen Phasen. In Abb. 3.1 ist ersichtlich, dass die Vorgehensweise somit vier Phasen und fünf inhaltliche Arbeitspunkte umfasst (Reuther 2003; Bödcher 2007):

- Die **Analyse** dient der Erhebung der Nutzeraufgaben und Nutzeranforderungen.
- Die Ergebnisse der Analyse werden im Rahmen der **Strukturgestaltung** in ein plattformunabhängiges, generisches Benutzungsmodell überführt. In diesem ist die Interaktion des Nutzers mit dem technischen System in einer abstrakten, von der Realisierung losgelösten Form beschrieben.
- Entsprechend der ausgewählten Hard- und Softwareplattformen wird das Benutzungsmodell in der Phase der **Gestaltung** konkretisiert. In Form eines Grobkonzepts wird es an die Anforderungen und Eigenschaften der Plattformen angepasst sowie Menüstrukturen und Interaktionsobjekte erarbeitet. In der Feingestaltung werden diese detailliert und nach ergonomischen Richtlinien optimiert.
- Gleichzeitig mit der Feingestaltung beginnt die **Realisierung**, also die hard- und softwaretechnische Umsetzung des Konzepts, da eine Vielzahl von Wechselwirkungen zwischen diesen Phasen bestehen.
- Allen gestalterischen Phasen gemein ist die ständige **Evaluation** auf Basis von Prototypen und Strukturdarstellungen, die von repräsentativen Nutzern bewertet werden und so das Entwicklungsergebnis iterativ verbessern.

Die wesentliche Grundlage der ergonomischen Gestaltung von Useware liegt in den frühen Phasen des Entwicklungsprozesses: Analyse, Strukturgestaltung und Gestaltung sowie in der entwicklungsbegleitenden Evaluation. In der Realisierung werden dagegen final die einzelnen Bildschirmmasken und Dialoge manuell ausgestaltet, wozu in der Regel grafische Entwicklungsumgebungen eingesetzt werden. Diese

Entwicklungsumgebungen werden ebenfalls für die funktionale Realisierung der Maschinenbenutzungsoberfläche genutzt und gewährleisten somit einen nahtlosen Übergang der Ausgestaltung der Bediensystems und der eigentlichen funktionalen Realisierung.

3.1.3 Die Bedeutung eines Moderators

Die Praxis zeigt, dass in den Unternehmen des Maschinen- und Anlagenbaus zu meist weder die Kapazitäten noch das Methodenwissen zur Durchführung einer Nutzeranalyse vorhanden sind. Deshalb ist es hilfreich, für diese Aufgabe externe Experten zu Rate zu ziehen. Der externe Partner sollte über einen technischen Background und die notwendigen Methodenkenntnisse verfügen sowie Erfahrungen auf dem Gebiet der Gestaltung von Mensch-Maschine-Systemen vorweisen.

Als neutrale Befragungsperson erhält ein „Unbeteiligter“ im Allgemeinen leichter Aussagen zu Nutzer-Wünschen oder technischen Problemen der Anwender, die – wie die Erfahrung zeigt – oftmals wertvolle Informationen für das Unternehmen liefern.

Ein weiteres Problem stellen die häufig über lange Jahre gewachsenen Beziehungsgeflechte in den Unternehmen dar. Gerade bei einem Thema wie der Useware-Gestaltung, bei dem eine Vielzahl individueller Anforderungen und Wünsche harmonisiert werden muss, wird internen Moderatoren häufig Voreingenommenheit vorgeworfen. Oft entstehen daraus Spannungen oder Verweigerungshaltungen, die ein Projekt zum Scheitern bringen können.

Aus der Praxis: Über viele Jahre hat sich ein Unternehmen durch den Zukauf kleinerer Betriebe vom Maschinenhersteller zum Systemanbieter umstrukturiert. Dabei agieren alle beteiligten Geschäftsbereiche eigenständig und besitzen jeweils unterschiedliche Kompetenzschwerpunkte. Als Systemanbieter liefert das Unternehmen seinen Kunden ganze Anlagen, die sich vornehmlich aus Maschinen der eigenen Unternehmensbereiche zusammensetzen.

Aufgrund der Unternehmensstruktur und -geschichte wird der Kunde – obwohl dessen Anlage nur Komponenten eines Herstellers besitzt – mit den unterschiedlichsten Bedienpanels und Interaktionsphilosophien konfrontiert. Diese tragen mittlerweile zwar alle das gleiche Logo, letztlich wurden aber alle Systeme unabhängig voneinander entwickelt. Der Kunde hingegen möchte aber nicht nur bei einem Hersteller kaufen, sondern auch ein einheitliches Bediensystem erwerben („aus einem Guss“).

Wer soll nun den Useware-Vereinheitlichungsprozess moderieren?

Da die Vereinheitlichung für alle Bereiche Entwicklungsaufwand bedeutet, wird ein Moderator aus einem der Unternehmensbereiche sich immer mit dem Vorwurf konfrontiert sehen, nur die eigenen Interessen zu vertreten. Viele Projekte haben gezeigt, dass bei derartig Problemstellungen die

Integration eines externen Moderators mit entsprechendem Methodenwissen sogar unabdingbar ist, um einen tragfähigen Kompromiss zu erarbeiten, der dann auch Akzeptanz in den einzelnen Unternehmensbereichen findet.

3.2 Analyse

Die Analysephase ist der erste Schritt bei der nutzergerechten Entwicklung von Maschinenbediensystemen. Wie der Überblick in Abb. 3.2 verdeutlicht, werden in der Analyse zunächst die Interaktionspartner Mensch und Maschine untersucht. Im Vordergrund der Useware-Entwicklung stehen dabei die Analyse der Nutzer, deren Arbeitsinhalt sowie der Kontext der Arbeit. An zweiter Stelle steht die Analyse der technischen Anwendung, um Aussagen über Funktionalität und Einsatzgebiet abzuleiten.

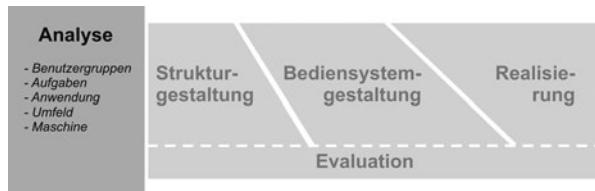
Die Analyse der bestehenden und zukünftigen Maschinenfunktionalität wird in Form eines Anwendungsmodells dokumentiert. Die Untersuchung der Nutzer und des Nutzerverhaltens erfolgt mit unterschiedlichen Analysemethoden und soll ein möglichst vollständiges Bild der Arbeitsaufgaben, Nutzergruppen und Arbeitsumgebung ergeben. Die Erfahrung zeigt, dass als positiver Nebeneffekt bei der Nutzeranalyse zumeist auch technische oder funktionale Verbesserungsvorschläge identifiziert werden, die in die technische Produktentwicklung nutzbringend einzbezogen werden können.

Sicher hat jeder Entwickler eine Vorstellung von den Nutzern, sei es aufgrund von Ausführungen des Service-Personals, des Vertriebs oder im Idealfall durch eigene Erfahrungen. Allerdings erhält der Entwickler meist ein rein subjektives Bild geprägt von einzelnen Kunden. In der Regel üben nur erfahrene Nutzer Kritik an der Bedienung oder fordern Sonderfunktionen für ihre speziellen Bedürfnisse. Die Berücksichtigung solcher speziellen Bedürfnisse führt wiederum zu einer Komplexitätserhöhung für die anderen Nutzer und damit nicht selten zu Entwicklungen, die am Markt vorbeigehen. Demgegenüber üben unerfahrene Nutzer selten Kritik am Bediensystem, wohl aus Angst sich zu blamieren oder als unfähig entlarvt zu werden.

Aufgrund dieser unzureichenden und subjektiven Informationen verlassen Entwickler sich aber allzu oft auf ihre Intuition und versuchen zu antizipieren, was für den Nutzer wichtig ist. Diese Intuitionen decken sich in der Praxis dann häufig nicht mit den Anforderungen der eigentlichen Nutzer.

Die Entwicklung von Bediensystemen sollte den Nutzer und seine Aufgaben in den Mittelpunkt stellen. 60 % der Nutzerprobleme im Bürobereich sind auf eine unzureichende Anpassung des Bediensystems an die Aufgabe zurückzuführen (Geis und Hartwig 1998), im Produktionsbereich ist mit ähnlichen Zahlen zu rechnen.

Abb. 3.2 Kernaspekte der Analysephase



Interessanterweise hat demgegenüber die visuelle Gestaltung, Anordnung und Platzierung von Dialogbausteinen ein geringeres Gewicht.

Diese Erkenntnisse lassen sich mühelos auch auf andere Gebiete übertragen. Nicht nur bei der Entwicklung von Bediensystemen werden die relevanten Aufgaben und Bedürfnisse der Anwender häufig außer Acht gelassen.

3.2.1 Der Nutzer, ein unbekanntes Wesen?

„Know the User“ (Shneiderman 2009) (= kenne deinen Nutzer) ist das erste und wichtigste Prinzip für die Entwicklung von Bediensystemen. Eine triviale Feststellung, die für das spätere Ergebnis aber von großer Bedeutung ist. Entwickler haben – wie bereits erläutert – oftmals die Vorstellung, den Nutzer und seine Bedürfnisse genau zu kennen. Sie haben das Gefühl, typische Anwender zu sein, aber sie sind es in der Regel nicht (Norman 1989). Daraus resultiert die in Abb. 3.3 dargestellte Problematik, bei der die Maschine eine Barriere zwischen Nutzer und Entwickler darstellt. Der Nutzer muss dabei auf Basis seiner Vorstellungen und Rahmenbedingungen mit einem technischen Gerät umgehen, das ein Entwickler auf Basis völlig anderer Vorstellungen und Rahmenbedingungen entwickelt hat. Diese Barriere können nur Entwicklungsansätze überwinden, die eine ganzheitliche Betrachtung des Mensch-Maschine-Systems beinhalten und nicht nur einfach die Gebrauchstauglichkeit testen.

Erfolgreiche Entwickler sind sich bewusst, dass Menschen auf ganz unterschiedliche Art und Weise lernen, denken und Probleme lösen. Der Entwicklungsprozess für ein Bediensystem muss darum in jedem Fall die Analyse der zukünftigen Nutzer, ihrer Aufgaben und des Arbeitsumfelds voranstellen. Es ist geradezu die Voraussetzung, um die Arbeit überhaupt aufnehmen zu können.

Für Anfänger spielt insbesondere die „Leichtigkeit des Lernens“ (ease-of-learning) eine große Rolle; für Experten ist es eher die „Leichtigkeit des Gebrauchs“ (ease-of-use), also wie schnell eine Bedienung oder der direkte Zugriff auf Information erfolgen kann. Häufig identifizieren sich die Entwickler eher mit dem Experten und vergessen dabei den Anfänger. Zwar ist die Interessenslage in vielen Fällen gleich, doch gibt es Unterschiede wie z. B. bei den Navigationsbedürfnissen:

Ungelernte Operatoren bzw. Anfänger kommen besser mit der relativen Navigation zurecht. Die Vorgehensweise der relativen Navigation „vom Groben zum Detail“ ist für sie geeignet, um ihnen über den jeweiligen Sachverhalt zunächst einen

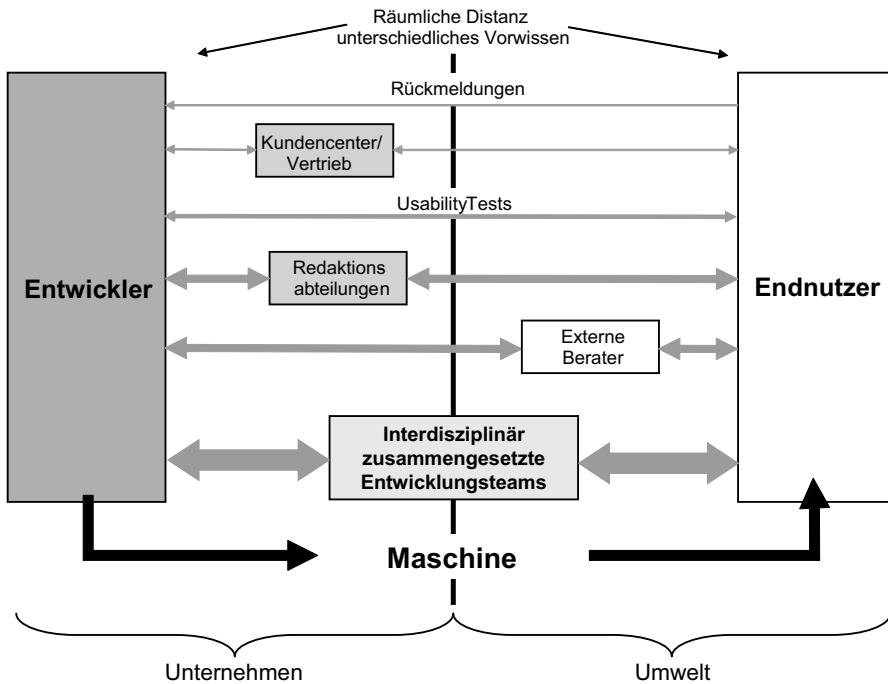


Abb. 3.3 Der Kontext des Entwicklungsprozesses

Überblick zu geben. Dies unterstützt das „Erforschen“ des Systems. Der Techniker bzw. Experte benötigt häufig einen schnellen Zugriff auf die Details, die er im Rahmen seiner Arbeitsaufgabe manipulieren oder einsehen muss. Dies wird durch eine absolute Navigation besser unterstützt, da diese es ihm ermöglicht, direkt die Seiten aufzurufen, auf denen die benötigten Informationen zu finden sind. Dieses muss bei der Auslegung des Bediensystems beachtet werden.

Im Gegensatz zum Konsumbereich gibt es in der Regel nicht nur eine Gruppe von Nutzern, sondern mehrere Nutzergruppen mit unterschiedlichen Aufgaben und Informationsbedürfnissen. Das Bediensystem sollte so gestaltet sein, dass es alle Nutzergruppen unterstützt, vor allem aber den Nutzer, der primär mit der Anlage umgeht, d. h. die meiste Zeit am Bediensystem verbringt. Dies ist im Produktionsumfeld in aller Regel kein Ingenieur sondern ein Werker.

3.2.2 Wie gehe ich konkret vor?

Es stellt sich die Frage, wie die Nutzer am besten in den Entwicklungszyklus integriert werden können, um ein optimales und auf die Nutzer zugeschnittenes System entwickeln zu können.

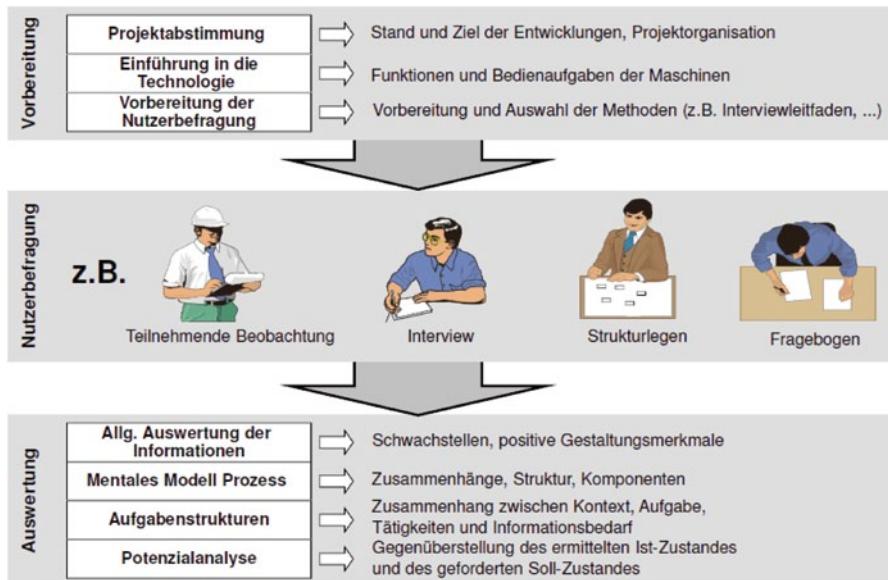


Abb. 3.4 Ablauf der Analysephase

Die Nutzer sollten so früh wie möglich in den Entwicklungsprozess integriert werden. Das Fundament für die Nutzerbeteiligung bildet die Analysephase, die durch eine besondere Berücksichtigung des Kommunikationspartners Mensch gekennzeichnet ist. Darüber hinaus sollte im Sinne der zyklischen Evaluation eine weitere Beteiligung von Nutzern im Entwicklungsprozess berücksichtigt werden.

Die Analysephase kann prinzipiell in die drei Bereiche Vorbereitung, Datenerhebung sowie Auswertung und Dokumentation gegliedert werden (s. Abb. 3.4).

In der Vorbereitungsphase erfolgt für die Projektmitglieder, die nicht über Entwicklungskenntnisse verfügen, zunächst eine Einführung in die Technologie. So ist es möglich, ein Verständnis für den Prozess und sein Umfeld zu entwickeln. Darüber hinaus gehört auch eine erste Definition von Nutzergruppen aus Sicht der Entwicklung, des Service und der Marketing-Abteilung dazu: Da ein Bediensystem als Eingriffsmöglichkeit für verschiedene Nutzer mit ebenfalls verschiedenen Arbeitsaufgaben dienen kann, werden hier grob die Arbeitsaufgaben definiert und den entsprechenden Nutzergruppen zugeordnet.

Aus der Praxis: Häufig gestaltet sich die Vorbereitung einer Befragung schwieriger, als es zunächst den Anschein hat. Ist es noch relativ einfach, repräsentative Kunden auszuwählen, so muss dann der Kontakt zu ihnen hergestellt werden. Dies erfolgt in der Regel über die Vertriebsmitarbeiter (Key Accounter), die es in den wenigsten Fällen tolerieren, wenn ohne ihr Wissen Kundenkontakt aufgebaut wird. Das bedeutet, dass zunächst der Vertriebsmitarbeiter vom Sinn und Zweck der Kundenbefragung überzeugt werden muss und dieser dann seinerseits den Kunden dafür gewinnen muss. Der Zeitrah-

men, der benötigt wird, um auf diesem Wege Termine beim Kunden zu erhalten, hängt also vor allem vom Engagement der Vertriebsmitarbeiter ab und kann eine Woche aber auch zwei Monate in Anspruch nehmen. Es hat sich als hilfreich erwiesen, dem Vertrieb hierzu aussagekräftige Kurzinformationen über Ablauf und Zweck der Befragung bereitzustellen.

Um die Kosten in Grenzen zu halten, müssen die Kundengespräche synchronisiert werden, was einen zusätzlichen Organisationsaufwand mit sich bringt. Häufig möchten die Vertriebsmitarbeiter bei den Befragungen dabei sein. Um die Neutralität der Befragenden zu wahren, sollten diese die Befragungen dennoch alleine durchführen. Im Idealfall kann der Vertriebsmitarbeiter die Vorgesetzten der Befragungsteilnehmer „beschäftigen“, so dass diese nicht die Befragung der Mitarbeiter „überwachen“ und somit deren Gesprächsbereitschaft behindern.

Beachtet werden sollte auch, dass nicht mehr als drei Werkstage einer Woche für Befragungen vorgesehen werden sollten. Nur so bleibt ausreichend Zeit, um die Befragungsergebnisse sauber zu dokumentieren. Außerdem sinkt mit steigender Anzahl an Befragungen in kurzer Zeit die Konzentrationsfähigkeit der Befragenden. Wichtig ist außerdem, dass auch kulturspezifische Faktoren berücksichtigt werden. Besonders in asiatischen Ländern, aber auch in Ländern wie den USA bestehen andere gesellschaftliche Rahmenbedingungen und werden andere Umgangsformen gepflegt.

Auf dieser Basis werden die Ziele und Inhalte der Analysephase abgestimmt. Dies beinhaltet die Abgrenzung der Arbeitsinhalte, die Methodenauswahl sowie die Projektplanung (Zeit/Ressourcen). Im Anschluss hieran kann die Vorbereitung und Organisation der Befragung beginnen. An dieser Stelle sei darauf verwiesen, dass der Aufwand für die Koordination der Befragung häufig unterschätzt wird. Diese gute Vorbereitung ist allerdings der Garant für eine reibungslose Befragung.

Sind alle Vorbereitungen abgeschlossen, so kann die Datenerhebung, d. h. die Nutzerbefragung erfolgen. Entsprechend der ausgewählten Methoden werden die notwendigen Daten vor Ort beim Anwender erfasst, protokolliert und aufbereitet. Nach den ersten 2–3 Befragungen (Pretest) sollten die daraus abgeleiteten Ergebnisse mit den benötigten Informationen verglichen werden, um bei Bedarf Änderungen in der Methodik oder dem Vorgehen vorzunehmen. In Anschluss an dieses Review werden die restlichen Nutzer befragt. Nach Abschluss der Befragungsreise werden die gesammelten Daten ausgewertet, interpretiert und entsprechend dokumentiert.

3.2.3 *Wen befrage ich?*

Im Rahmen des Abstimmungsgespräches stellt sich als erstes die Frage, wer die Nutzer sind, ihr Profil, ihre Kenntnisse, ihr Vorwissen. Wie schon angesprochen, gibt es im Produktionsumfeld nicht nur eine Gruppe von Nutzern, sondern in der

Regel mehrere. Als erstes ist es deshalb notwendig, eine grobe Nutzergruppeneinteilung im Gespräch mit Entwicklern und/oder Vertriebsmitarbeitern vorzunehmen. Nachdem dies geschehen ist und Zielgruppen für das Bediensystem identifiziert worden sind, kann eine konkrete Auswahl der zu befragenden Nutzer bzw. Unternehmen erfolgen. Diese Auswahl hat eine entscheidende Bedeutung für die Qualität der Ergebnisse und damit die Gestaltung und Ausrichtung des Bediensystems.

Anzahl der zu befragenden Nutzer? Insgesamt sollte aus jeder Nutzergruppe ein möglichst repräsentativer Querschnitt an Nutzern betrachtet werden. Je nach Homogenität der Nutzergruppen sind nach eingängiger Forschungsmeinung mindestens 5 Nutzer je Nutzergruppe zu befragen, in der Praxis hat sich eine Anzahl zwischen 8 und 10 Nutzern als optimal herausgestellt (s. auch Kap. 3.6.2).

„Alte“ oder „Neue“ Nutzer? Entgegen der allgemeinen Vermutung sind gerade die „alten“, erfahrenen Nutzer nicht unbedingt für die Befragung geeignet. Sie haben sich bereits so stark an eine Maschine gewöhnt, dass Sie in der Regel keine Probleme mehr mit der Bedienung haben und sich auch nicht mehr an ihre anfänglichen Probleme erinnern können. Darüber hinaus sind viele Arbeitsabläufe mental derart automatisiert, dass sie nicht mehr in der Lage sind, Arbeitsabläufe genau zu beschreiben. Allerdings sind auch ganz „neue“ Nutzer nicht für eine Befragung geeignet, da sie noch nicht genügend Erfahrungen im Umgang mit der Maschine besitzen und so den vollen Funktionsumfang der Maschine noch nicht überblicken können.

Aus der Praxis: Die Befrager gehen häufig davon aus, dass die befragten Maschinenbediener mit großer Begeisterung an einer Befragung teilnehmen. Leider ist dies eher die Ausnahme. Maschinenbediener fühlen sich häufig durch die Beobachtung und Befragung gestört und reagieren entsprechend. Viele Probleme entstehen, weil die Befragten nicht wissen, wofür sie an der Befragung teilnehmen und was mit den erhobenen Daten passiert.

Es gibt eine Reihe von Maßnahmen, die zu einer Verbesserung dieser Situation beitragen. Die Firmen sollten möglichst im Vorfeld der Befragung ihren Mitarbeitern den Sinn und Zweck dieser Untersuchungen erläutern und deutlich machen, dass sie dahinter stehen. Das Untersuchungsteam sollte zu Beginn der Befragungen ebenfalls die Gelegenheit haben, mit den Maschinenbedienern zu reden und ihnen den Ablauf der Befragung vorzustellen. Die Befragten sollen über das Ziel der Befragung sowie die langfristigen Vorteile für sie informiert werden. Außerdem gilt es, ihnen die Angst vor einem möglichen Missbrauch der Daten zu nehmen, indem man sie auf die Anonymisierung der Erhebungsdaten hinweist.

Befragungen unterliegen darüber hinaus länderabhängigen Gesetzmäßigkeiten. So müssen sie vielfach mit den jeweiligen Mitarbeitervertretungen bzw. Gewerkschaften abgestimmt werden, was häufig viel Zeit kostet und Einschränkungen in der Befragung zur Folge haben kann.

Am besten geeignet sind die Nutzer, die seit ca. 6-18 Monaten mit einer Maschine umgehen. Sie verwenden in der Regel schon den vollen Funktionsumfang, den sie zur Bewältigung ihrer spezifischen Aufgaben benötigen, können sich andererseits aber noch an Ihre anfänglichen Probleme erinnern. Ebenso gut geeignet sind auch Nutzer, die bereits Erfahrungen mit vergleichbaren Maschinen der Konkurrenz gesammelt haben. Gestaltungsfehler und Abgrenzungsmerkmale können so zusätzlich erfasst werden und trotzdem sind die aufgabenorientierten Aspekte der Analyse vergleichbar.

National oder International? Die Auswahl der Nutzer sollte einen Querschnitt aller Kunden abbilden. Insbesondere, wenn außereuropäische Länder zu den primären Absatzgebieten zählen, sollten auch Befragungen in diesen Ländern durchgeführt werden.

Während sich eine Befragung in den USA und im europäischen Ausland in aller Regel bei vorhandenen Kontakten als unproblematisch erweist, ist eine Befragung in den asiatischen Ländern erfahrungsgemäß mit großem Aufwand verbunden. Um überhaupt an sinnvolle Informationen zu gelangen, müssen aufgrund der deutlichen Kulturunterschiede in jedem einzelnen Land einheimische Experten herangezogen werden, die die Befragung durchführen.

In Anbetracht des Aufwandes wird oftmals auf eine internationale Nutzerbefragung verzichtet. Ratsam ist es allerdings, in diesem Fall zumindest einen Experten im Bereich interkulturellen Designs zu konsultieren. Gerade im asiatischen Raum sind die kulturellen Unterschiede nicht zu unterschätzen und müssen bei der Useware-Gestaltung berücksichtigt werden (s. Kap. 4).

Aus der Praxis: Häufig wird in Useware-Gestaltungsprojekten die Bedeutung der Befragung „echter“ Maschinennutzer unterschätzt. Vielfältige Argumente werden dafür als Begründung zitiert. Zumeist ist man der Auffassung, dass die hauseigenen Servicekräfte ein genaues Bild über die Kunden und Ihre Probleme haben oder dass die Maschinen ja in so unterschiedlichen Anwendungen eingesetzt würden, dass es keine wirklich repräsentativen Nutzergruppen gibt. Die Praxis zeigt aber, dass nur eine unabhängige Untersuchung der Nutzer vor Ort ein reales Bild von der Maschinenanwendung zeichnet.

So erfuhr beispielsweise der Hersteller einer programmierbaren Maschinensteuerung erst durch eine solche Analyse vor Ort davon, dass das Programmiersystem direkt an der Maschine von den Nutzern so gut wie nie benutzt wurde, da diese die Programme lieber an einem normalen PC bearbeiteten. In der Entwicklungsabteilung wurde aber über Jahre viel Zeit in die besonders nutzerfreundliche Gestaltung dieses steuerungsintegrierten Programmiersystems gesteckt, während auf das Offline-Programmiersystem nie Gewicht gelegt wurde. Dieser Zielkonflikt konnte trotz vorangehenden Gesprächen mit Servicekräften erst durch eine ganzheitliche Betrachtung der

Maschinenanwendungen vor Ort identifiziert werden und in der Folge durch eine an den Anforderungen der Nutzer orientierte Entwicklungsstrategie Aufwand einsparen.

Weitere Differenzierungsmerkmale? Neben den genannten Faktoren sind je nach Anwendungsgebiet auch weitere Differenzierungsmerkmale bei der Identifikation von Nutzergruppen zu berücksichtigen. So kann es z. B. von Interesse sein, Unterschiede in der Arbeitsweise bei weiblichen oder männlichen Nutzern zu identifizieren. Ebenso ist evtl. das Alter der Nutzer von Bedeutung und sollte daher in die Nutzerabgrenzung einbezogen werden. Diese und viele weitere Faktoren müssen von Fall zu Fall individuell zu den oben genannten Merkmalen hinzugezogen werden, um so ein repräsentatives Abbild der Nutzerstruktur aufzuzeichnen.

3.2.4 Welche Methoden sind geeignet?

Die unterschiedlichen Betrachtungsgegenstände und Befragungspartner lassen keine Empfehlung bestimmter Datenerhebungstechniken zu. Grundsätzlich gilt, dass für die jeweilige Befragung eine geeignete Methode ausgewählt werden muss. Zur strukturierten Datenerhebung in der Analysephase wird daher meist ein **Methoden-Mix** angewandt. In diesem werden verschiedene Analysemethoden mit dem Ziel kombiniert, den Betrachtungsgegenstand aus möglichst unterschiedlichen Blickwinkeln und mit verschiedenen Detaillierungsgraden zu beleuchten.

Bei der Analyse der Nutzer und ihrer Arbeitsaufgaben ist eine Überschneidung zwischen Arbeits- und Bediensystemgestaltung zu berücksichtigen (Hamburg und Schweppenhäuser 1993). So müssen im Rahmen der Analysephase die übergeordneten, dem **eigentlichen Arbeitsergebnis dienenden Aufgaben** identifiziert werden. Diese sind im Wesentlichen organisatorisch bedingt. Daneben existieren für das technische System Sekundäraufgaben, die sich durch technische Rahmenbedingungen ergeben und damit impliziter Bestandteil der Aufgabengestaltung sind (Streitz 1988).

In Tab. 3.2 sind verschiedene Erhebungsmethoden dargestellt (Eissler 2001). Es wird dabei aufgezeigt, für welches Analyseziel die jeweilige Methode von Bedeutung ist (vgl. Kap. 3.6.2).

Aus der Praxis: In vielen industriellen Useware-Projekten hat sich bei der Analyse eine Kombination aus teilnehmender Beobachtung, Strukturlegen und strukturiertem Interview bewährt (s. Kap. 6.2). Die Maschinennutzer werden dazu zunächst direkt an ihrem Arbeitsplatz besucht. Während der Beobachter die Arbeit des Nutzers kennen lernt, werden auch erste interper-

Tab. 3.2 Gegenüberstellung von Analysemethoden

Methode	Charakteristika	Ziel
<i>Strukturlegetechnik</i>	<ul style="list-style-type: none"> – Erhebung von Strukturen – Erhebung von mentalen Modellen – Kreativer Prozess – Statistische Aussagekraft schwer erreichbar – Zeitaufwändig 	<ul style="list-style-type: none"> – Nutzeraufgaben – Strukturierung der Benutzungsobjekte – Zuständigkeiten – Erhebung des Anwendungsmodells – Überprüfung von Strukturen
<i>Strukturiertes Interview</i>	<ul style="list-style-type: none"> – Gezieltes Fragen und Nachhaken möglich – Statistische Aussagekraft möglich – Vorwissen erforderlich 	<ul style="list-style-type: none"> – Arbeitsabläufe – Nutzeraufgaben – Bedingungen/ Gewichtungen – Zuständigkeiten – Funktionen
<i>Teilnehmende Beobachtung</i>	<ul style="list-style-type: none"> – Erhebung an realen Systemen – Keine statistische Aussagekraft – Oft nur unvollständige Informationen 	<ul style="list-style-type: none"> – Arbeitsabläufe – Zugriffsorte – Nutzeraufgaben – Bedingungen/ Gewichtungen – Verbesserungsvorschläge
<i>Freies Interview</i>	<ul style="list-style-type: none"> – Hoher Detaillierungsgrad – Flexibilität – Keine statist. Aussagekraft – Hohe Kompetenz des Interviewers erforderlich 	<ul style="list-style-type: none"> – Nutzeraufgaben – Funktionen – Bedingungen/ Gewichtungen – Zuständigkeiten – Verbesserungsvorschläge
<i>Fragebogen</i>	<ul style="list-style-type: none"> – Statistische Aussagekraft – Objektive Ergebnisse – Hoher Aufwand – Nur Bekanntes abfragbar (kein kreativer Prozess) 	<ul style="list-style-type: none"> – Nutzeraufgaben – Gewichtungen – Arbeitsplatzmerkmale

sonelle Hemmschwellen abgebaut. Im Anschluss an die etwa halbstündige teilnehmende Beobachtung ziehen sich Beobachter und Nutzer an einen ruhigeren Ort zurück und es werden die beobachteten Aufgaben des Nutzers mit Hilfe der Strukturlegetechnik geordnet und ergänzt. Hierzu werden einfache Papierkärtchen benutzt, auf denen Stichworte des Nutzers notiert werden und die dann zur Visualisierung der Struktur auf dem Tisch hierarchisch angeordnet werden. Bereits hierbei können bestimmte Fragestellungen im Sinne des strukturierten Interviews diskutiert werden. Ist eine vollständige Struktur gelegt, werden noch offene Fragen diskutiert.

Für das strukturierte Interview hat der Befragende einen Fragenkatalog, den er Schritt für Schritt mit dem Nutzer abarbeitet. Je nach Gesprächigkeit des Befragten können nicht immer alle Fragen beantwortet werden. Dabei muss darauf geachtet werden, Fragenkomplexe so auf mehrere Befragungen zu verteilen, dass am Ende der Befragung trotzdem ein vollständiges Bild entsteht.

Das Gespräch sollte dabei möglichst individuell mit jedem Nutzer geführt werden. Es hat sich gezeigt, dass bei mehreren gleichzeitig befragten Personen sich häufig die dominante Persönlichkeit besonders hervortut und die ruhigeren Personen kaum die Gelegenheit haben, sich zu artikulieren. Vor allem sollten die Vorgesetzten der Nutzer den Befragungen nicht beiwohnen. Es hat sich immer wieder gezeigt, dass sich die Nutzer aus Angst vor möglichen Konsequenzen dann kaum noch kritisch äußern.

3.2.5 *Wie sehen die Ergebnisse aus?*

Durch Anwendung der in Tab. 3.2 genannten Analysemethoden werden verschiedene Arten von Informationen und Daten von den Nutzern erhoben (Bödcher 2007). Die Beschreibung der Daten, die sich aus der Analyse ergeben können, werden je nach Formalisierungsgrad in informell, semiformal und formal unterschieden:

- **Unstrukturierte, textuelle Daten** sind informelle Daten, die qualitative Aussagen aus Befragungen wiedergeben. Diese Daten werden lediglich zu Vergleichszwecken zusammengefasst und können zu jedem Zeitpunkt der Entwicklung Verwendung finden.
- **Systemisierte, textuelle Daten** sind ebenfalls informell, spezifizieren unstrukturierte Daten jedoch aufgrund von Strukturelementen (Absätze etc.) und Hervorhebungen (Fettdruck etc.).
- **Strukturierte, textuelle und numerische Daten** sind semiformalen Daten, die eine objektive Bewertung aufgrund von Vergleichen oder mathematischen Verfahren zulassen.
- **Strukturelle Daten** sind formale Modelle (bspw. Aufgabenmodelle), die aufgrund einer definierten Syntax eine elektronische Weiterverarbeitung zulassen.

3.2.5.1 **Mentales Modell der Nutzer**

Genau wie eine Sprache besteht auch eine Bedienstruktur aus Regeln und Gesetzmäßigkeiten, die von allen Nutzern eines Sprachraumes verstanden werden. Auch das Bediensystem muss daher die Sprache des Nutzers sprechen, um von diesem richtig verstanden zu werden.

Das Verständnis, welches auf Regeln und Gesetzmäßigkeiten basiert, wird als mentales Modell bezeichnet und beschreibt somit die Denkweise und Auffassung der Nutzergruppe (s. Kap. 2.2.2). Ist das Bediensystem am mentalen Modell der Nutzer orientiert, wird es von diesen sofort verstanden und sie werden „intuitiv“ interagieren.

Aus der Praxis: Der typische Aufwand für eine Analysephase ist natürlich abhängig von der Anzahl der Befragungsteilnehmer wie auch von der Komplexität der Anwendung. Erfahrungsgemäß ist für die gesamte Analysephase ein Zeitraum von 1–3 Monaten vorzusehen. Dabei entfallen 2–5 Tage auf die Einweisung des (evtl. externen) Befragungspersonals in die Technologie und die Abstimmung der Analyseinhalte. 1–2 Wochen sind für die Vorbereitung der Analyseinstrumente (Fragebogen, Beobachtungstechnik, ...) sowie Probebefragungen einzurechnen. Parallel hierzu sollten bereits Termine bei den repräsentativen Nutzern vor Ort organisiert werden, da hier häufig lange Vorlaufzeiten zu berücksichtigen sind.

Aufgrund des Betriebsverfassungsgesetzes sind Befragungen in Deutschland meistens nur nach Zustimmung der Mitarbeitervertretung zulässig, was nicht selten mehrere Wochen erfordert. Für die eigentlichen Befragungen sind je nach Anzahl mehrere Wochen vorzusehen. Erfahrungsgemäß lassen sich 3–5 Kundetermine pro Woche realisieren, stark abhängig von den Reisezeiten zwischen den Terminen. Da die Befragungsergebnisse idealerweise direkt im Anschluss an die Befragung dokumentiert werden, sollte der Terminplan nicht zu eng gesteckt werden. Für die Gesamtdokumentation der Analysephase sollten weitere 1–2 Wochen vorgesehen werden.

Sehr hilfreich bei Befragungen sind Gesprächsaufzeichnungen mit einem Diktiergerät oder einem Camcorder. Auch Fotos von existierenden Installationen beim Kunden helfen später bei der Erstellung des Untersuchungsberichts. Viele Firmen gestatten solche Aufnahmen wenn überhaupt, nur nach einer aufwändigen Genehmigungsprozedur, was wiederum die Zeitspanne verlängern kann.

3.2.5.2 Strukturierte Informationen

Ziel der Analysephase ist die Bereitstellung einer strukturierten Informationsbasis für den nachfolgenden Gestaltungsprozess beginnend mit der Strukturgestaltung. Zwischen der Auswertung der Information am Ende der Analysephase und der Strukturgestaltung ist ein fließender Übergang. So wird durch Abstraktion der identifizierten Einzelstrukturen eine Referenzstruktur der Bedienung abgeleitet. Des Weiteren können drei unterschiedliche Datenarten unterschieden werden:

- Technologieorientierte Daten und Organisation:
Hierzu zählen z. B. Funktionslisten und Nutzungskontexte.
- Nutzungsorientierte Daten:
Hierzu zählen z. B. Aufgabenklassifizierung und Informationsbedarf nach Nutzergruppen.
- Handlungsorientierte Daten:
Hierzu zählen z. B. Bedienabläufe, Handlungsmodule, Workflows.

Das Ergebnis der Analysephase ist somit eine Datenbasis, in der Einzelergebnisse präzise und vollständig beschrieben sind, um von allen Personen oder Prozessen der Entwicklung zu unterschiedlichen Zwecken genutzt werden zu können. Die Daten dienen dabei als Grundlage für weitere Entscheidungen. Dabei ist eine Unterscheidung zwischen Zugriff auf Einzelergebnisse, dem Zugriff auf zusammengefasste Einzelergebnisse durch Personen und dem Zugriff auf zusammengefasste Einzelergebnisse durch Prozesse möglich.

Ein Zugriff auf Einzelergebnisse ist notwendig, wenn Abhängigkeiten zwischen zwei oder mehreren Erhebungsgegenständen untersucht werden (z. B. ob die befragte Person innerhalb der Erhebung generell eine ablehnende Haltung im Hinblick auf Neuerungen des technischen Systems angenommen hat). Diese Korrelationen werden manuell untersucht, da im Vorfeld keine Aussage über mögliche Abhängigkeiten zwischen Daten getroffen werden kann. Hierbei zeigt sich z. B. die Wichtigkeit rein qualitativer Ergebnisse, bei denen Einzelpersonen aufgrund bestimmter, zusammenhängender Eindrücke Entscheidungen treffen können.

Zusammengefasste Ergebnisse beschreiben die (repräsentative) Meinung aller Befragten zu einem einzigen Erhebungsgegenstand (z. B. Frage nach der Notwendigkeit eines größeren Displays). Diese Ergebnisse dienen beim Zugriff durch Personen als Argumentations- oder Diskussionsgrundlage für weitere Entscheidungen. In der Praxis werden diese Ergebnisse den Projektbeteiligten meist in Präsentationen, Workshops etc. vorgestellt. Hierbei fallen sowohl quantitative als auch qualitative Daten (häufig quantifiziert) an.

Neben Personen können auch (mehr oder weniger) automatisierte Prozesse auf zusammengefasste Ergebnisse zugreifen. Dabei muss sichergestellt werden, dass diese Prozesse genau spezifiziert sind und die benötigten Daten in weiterverarbeitbarer Form (Strukturelle Daten) zur Verfügung gestellt bekommen (z. B. können Aussagen über die Displaygröße in der Gestaltung als Hinweis oder sogar Vorgabe für den Designer verwendet werden). Diese Daten sind im Rahmen modellbasierter Vorgehensweisen Instanzen der generischen Modelle (z. B. Aufgabenmodelle einzelner Nutzer/Nutzergruppen).

Zur Unterstützung der Interviewer in der Phase der Analyse können Software-Werkzeuge wie bspw. useDATA (Bödcher 2007) oder TAMaRA (s. auch Kap. 5.4.3) zur Erhebung und Auswertung der Daten eingesetzt werden.

Zusammengefasst ist das Ergebnis der Analysephase die Bereitstellung strukturierter Informationen für die Gestaltungsphase.

3.3 Strukturgestaltung

Bei der Nutzung technischer Systeme stellt die mangelnde Aufgabenorientierung weit vor der Anordnung und Gestaltung von Dialogobjekten das größte Problem für die Nutzer dar. Die gängigen Entwicklungswerkzeuge unterstützen die Entwickler

Abb. 3.5 Inhalt der Strukturgestaltungsphase



bei der Programmierung von Prozess- und Anlagenfunktionalität und stellen ihnen sogar leistungsfähige Editoren zum „Malen“ von Bedienoberflächen zur Verfügung. Was fehlt, sind Hilfsmittel und Methoden zur nutzer- bzw. aufgabengerechten Strukturierung der zu entwickelnden Useware.

Im Rahmen der durchgeführten Analysephase wurden vor Ort Nutzer befragt, die im Umgang mit dem technischen Gerät stehen. Hierbei werden Aufgabenstrukturen identifiziert, die beschreiben, in welchen Kontexten der Nutzer wie mit der Maschine umgeht. Neben dem Nutzer wird im Rahmen der Analyse auch das technische System betrachtet und eine Funktionsstruktur angelegt, aus der die Einzelfunktionen und deren Zusammenhänge hervorgehen.

Ziel der Strukturgestaltungsphase (s. Abb. 3.5) ist die aufgabenorientierte Strukturierung der Bedienung des technischen Geräts. Hierzu werden die erhobenen Aufgabenstrukturen harmonisiert und eine grundlegende Struktur für die Interaktion mit dem Gerät erstellt. Diese wird dann um elementare Aufgaben ergänzt und mit den technischen Funktionen verknüpft. Da dieses sog. Benutzungsmodell nicht ganz losgelöst von der Bediensystem-Hardware erstellt werden kann, muss parallel zur Strukturgestaltung auch eine erste Betrachtung möglicher Hardware-Plattformen durchgeführt werden.

An diesem Prozessschritt sollten neben den Entwicklern vor allem Mitarbeiter aus der technischen Dokumentation beteiligt werden, da diese über ein hohes Maß an Erfahrung im Bereich der nutzergerechten Informationsstrukturierung verfügen. Außerdem sollten bereits in dieser Phase erste Nutzertests durchgeführt werden. Hierzu können z. B. Servicemitarbeiter herangezogen werden.

3.3.1 Beschreibung einer grundlegenden Benutzungsstruktur

Im Rahmen der Strukturgestaltung wird eine aufgabenorientierte, plattformunabhängige Sichtweise auf das Mensch-Maschine-System modelliert. Zur Hervorhebung der zentralen Position des Nutzers und zur Unterstreichung des Werkzeugcharakters der Maschine wird dieses Modell als **Benutzungsmodell** bezeichnet. Dementsprechend werden die im Benutzungsmodell beschriebenen plattformunabhängigen Aufgaben, Tätigkeiten und Handlungen abstrahiert als **Benutzungsobjekte** bezeichnet. Das Benutzungsmodell bildet somit die Schnittstelle zwischen Nutzeraufgaben und Maschinenfunktionalität und ist Grundlage für die konkrete Aufbereitung des Bediensystems (s. Abb. 3.6).

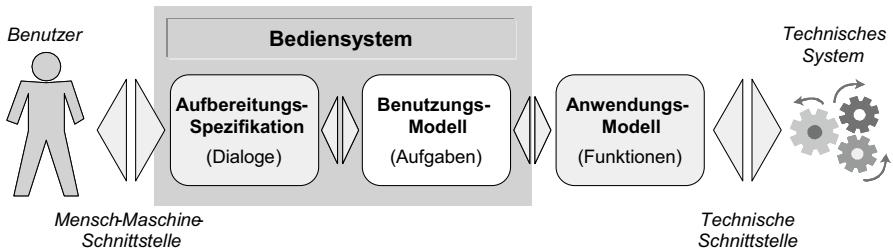


Abb. 3.6 Einordnung des Benutzungsmodells in das Mensch-Maschine-System

Die Grundlage für das Benutzungsmodell sind Informationen, die direkt bei den Anwendern erhoben werden. Primäres Augenmerk liegt dabei auf den Ergebnissen der Aufgabenanalyse (Rauterberg 1994). Dabei sind insbesondere folgende Aufgabenmerkmale von Bedeutung:

- Statische Aufgabenstruktur
- Dynamische Handlungsabläufe
- Benötigte Funktionen und Informationen
- Ausführungsregeln, Vor- und Nachbedingungen
- Kontextparameter wie bspw. Häufigkeit der ausführbaren Aufgaben

Das Benutzungsmodell wird mit Hilfe einfach verständlicher Konstrukte erstellt. Die in UML notierte Abb. 3.7 zeigt, dass das Benutzungsmodell auf einer Hierarchie von **Benutzungsobjekten** basiert. Diese beschreiben den Nutzungskontext und die plattformunabhängigen Aufgaben, Tätigkeiten und Handlungen. Sie enthalten zudem Informationen über Zuständigkeiten, Rechte, Bedingungen und Beziehungen. Den jeweiligen, durch Benutzungsobjekte beschriebenen Aufgaben werden **elementare Benutzungsobjekte** zugeordnet. Diese lassen sich in die fünf elementaren Benutzungsobjekte

- Ändern
- Auslösen
- Auswählen
- Eingeben
- Informieren

untergliedern. Die fünf elementaren Benutzungsobjekte resultieren aus einer abstrakten Betrachtung der Interaktion des Nutzers mit einem Interaktionssystem. Die Einteilung in fünf elementare Benutzungsobjekte hat den Vorteil, alle Projektbeteiligten zu einem konsequenten Überdenken der bisherigen Interaktionsobjekte zu zwingen und somit die Konsistenz zu steigern. In vielen – auch komplexen – Maschinenbediensystemen haben sich diese Elemente als ausreichend erwiesen.

Mit Hilfe der elementaren Benutzungsobjekte lassen sich alle Interaktionsaufgaben in den hier betrachteten Mensch-Maschine-Systemen vollständig und mit einheitlicher Nomenklatur beschreiben. Aufgrund der Plattformunabhängigkeit der Benutzungsobjekte ist damit eine von konkreten Interaktionsobjekten (z. B. Buttons, Sliders o. Ä.) unabhängige Modellierung des Maschinenbediensystems möglich.

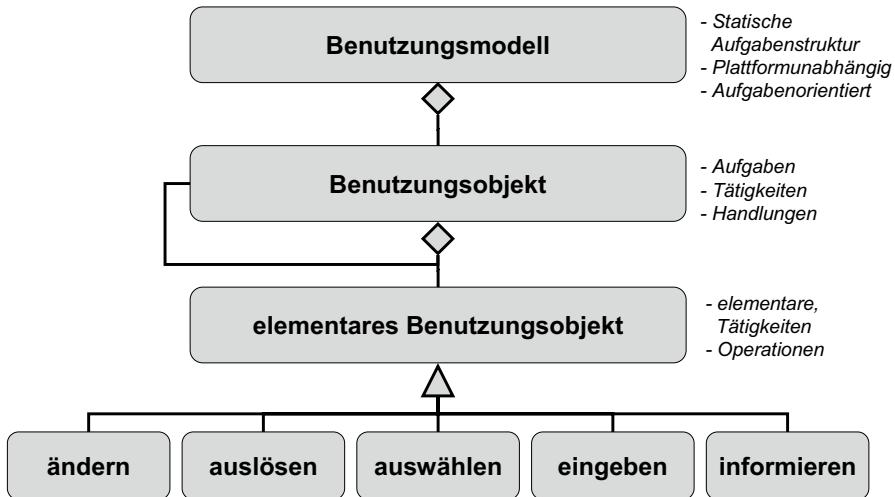


Abb. 3.7 Schematischer Aufbau des Benutzungsmodells

Zur Beschreibung einer übergeordneten Benutzungsstruktur werden also die identifizierten Nutzeraufgaben verwendet. Diese können durch Beziehungen im Sinne einer Hierarchie miteinander verknüpft werden. Außerdem können sie mit zusätzlichen Informationen (→ Attribute) angereichert werden, die sie näher spezifizieren (z. B. hinsichtlich der Nutzergruppe oder des Zugriffsortes). Basis für diese Struktur bilden die in der Analysephase identifizierten Aufgabenstrukturen der befragten Nutzer, die im Benutzungsmodell harmonisiert werden (s. Abb. 3.8). Hierzu werden globale Handlungskontexte identifiziert, diese dann in Unteraufgaben zerlegt und in Form der sog. Benutzungsobjekte (BO) im Benutzungsmodell notiert. Je nach Menge an befragten Nutzern können hierzu auch statistische Methoden herangezogen werden.

Die Aufgaben werden im Benutzungsmodell vom übergeordneten Handlungskontext bis in Teilaufgaben unterteilt. Diese Einteilung lässt es zu, die Struktur hinsichtlich ihres grundsätzlichen Aufbaus zu untersuchen und übergeordnete Bedienkontakte wie beispielsweise „Maschine betreiben“, „Probleme diagnostizieren“ und „Einstellungen vornehmen“ zu identifizieren. Ebenso lässt sich über die Auszeichnung von Aufgaben und Teilaufgaben bei der späteren plattformspezifischen Aufbereitung der Useware die Entscheidung treffen, in welcher Art Inhalte in den Ausgabemedien zusammengefasst werden.

In ihrer Funktion als Strukturierungselemente dienen Benutzungsobjekte bei der Abbildung auf ein konkretes Interaktionskonzept weniger der Information oder Interaktion als vielmehr der Navigation und Orientierung. Daher enthalten Benutzungsobjekte neben ihrer Bezeichnung vor allem Informationen über berechtigte Nutzergruppen und mögliche Interaktionsorte.

Den Benutzungsobjekten werden elementare Benutzungsobjekte zugeordnet. Diese repräsentieren Teilaufgaben, die zur Durchführung der Arbeitsaufgabe erforderlich sind und sich nicht in weitere Teilaufgaben zur Erfüllung des Arbeitszwecks

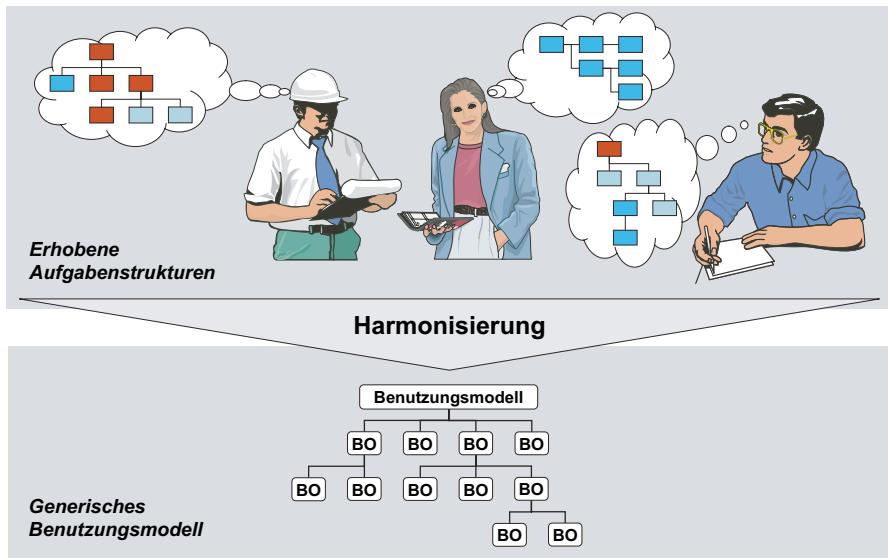


Abb. 3.8 Harmonisierung der Aufgabenstrukturen

zerlegen lassen (Atomarität). Die elementaren Benutzungsobjekte beschreiben somit die eigentlichen Interaktionshandlungen des Nutzers an der Maschine, die bei der Implementierung durch Interaktionsobjekte visualisiert und mit Maschinenfunktionen verknüpft werden.

Grundsätzlich lassen sich die elementaren Benutzungsobjekte in die zwei Grundtypen unidirektional und bidirektional einteilen. Informieren beschreibt dabei die elementare Aufgabe des Nutzers, Kenntnis über einen bestimmten Sachverhalt oder ein bestimmtes Datum zu erlangen. Dieses elementare Benutzungsobjekt ist unidirektional, d. h. das System stellt entsprechende Daten zur Verfügung, und es wird keine Aktion vom Nutzer erwartet. Demgegenüber stehen die elementaren Benutzungsobjekte, die den bidirektionalen Informationsaustausch zwischen Nutzer und Maschine beschreiben. Aufgrund der Vielfältigkeit werden vier strukturell oder inhaltlich unterschiedliche elementare Benutzungsobjekte „ändern“, „auslösen“, „auswählen“ und „eingeben“ unterschieden.

- **Informieren**

Informieren beschreibt die elementare Aufgabe des Nutzers, Kenntnis über einen bestimmten Sachverhalt zu erlangen. Diese Form der Interaktion ist unidirektional, d. h. das System stellt entsprechende Daten zur Verfügung und es wird keine Aktion vom Nutzer erwartet.

- **Auslösen**

Das Auslösen repräsentiert die direkte Betätigung einer Maschinenfunktionalität. Dabei kann das Verhalten der verknüpften Funktion die eines Triggers (z. B. „Zähler zurücksetzen“) oder die eines Tasters (z. B. „in X-Richtung verfahren“) sein.

- **Auswählen**

Beim Auswählen kann der Nutzer auf eine bestimmte Menge an vorgegebenen Auswahlmöglichkeiten zurückgreifen. Dieses elementare Aufgabenobjekt ist hinsichtlich mehrerer Aspekte zu unterscheiden.

Das Auswählen kann eine direkte Funktionsauslösung zur Folge haben (z. B. Modus-Auswahl: „Ein“, „Aus“, „Automatik“) oder nur das Setzen eines Parameters in der Maschinensteuerung oder dem Bediensystem (z. B. Maßeinheit: „m“, „cm“, „mm“). Ebenso muss unterschieden werden, ob eine oder mehrere Möglichkeiten ausgewählt werden können. Zudem kann die Menge der vorgegebenen, auswählbaren Möglichkeiten fest oder erweiterbar sein.

- **Ändern**

Das Ändern ermöglicht dem Nutzer die Manipulation von Daten oder Parametern in der Steuerung oder dem Bediensystem (z. B. Geschwindigkeit: 15 km/h). Eine charakteristische Eigenschaft des Änderns ist, dass die Manipulation relativ zu einem bereits vorhandenen Wert erfolgt.

- **Eingeben**

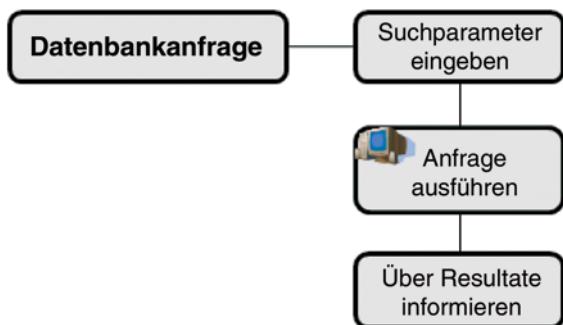
Das Eingeben ist grundsätzlich dem Ändern ähnlich. Allerdings mit dem prinzipiellen Unterschied, dass der Nutzer hierbei absolut vorgeht. D. h. dass er beliebige Texte oder Werte zu einem bestimmten Kontext mitteilen kann (z. B. „Passwort eingeben“, „Zielkoordinaten angeben“). Auch hier ist die Angabe eines Vorgabewerts möglich, dieser wird aber bei einer Eingabe überschrieben.

Die elementaren Benutzungsobjekte können beim Prototyping oder bei der späteren Realisierung des Bediensystems auf die plattformspezifischen Interaktionsobjekte abgebildet werden. So könnte die elementare Aufgabe „Betriebssystem auswählen“ durch einen Dialog mit Radiobuttons oder durch eine Drop-Down-Liste realisiert werden. Durch die abstrakte Beschreibung im Benutzungsmodell ist jedoch sichergestellt, dass gleichartige Elementaraufgaben immer mit dem gleichen Interaktionsobjekt umgesetzt werden. Außerdem ist zu bedenken, dass beim Einsatz unterschiedlichster Hardware-Plattformen auch unterschiedlichste Interaktionsobjekte verwendet werden können bzw. müssen. So müsste ein Auswahldialog bei einem sprachgesteuerten System z. B. durch eine Sprachausgabe der auszuwählenden Vorgaben an die verfügbaren Möglichkeiten angepasst werden.

Trotz der konsequent an den Aufgaben der Nutzer orientierten Vorgehensweise lässt sich ab einer bestimmten Konkretisierungsstufe die aufgabenorientierte Sichtweise nicht mehr von der funktionalen Sichtweise trennen, da hier technische Vorgaben und Einschränkungen der Maschine die Bedienphilosophie beeinflussen oder auch bestimmen. Intuitivität ist eben nur bis zu einem bestimmten Grad umsetzbar.

Beim Übergang von der rein aufgabenorientierten zur vermischt aufgaben- und funktionsorientierten Modellierung werden daher die elementaren Benutzungsobjekte zur Strukturierung der Inhalte verwendet. Sie definieren die Art der Funktionsauslösung zumindest ohne ihr dabei ein spezifisches Aussehen zuzuweisen und gewährleisten, dass die Art der Interaktion auf allen Plattformen/Produktfamilien gleich ist.

Abb. 3.9 Beispielhafte Verwendung von Systemaufgaben



3.3.2 Erweiterung des Benutzungsmodells

Das Benutzungsmodell konnte sich in einer Vielzahl von Industrie- und Forschungsprojekten als ein wichtiges Hilfsmittel zur Modellierung einer Benutzungsschnittstelle beweisen. Über mehrere Jahre Nutzungserfahrung stellte sich allerdings heraus, dass das Benutzungsmodell Nachbesserungsbedarf im Bereich der dynamischen Modellierung einer Benutzungsschnittstelle aufweist (Meixner 2010). Auf Basis der Defizite wurde das Benutzungsmodell daher um wichtige Modellierungsmöglichkeiten erweitert.

Systemaufgaben: Bei der Definition eines Benutzungsobjekts muss der Entwickler zwischen interaktiven Benutzaufgaben und Aufgaben des Systems unterscheiden können, was durch das aktuelle Benutzungsmodell nicht unterstützt wird. Eine feingranularere Aufteilung der Systemaufgaben, wie dies bei elementaren Benutzungsobjekten möglich ist, ist nicht durchführbar, da ein komplexes Computersystem eine schier endlose Anzahl an verschiedenen Aufgaben wie z. B. Datenbankabfragen, Berechnungen, Emails-versenden durchführen kann. Die Systemaufgaben fungieren somit als Verbindungsglied zwischen der Modellierung der Aufgaben und der Modellierung der Applikationslogik. Abbildung 3.9 visualisiert die Nutzung von Systemaufgaben („Anfrage ausführen“) zur Modellierung von Datenbankabfragen. In diesem Beispiel gibt der Nutzer Suchparameter für eine Datenbankabfrage ein, die dann durch das Computersystem automatisiert ausgeführt wird. Wurde die Anfrage erfolgreich durch das Computersystem beendet, so wird der Nutzer über das Resultat informiert.

Optionalität: Um die Optionalität der Aufgaben spezifizieren zu können, wird die Semantik der Benutzungsobjekte und aller elementaren Benutzungsobjekte erweitert. Abbildung 3.10 zeigt die Verwendung von erforderlich (durchgängige Umrandung) und optional (gestrichelte Umrandung) deklarierten Aufgaben. Wird eine Datei geladen, so kann es optional sein, dass die Datei mit einem Passwortschutz versehen ist. Ist dies der Fall, muss der Nutzer das korrekte Passwort eingeben, bevor die Datei geöffnet werden kann. Benutzungsobjekte und elementare Benutzungsobjekte werden formal nach ihrer Optionalität spezifiziert, d. h. sie können als optional oder erforderlich deklariert werden.

Abb. 3.10 Beispielhafte Verwendung von optionalen Aufgaben

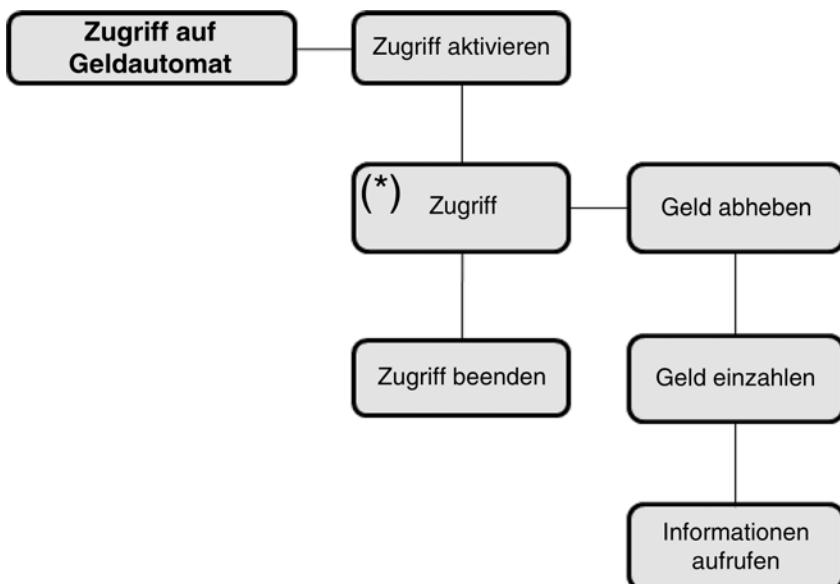
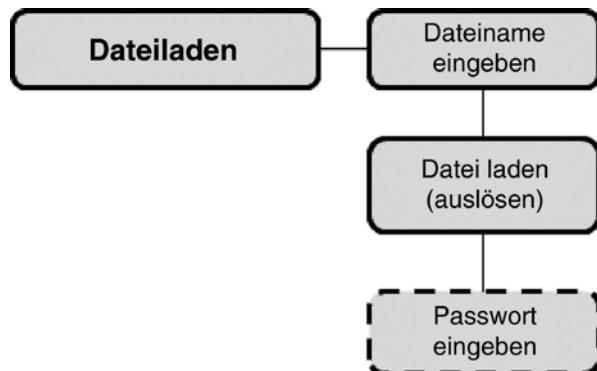
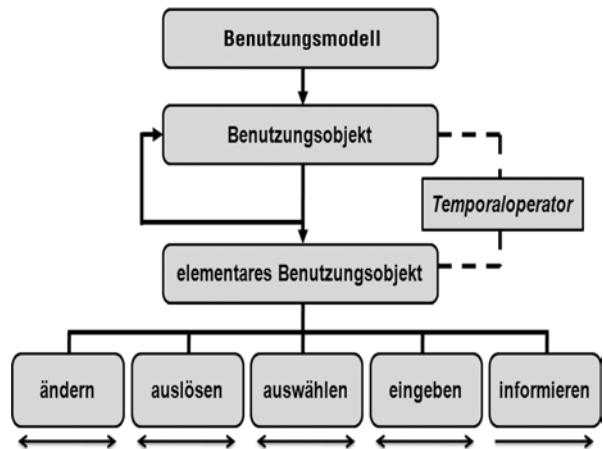


Abb. 3.11 Beispielhafte Verwendung von Mehrfachausführungen

Mehrfachausführung: Eine Mehrfachausführung von Aufgaben wird auf jeder Ebene der Aufgabenspezifikation durchgeführt. Demnach können Benutzungsobjekte, wie auch alle elementaren Benutzungsobjekte mehrfach ausgeführt werden. Hierbei muss beachtet werden, dass eine Aufgabe beliebig (die Anzahl der Ausführungen ist bekannt) oder unbegrenzt (die Anzahl der Ausführungen ist unbekannt, jedoch mindestens einmal) oft ausgeführt werden kann.

Abbildung 3.11 zeigt die beispielhafte Beschreibung beim Zugriff auf einen Geldautomaten. Der Zugriff wird durch drei Aufgaben spezifiziert: „Zugriff aktivieren“, „Zugriff“ und „Zugriff beenden“. Die Aufgabe „Zugriff aktivieren“ wurde zum Zwecke der einfacheren Darstellung nicht weiter ausgeführt. Die Aufgabe

Abb. 3.12 Schematischer Aufbau des erweiterten Benutzungsmodells



„Zugriff“ wird durch einen Stern (*) im linken oberen Eck ergänzt. Dies stellt in diesem Beispiel die Mehrfachausführung dar, d. h. die Aufgabe „Zugriff“ kann unbegrenzt oft ausgeführt werden. Beispielsweise könnte ein Nutzer erst „Geld abheben“ um sich dann im Anschluss Informationen wie bspw. den Kontostand anzeigen zu lassen („Informationen aufrufen“).

Temporaloperatoren: Die umfangreichste Neuerung des erweiterten Benutzungsmodells ist die Einführung von Temporaloperatoren (vgl. Abb. 3.12). Diese ermöglichen es, Aufgaben einer hierarchischen Ebene in einen zeitlichen Zusammenhang zu stellen. Für den modellbasierten Useware-Engineering Prozess wurden die (binären) Temporaloperatoren *Reihenfolgeunabhängigkeit*, *Auswahl*, *Nebenläufigkeit*, *Deaktivierung* und *Sequenz* als Erweiterung für useML gewählt.

- **Reihenfolgeunabhängigkeit** (engl. Order Independence, Abkürzung: IND): Die Aufgaben können in beliebiger Reihenfolge starten. Wurde eine Aufgabe gestartet, so muss die andere Aufgabe allerdings warten, bis die vorherige beendet ist. Beispiel: In einem Formular müssen Vor- und Nachname in jeweils einem Textfeld eingetragen werden, die Reihenfolge der Eintragung ist irrelevant.
- **Auswahl** (engl. Choice, Abkürzung: CHO): Genau eine der Aufgaben (BO oder eBO) wird ausgeführt. Wurde sich für die Ausführung einer der beiden Aufgaben entschieden, steht die andere Aufgabe nicht mehr zur Ausführung zur Verfügung. Beispiel: Beim Start eines Textverarbeitungsprogrammes kann der Anwender entscheiden, ob ein neues Dokument erzeugt oder ein bestehendes geöffnet wird.
- **Nebenläufigkeit** (engl. Concurrency, Abkürzung: CON): Aufgaben können in beliebiger Reihenfolge oder zur gleichen Zeit ausgeführt werden. Dabei ist es auch möglich, dass eine Aufgabe startet, während eine andere noch nicht beendet wurde. Beispiel: Der Benutzer liest Informationen vom Monitor und spricht gleichzeitig in ein Mikrofon.

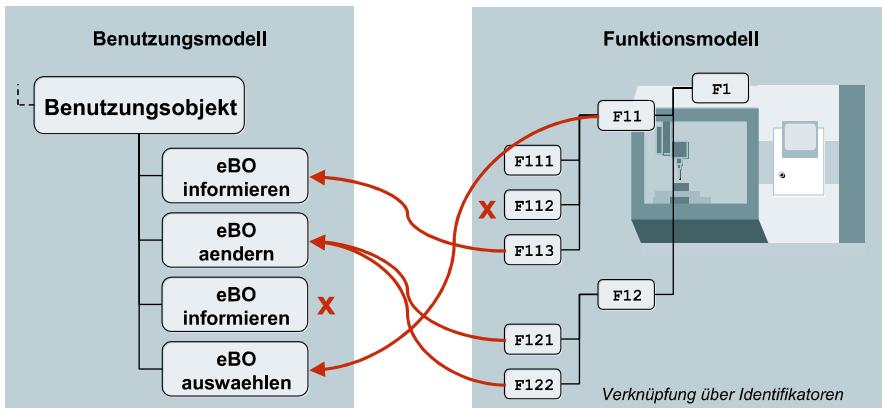


Abb. 3.13 Verknüpfung des Benutzungsmodells mit dem Funktionsmodell

- Deaktivierung (engl. Deactivation, Abkürzung: DEA): Die erste Aufgabe wird unterbrochen, wenn die zweite Aufgabe gestartet wird. Beispiel: Das Abspielen einer CD wird durch Drücken eines Beenden-Knopfes abgebrochen.
- Sequenz (engl. Sequence, Abkürzung: SEQ): Die Aufgaben werden in der angegebenen Reihenfolge nacheinander ausgeführt. Die zweite Aufgabe kann erst starten, wenn die erste beendet wurde. Beispiel: Der Benutzer einer Datenbank muss sich zuerst an der Datenbank anmelden, bevor er auf die Daten zugreifen kann.

3.3.3 Verknüpfung von Benutzungs- und Funktionsmodell

Im generischen Benutzungsmodell wird auch die Verbindung zum Funktionsmodell der Maschine erstmalig hergestellt. Das Funktionsmodell beinhaltet – analog zur baulich orientierten Bauteilstruktur – die strukturierte Darstellung aller Softwarefunktionen der Maschine. Deren Identifikatoren werden einfach als Attribut der Benutzungsobjekte notiert, um eine Verknüpfung zwischen Aufgabe und Maschinenfunktionalität herzustellen. Dieser Vorgang ist unter Einbeziehung aller Projektbeteiligten durchzuführen, um vor allem bei den technischen Entwicklern ein gemeinsames Verständnis vom Unterschied zwischen Nutzeraufgaben und Maschinenfunktionalität zu erzeugen.

Diese Verknüpfung hat weniger zum Ziel, automatisch lauffähige Software aus den beiden Modellen zu generieren, sondern vielmehr soll sichergestellt werden, dass alle beschriebenen Informations- und Interaktionsbedürfnisse der Nutzer auch durch entsprechende Maschinenfunktionen umgesetzt werden können. Außerdem können bei dieser Zuordnung auch die Funktionen identifiziert werden, die der Nutzer nicht oder nur selten braucht (s. „F112“ in Abb. 3.13) und die daher mit niedriger Priorität behandelt werden können. Auf diese Weise ist die Strukturierung des

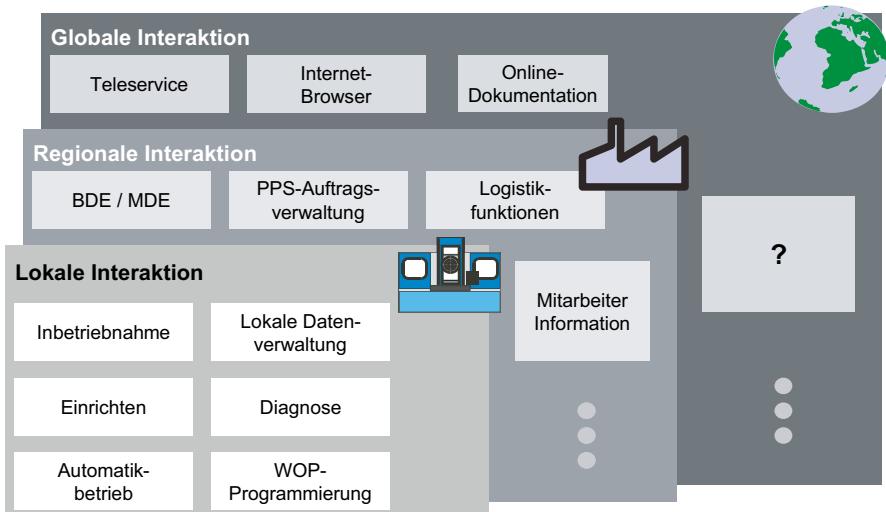


Abb. 3.14 Globale, regionale und lokale Bedienung

Benutzungsmodells gleichzeitig ein Hilfsmittel zur Planung und Abschätzung von Entwicklungsaufwänden in frühen Projektphasen.

3.3.4 Berücksichtigung weiterer Attribute

Die Möglichkeiten, die mit dem Einzug der PC-Technologie in das industrielle Umfeld gegeben sind, lassen die Realisierung vernetzter Systeme im Produktionsbereich zu. Der Werker hat damit zunehmend organisatorische und dispositive Aufgaben und muss seine Aufgaben an unterschiedlichen Orten erledigen. Anwendungen wie Email, Internet, Terminplaner, Bestellwesen, Online-Dokumentation, Teleservice, usw. werden zukünftig auch bei der Maschinenbedienung relevant werden.

Beim **Zugriffsort** handelt es sich daher nicht um den physikalischen Aufstellort der Maschine, sondern um den Ort, von dem aus die Maschine bedient wird. Es wird zwischen der **lokalen** (die Maschine betreffende), der **regionalen** (das Maschinenumfeld/und die Produktion betreffende) und der **globalen** (alle Bereiche außerhalb des Maschinenumfeldes betreffende) Bedienung unterschieden (s. Abb. 3.14). Drei grundverschiedene Bedienbereiche bei der Maschinenbedienung erfordern natürlich auch Berücksichtigung bei der Strukturgestaltung. Aufgaben müssen in einer Grundstruktur dahingehend gekennzeichnet werden, welche Bedienbereiche sie betreffen. Lokale, sicherheitskritische Aufgaben dürfen beispielsweise nur direkt an der Maschine und niemals aus der Ferne ausgeführt werden. Derartig grundlegenden Eigenschaften können und sollten bereits bei der Strukturgestaltung festgehalten werden.

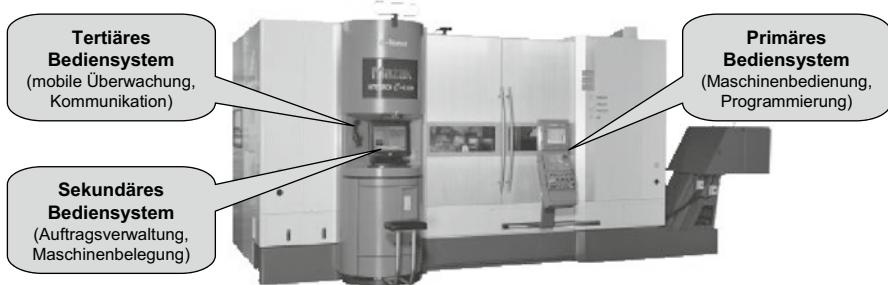


Abb. 3.15 Primäres, Sekundäres und Tertiäres Bediensystem an einer Werkzeugmaschine.
(Quelle: MAZAK)

Aus dieser Betrachtung folgen drei charakteristische Typen von Bediensystemen (s. Abb. 3.15):

1. Die lokalen Bedienfunktionen umfassen die eigentlichen Maschinen- und Steuerungsfunktionen und sind dabei auf dem **primären Bediensystem** direkt an der Maschine zu finden. Die Entwicklung der Funktionalitäten obliegt hier dem Maschinenhersteller.
2. Die regionalen Organisations- und Kommunikationsfunktionen sollten auf ein **sekundäres Bediensystem** ausgelagert werden, welches sich direkt neben der Maschine befindet. Hierbei handelt es sich in der Regel um normale WINDOWS-Systeme, welche um einen Bereich zur Statusanzeige und Manipulation erweitert wurden. Durch die Verwendung von Standard-WINDOWS-Software wird der Entwicklungsaufwand für den Maschinenhersteller an dieser Stelle drastisch verringert. Zusätzlich kann dieses „Leitstandsprinzip“ auch als Investitionssicherung für ältere Maschinen gesehen werden, denn diese können mit verhältnismäßig geringem Aufwand nachgerüstet werden.
3. Das **tertiäre Bediensystem** ist dem weltweiten also globalen Informationsabruft via Mobiltelefon, Notebook oder PC zugeordnet. Schwerpunkt der globalen Interaktion ist allerdings die reine Informationsvermittlung, das Auslösen von Funktionen sollte hierüber aus Sicherheitsgründen nicht möglich sein.

Den Anforderungen verteilter Bediensysteme muss im Benutzungsmodell Rechnung getragen werden. Es muss neben der Gesamtheit der möglichen Aufgaben, die bei der Benutzung der Maschine anfallen, festgehalten werden, wer die Aufgabe an welchem Zugriffsort durchführen darf. Hierzu werden die Benutzungsobjekte im Benutzungsmodell mit Hilfe von Attributen gekennzeichnet.

Gleiches gilt für die **Nutzergruppe**. Werden in der Analyse verschiedene Nutzergruppen mit unterschiedlichen Aufgaben oder Kompetenzen ermittelt, können diese bei der Strukturgestaltung direkt den Aufgaben zugeordnet werden. So kann schon in dieser Phase festgelegt werden, welche Nutzer welche Aufgaben übernehmen dürfen. Aufgaben, zu deren Ausführung die Nutzer nicht autorisiert sind, können für diese ausgeblendet werden. Somit werden wenig qualifizierte Nutzer nicht überfordert, aber hoch qualifizierte Nutzer gleichzeitig nicht unterfordert. Die nut-

zergruppenspezifischen Benutzungsstrukturen entsprechen dem Grundsatz: „Soviel wie nötig, so wenig wie möglich!“

Des Weiteren können Benutzungsobjekte und elementare Benutzungsobjekte durch **Personas** unterschieden werden. Eine Persona ist eine fiktive Person, die Wünsche, Anforderungen und Ziele an bzw. mit einem bestimmten interaktiven System hat. Diese Beschreibung der Persona beruht auf realen Untersuchungen, die im Vorfeld der Persona Entwicklung durchgeführt werden. Man unterscheidet (nach der Wichtigkeit der Ziele) primary, secondary aber auch supplementary persona. Die Persona sind sehr konkret formuliert z. B. Name, Geschlecht, berufliche Ausbildung, Wünsche, Interessen, Bild der Person, die durch die Persona repräsentiert wurde etc. (Thiels 2008).

Mittels unterschiedlicher **Gerätetypen** können Gerät- und Produktfamilien unterschieden werden, die auf Basis eines gemeinsamen Benutzungsmodells modelliert werden. Das Attribut Gerätetyp gibt dabei an, welche Maschine, welches Gerät oder welche Variante entsprechende Benutzungsobjekte und elementaren Benutzungsobjekte umsetzen kann.

Aus der Praxis: Die Bedeutung, die die so genannten verteilten Bediensysteme auch im industriellen Umfeld bekommen werden, wird an folgendem Beispiel deutlich:

Gerade einfachere Geräte aus der Automatisierungstechnik verfügen häufig aus Kostengründen nicht über große Displays und ergonomische Eingabegeräte. In diesem Feld setzen sich daher immer häufiger Interaktionskonzepte durch, die die Nutzeraufgaben auf weitere Geräte ausdehnen. So wird beispielsweise die Einrichtung von einem PDA aus vorgenommen, während die Programmierung bequem im Büro am Office-PC durchgeführt wird. Dennoch wird durch das Benutzungsmodell eine übergeordnete Benutzungsstruktur formuliert und dahingehend gekennzeichnet, welche Teilaufgaben durch bspw. welche Nutzergruppe an welchem Zugriffsort durchgeführt werden können.

Vorteil dieser Herangehensweise ist, dass der Nutzer nicht an allen Geräten unterschiedliche Interaktionsstrukturen vorfindet, sondern alle von einer gemeinsamen übergeordneten Struktur abgeleitet sind.

3.3.5 Ergebnis der Strukturgestaltung

Das Ergebnis der Strukturgestaltung ist die grundlegende Struktur des zukünftigen Bediensystems, die aber noch keine exakte Festlegung auf Soft- und Hardware-Plattform durchführt. In dieser noch frühen Phase des Useware-Engineering Prozesses werden lediglich initiale Hinweise zu möglichen Verknüpfungen zur Soft- und Hardware-Plattform beschrieben. Dieses sog. **Benutzungsmodell** entsteht im

Prozess der Strukturgestaltung schon sehr früh und wird im weiteren Verlauf dieser Entwicklungsphase immer weiter verfeinert. Grundlage für die aufgabenorientierte Struktur des Benutzungsmodells sind die Aufgaben der Nutzer, die in der Analysephase identifiziert wurden. Es ist damit für die nutzerorientierte Gestaltung des Bediensystems unabdingbar. Im weiteren Projektverlauf wird diese Aufgabenorientierung dann Basis für die gerätespezifischen Menüstrukturen sein. Merkmal des Benutzungsmodells ist, dass es Strukturen liefert, die plattform- und systemunabhängig für alle Plattformen/Produktfamilien angewendet werden und somit ein (innerhalb der Grenzen des Modells) intuitives und wieder erkennbares Bedienen ermöglicht.

Bereits in dieser Phase wird eine Evaluation des Entwicklungsergebnisses in Form des Benutzungsmodells vorgenommen. Sie zielt damit hauptsächlich auf die Übereinstimmung der in der Analyse erfassten und anschließend harmonisierten Strukturen mit den mentalen Modellen der Nutzer ab (Dutke 1994). Die Messung der Übereinstimmung kann bspw. mit strukturierten Interviews erfolgen, die Aussagen darüber liefern, ob die erarbeiteten Strukturen nutzungsfreundlich oder ob Änderungen erforderlich sind.

Die Evaluation der Struktur des Benutzungsmodells wird mit Strukturdarstellungen vorgenommen. Diese verlangen ein relativ hohes Abstraktionsvermögen beim Probanden. So ist es bereits in dieser Phase ratsam, den Nutzern erste Prototypen zur Verfügung zu stellen. Ihre Aufbereitung ist an dieser Stelle zwar zweitrangig, es können aber bereits erste Aussagen über die Plattform erhoben werden, die für die Auswahl wichtig sind.

Die Strukturgestaltung ist somit der letzte prinzipiell plattformunabhängige Schritt des Useware-Entwicklungsprozesses. Obwohl Strukturgestaltung und Plattformauswahl damit voneinander unabhängig sind, ist eine Überschneidung sinnvoll. Ansonsten werden eventuell zur Strukturevaluation Prototypen erstellt, die keine Gemeinsamkeiten mit der Zielplattform aufweisen. Um unnötige Aufwände zu vermeiden, sollten die Phasen synchronisiert und die Ergebnisse ausgetauscht werden.

Aus der Praxis: Typischerweise wird der Strukturgestaltung nur geringer Raum in den Entwicklungsprojekten eingeräumt. Die Erstellung einer tragfähigen Struktur auf Basis der Analysedaten hat aber eine große Bedeutung und sollte daher nicht zu kurz kommen. Als effizientes Mittel hat sich die Durchführung eines moderierten Workshops (2–3 Tage) unter Einbeziehung aller Projektbeteiligten (aber max. 7–8 Personen) erwiesen. Somit kann der Aufwand für die Strukturgestaltung sehr gering gehalten werden und gerade die Vereinheitlichung von Terminologien und Strukturen erfolgt im Konsens. Ein Moderator kann dazu beitragen, dass sich die Teilnehmer nicht in endlosen Diskussionen „verrennen“, sondern zielorientiert und schnell gearbeitet wird. Die im Rahmen des Workshops erarbeitete Grundstruktur wird dann im Rahmen der Gestaltungsphase weiter detailliert und iterativ verbessert.

Tab. 3.3 Hilfsmittel bei der Strukturgestaltung

Hilfsmittel	Eigenschaft
Strukturlege-Kärtchen	<ul style="list-style-type: none"> - Ähnlich der Strukturlegetechnik - Gut im Team einsetzbar - Schlecht zu vervielfältigen oder weiterzuverwenden
MS Excel o. ä.	<ul style="list-style-type: none"> - Tabellarische Anordnung und Beschreibung der Aufgaben - Bekanntes und überall vorhandenes Werkzeug - Wenig übersichtlich aufgrund der Bildschirmgröße
Modellierungswerzeuge (z. B. useML)	<ul style="list-style-type: none"> - Modellbasierte Beschreibung der Aufgaben - Schnelles Prototyping durch (semi-)automatische Generierung von Prototypen

3.3.6 Hilfsmittel für die Strukturgestaltung

Für die **strukturierte Aufarbeitung der Daten** der Strukturgestaltungsphase stehen unterschiedliche Werkzeuge zur Verfügung. Zu beachten ist bei deren Auswahl, ob und wie viel Aufwand für deren Einsatz erbracht werden muss. Dabei sind Aufwände zur Anwendung des Hilfsmittels selbst wie auch Aufwände zur Weiterverarbeitung der erzeugten Ergebnisse zu berücksichtigen.

Hilfsmittel aus dem Bereich der Modellierungstechniken können einerseits die Konsistenz der Daten sicherstellen und andererseits Medienbrüche, Redundanzen und Mehrfach-Arbeit vermeiden. Die Modellierung selbst bringt zunächst einmal kaum Vorteile, da jeder Entwickler sich mental ein eigenes Verständnis des Projekts schafft. Erst eine gemeinsame Modellierungsstrategie und deren Umsetzung in Form einer für jeden nutzbaren Modellierungssprache bzw. -technik fördert das gemeinsame Arbeiten am selben Projekt, die Kommunikation sowie den Informationsaustausch.

Mögliche Hilfsmittel für die Strukturgestaltung werden in Tab. 3.3 dargestellt

Aufgrund der Einfachheit in der Anwendung stellt die Verwendung von **Strukturlege-Kärtchen** zur Erarbeitung des Benutzungsmodells eine schnell umzusetzende Technik im Rahmen der Strukturgestaltung dar. Hierzu werden zunächst Vorlagen für die Aufgaben- und elementaren Benutzungsobjekte angefertigt (s. Abb. 3.16). Idealerweise werden ähnlich einem Formular direkt die relevanten Eigenschaften aufgeführt und damit deren Angabe erzwungen. Nachteil ist, dass die Daten zwar übersichtlich an Pinnwänden oder Flipcharts visualisiert werden, aber für die folgenden Arbeitsschritte nicht computerunterstützt weiterverwendbar sind.

Der Einsatz von **tabellenbasierten Software-Programmen** wie z. B. MS Excel stellt zwar eine grundsätzliche elektronische Verfügbarkeit der Daten sicher, reduziert allerdings die Übersichtlichkeit der Darstellung. Die Tabellenansichten sind relativ unkomfortabel und auf der kleinen Bildschirmfläche schlecht lesbar. Abhilfe schaffen da schon eher Werkzeuge auf XML-Basis, die mit entsprechenden For-



Abb. 3.16 Strukturlegekärtchen

matierungseinstellungen eine übersichtliche Darstellung bzw. einen übersichtlichen Ausdruck der Strukturen erlauben.

Ein weiteres Hilfsmittel für die Strukturgestaltungsphase stellen **Modellierungswerzeuge** wie z. B. die Useware Markup Language (useML) dar, die es ermöglicht, ein Benutzungsmodell zu beschreiben und gleichzeitig die Daten für folgende Projektphasen weiter zu verwenden (Reuther 2003; Meixner 2010). Die Benutzung des technischen Geräts wird hierbei mit Hilfe semantischer Beschreibungskonstrukte in einem Modell dokumentiert. Vorteil dieses Ansatzes ist, dass das Modell aufgrund der Semantik rechnergestützt auswertbar und maschinell weiter verarbeitbar ist. Eine umfassendere Darstellung des useML-Ansatzes befindet sich in Kap. 5.4.

3.4 Gestaltung

Durch die steigende Komplexität von Produkten sind Funktionen für den Nutzer (Bediener und Einrichter) immer schwerer zu erfassen. Durch Berücksichtigung nutzerspezifischer Anforderungen – wie in den vorangegangenen Abschnitten dargestellt – kann dieses Problem weitgehend gelöst werden: Die in der Analysephase erhobenen und in der Strukturgestaltung harmonisierten Ergebnisse stellen die Berücksichtigung der Nutzeranforderungen sicher und bilden damit die Basis für die Gestaltung.

Neben der Aufgaben- und Nutzerorientierung ist aber vor allem die ergonomische Gestaltung des Bediensystems von großer Bedeutung. Die Struktur eines Bediensystems und die Anordnung der Elemente tragen wesentlich zur Akzeptanz durch den Nutzer bei. Ihre Gestaltung kann das Erkennen und Verstehen positiv oder negativ beeinflussen.

Konkret beschäftigt sich die Gestaltung mit der ergonomischen Umsetzung des Benutzungsmodells auf eine Interaktionsplattform unter Berücksichtigung der in der Analysephase erhobenen Anforderungen. Dieser Prozess der Gestaltung umfasst das gesamte Design der Benutzungsschnittstelle. Dies beinhaltet die Auswahl der Hard- und Softwareplattform sowie der Interaktionsform, die Dialoggestaltung und die Entwicklung des grafischen Layouts.



Abb. 3.17 Bierzapfhähne in einer amerikanischen Bar

Aus der Praxis: Ein anschauliches Beispiel für das Ergebnis einer anforderungsgerechten Gestaltung zeigt Abb. 3.17 Der Wirt muss in jeder Situation intuitiv den richtigen Zapfhahn bedienen, vor allem, wenn das „Zapfen“ z. B. bei der Aufnahme einer weiteren Bestellung nur als Sekundärtätigkeit ausgeführt wird. Dabei spielen – nicht nur für dieses Beispiel – vor allem die Auswahl, Anordnung, Form und Farbe der Interaktionselemente eine wesentliche Rolle. Der Wirt kann den richtigen Zapfhahn in Form und Beschriftung sehen und auch fühlen. Dies verdeutlicht die Bedeutung der Gestaltung für die Umsetzung der Anforderungen aus der Analysephase und der Nutzungsstruktur aus dem Benutzungsmodell.

3.4.1 Vorgehensweise

Wie angesprochen, sind die in diesem Buch dargestellten Phasen nicht als streng sequentieller Ablauf zu verstehen, insbesondere die Strukturgestaltung und die Gestaltung weisen starke Wechselwirkungen auf. So wird z. B. für ein komplexes Benutzungsmodell in der Regel eine große Anzeigefläche benötigt und die Anzeigefläche determiniert im Gegenzug die Breite und Tiefe der Benutzungsstruktur. Es handelt sich hier also um einen verzahnten Prozess, bei dem Strukturgestaltung und Gestaltung im gewissen Rahmen als parallele Aufgabenstränge mit Interdependenzen zu verstehen sind.

Die Vielfalt der Entscheidungen, die in dem Prozessabschnitt „Gestaltung“ getroffen werden müssen, erfordert die besonders intensive Evaluation der Entwick-

Abb. 3.18 Inhalt der Gestaltungsphase



lungsarbeit durch die zukünftigen Nutzer. Wie Abb. 3.18 aufzeigt, stellt die Gestaltung die Schnittstelle zwischen der aufgabenorientierten Strukturgestaltung und der implementierungsorientierten Realisierung dar. Fehler, die bei der Grobkonzeption des Maschinenbediensystems gemacht werden, sind somit nur mit erhöhtem Aufwand zu beseitigen. Aus diesem Grund müssen die erarbeiteten Konzepte intensiv und zeitnah mit möglichen Nutzern diskutiert werden.

Den ersten Schritt der Gestaltung stellt die Auswahl der Bediensystemplattform da. Hierzu zählen unter anderem Rechnerhardware, Display- und Interaktionsgeräteauswahl.

Auf Basis dieser Plattform wird das Groblayout für das Bediensystem entwickelt, dies beinhaltet sowohl die Anordnung von Hardwareelementen als auch die grundsätzliche Maskenaufteilung bei bildschirmgestützten Bediensystemen. Parallel dazu ist das Navigationskonzept für das Interaktionssystem festzulegen. Diese Auslegung steht in engem Zusammenhang mit den in der Analysephase erhobenen Eigenschaften des primären Bedieners.

Im Anschluss daran sollte eine beispielhafte Umsetzung von einzelnen Dialogen erfolgen, die die grundsätzliche Handhabung mit dem System verdeutlichen. Eine Optimierung dieser Dialoge ist aufgrund der Anschaulichkeit einfach zu bewerkstelligen und für die spätere Bedienbarkeit des Systems von immenser Bedeutung.

Nach der Optimierung wird ein Katalog mit Interaktionsobjekten erstellt und die Interaktionsobjekte hinsichtlich ihrer Eignung zur Visualisierung der elementaren Aufgabenobjekte überprüft. In dieser Phase wird in Zusammenarbeit mit Designern das detaillierte Layout für das System inklusive der Farben festgelegt.

3.4.2 Plattformspezifikation

Dominierten vor einigen Jahren noch textuelle monochrome Benutzungsoberflächen mit Tastenbedienungen und weitestgehender Konzentration auf die unmittelbare Maschinensteuerung sowie -überwachung, so können heutige Neuentwicklungen auf ein breitgefächertes Angebot von einfachen 2-Zeilens-Textdisplays bis hin zu hochauflösenden Farbgrafik-Bildschirmen, von proprietären Echtzeitbetriebssystemen bis hin zu multiuser- und multitaskingfähigen Betriebssystemen mit direkter Manipulation und von der Tastenbedienung über Touch-Screen und Trackball bis hin zu Sprachsteuerung zurückgreifen. Der Auswahl der „richtigen“ Plattform für den Einsatzzweck kommt in diesem Zusammenhang eine große Bedeutung zu.

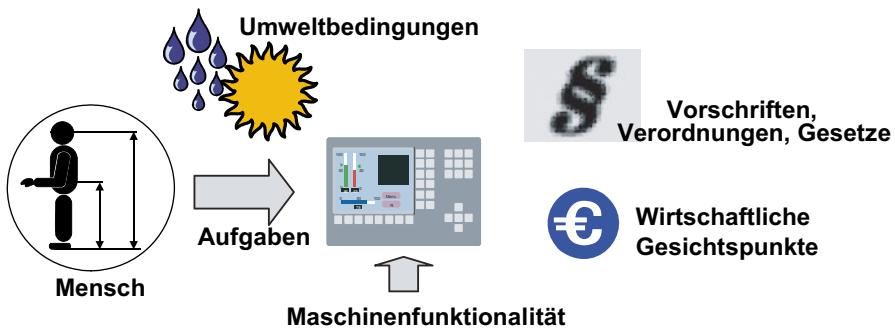


Abb. 3.19 Einflussgrößen bei der Gestaltung von Bediensystemen

Bei der Auswahl einer geeigneten Plattform sind eine Reihe an Einflussfaktoren zu berücksichtigen (s. Abb. 3.19).

Im Gegensatz zum Bürobereich (PC mit Maus, Tastatur, 19–24" Bildschirm) kann sich im Produktionsbereich aufgrund der stark differierenden Anforderungen keine Standard-Plattform etablieren. Die Einflussgrößen Maschine, Normen und Richtlinien, Wirtschaftlichkeit sowie Umweltbedingungen sind firmen- und produktspezifische Faktoren, welche meistens Individuallösungen bei der Hardwaregestaltung für den jeweiligen Einsatzbereich bedingen. Wird z. B. eine Maschine oder Anlage in Umgebungen mit hohem Verschmutzungsgrad eingesetzt, können verschmutzungsempfindliche Eingabeelemente wie z. B. Mäuse nicht eingesetzt werden. Weitere beispielhafte Einschränkungen in diesen Bereichen sind Größenverhältnisse (beispielsweise wird an einer kleinen Pumpe kein 18"-Display angebracht werden können) und Kostenaspekte (Maschinen unter großem Preisdruck verlangen auch preisgünstige Bediensysteme).

Im Rahmen der Plattformspezifikation sind insbesondere folgende Entscheidungen zu treffen:

- Steuerungsplattform (Eigenentwicklung, SPS, PC)
- Betriebssystem
- Entwicklungsumgebung
- Anzeigesystem
- Interaktionsgeräte und -techniken
- Einbaubedingungen (Festeinbau/Gondel, Lage: Bedien- vs. Visualisierungspriorität)

Da die Auswahl einer Bediensystemplattform von sehr vielen Randvariablen abhängig ist, werden im Folgenden nur einige Anmerkungen und Trends angeführt, die für die Auswahl relevant sind.

3.4.2.1 Steuerungsplattform

Die Steuerung hat in den letzten Jahren enorm an Bedeutung im Gesamtsystem Maschine gewonnen. Moderne Maschinenkonzepte sind mehr und mehr mechatro-

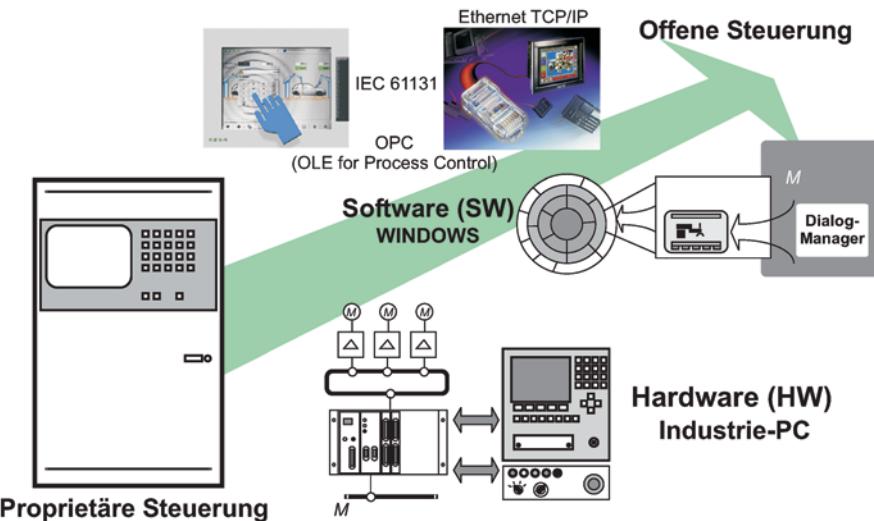


Abb. 3.20 Wandel in der Steuerungswelt

nische Systeme, in denen erst das Zusammenspiel von Mechanik und Elektronik zu den Leistungen führt, die der Kunde wünscht.

Bei ihrer Gestaltung muss ein Optimum zwischen den Produktlebenszyklen moderner Prozessoren mit 12–24 Monaten und Maschinenlaufzeiten von 20 Jahren und mehr gefunden werden.

Bislang war der Produktionsbereich gekennzeichnet durch proprietäre, herstellerspezifische Hard- und Softwarelösungen, die nur über wenige, meist nicht standardisierte Schnittstellen verfügten. Mittlerweile ist auch im Bereich der Produktionstechnik der Trend unverkennbar, diese herstellerspezifischen Lösungen soweit wie möglich durch offene Steuerungen auf Basis von Standards der PC-Technik zu ersetzen (s. Abb. 3.20).

Die Maschinenhersteller werden weiterhin Automatisierungsgeräte in den Produktklassen SPS, IPC und PC-basierte Steuerungen verwenden. Immer weniger Steuerungsanwender setzen vor diesem Hintergrund aber auf Eigenbau-Elektronik. Stattdessen wird auf leistungsfähige Standardtechnik gesetzt, die in der Lage ist, die durchsetzungsstärksten Technologien zu integrieren. Zu nennen sind hier vor allem Ethernet mit TCP/IP, OPC (UA), Web, die IEC 61131-Programmiersprachen sowie USB (zur Einbindung von Standardperipherie). Um zukunftssicher zu sein, müssen die Steuerungen in der Lage sein, eine Integration zukünftiger Technologien zu gewährleisten. Insbesondere die Steuerung auf Basis eines IPC wird zukünftig als Alternative zu proprietären SPS-Systemen an Bedeutung gewinnen.

3.4.2.2 Betriebssystem

War Ende der 90er Jahre noch klar der Trend zu (modifizierten) Office-Betriebssystemen zu erkennen, so wird die Auswahl heute differenzierter betrachtet.

Insbesondere die kurzzeitigen Innovationszyklen bei diesen Betriebssystemen mit der Notwendigkeit von ständigen Updates sind in der Automatisierungstechnik unerwünscht. Die in der PC-basierten Steuerungstechnik genutzten Betriebssysteme (WINDOWS oder LINUX) werden die proprietären Systeme nicht vollständig vom Markt verdrängen können. Im Folgenden sind die Vor- und Nachteile einer PC-basierten Lösung aufgeführt:

- + Verbreitetes und (quasi) standardisiertes Betriebssystem
- + Grafische Benutzeroberfläche integriert
- + Kurze Einarbeitungszeit aufgrund weiter Verbreitung
- + Immer aktuelle Hardware am Markt
- Beschränkte Echtzeitfähigkeit
- Schnelle Innovationszyklen/Notwendigkeit zum Update

3.4.2.3 Entwicklungsumgebung

Um eventuellen Kompatibilitätsproblemen aus dem Weg zu gehen, werden Betriebssystem und Entwicklungswerkzeug zumeist aus einer Hand gekauft. Im Falle der PC-basierten Betriebssysteme impliziert dies jedoch nicht, dass zukünftige Benutzeroberflächen aus den vordefinierten Oberflächenobjekten z. B. der WPF-Klassenbibliothek aufgebaut sind. Von verschiedenen Herstellern werden Erweiterungsbibliotheken für unterschiedliche Anwendungsfelder angeboten (z. B. VISI-WINNET (INOSOFT GmbH), INVISU (epro GmbH) und auch die Anpassung oder Eigenentwicklung von Objekten ist möglich.

Dennoch wird sich der Trend zur Anwendungsprogrammierung mit standardisierten Softwaretools verstärken. Dadurch können Projektierungs- bzw. Engineering-systemen (z. B. INTOUCH (Wonderware GmbH), WINCC (Siemens AG)) größere Chancen eingeräumt werden, weil diese die Arbeitsmöglichkeiten optimieren und zusätzlich die verschiedenen Kommunikationsbeziehungen integrieren.

3.4.2.4 Anzeigesystem

Dem Anzeigesystem kommt mit der steigenden Funktionalität eine immer wichtigere Rolle zu. Als primäre Schnittstelle zwischen Maschine und Mensch besitzt der Bildschirm eine zentrale Bedeutung für die Akzeptanz des Bediensystems. In der Regel kommen LCD-Bildschirme zum Einsatz, die nicht nur zunehmend herkömmliche Kathodenstrahlbildschirme (CRT) ersetzen, sondern aufgrund ihrer überlegenen Eigenschaften, wie z. B. Platz- und Gewichtersparnis, Energieverbrauch usw., neue Einsatzfelder ermöglichen.

Aufgrund der großen Stückzahlen aus dem Konsumerbereich (Notebooks, Handys, PDAs, Digitalkameras) sind diese Bildschirme in unterschiedlichen Größen günstig verfügbar. Reine Textdisplays und SW-Bildschirme werden immer stärker durch grafikfähige Farbdisplays substituiert, die mehr Möglichkeiten und Flexibilität in der Entwicklung liefern. Insofern wird die wesentliche Entscheidung im Be-



Abb. 3.21 Interaktionsgeräte in der Industrie. (Krauss 2003)

reich der Anzeigesysteme die Größenauswahl sein. Maßgebliche Kriterien für diese Auswahl sind die Anforderungen der Nutzer, die Komplexität des Benutzungsmodells sowie der Preis und die Größe in Bezug zum Gesamtsystem.

Für die Nutzerpräferenzen können die in der Analysephase gewonnenen Erkenntnisse dienen. Bei stark vernetzten Benutzungsmodellen mit sehr vielen Benutzungsobjekten ist eine entsprechend große Bildschirmfläche zur Visualisierung von Informationszusammenhängen meist unabdingbar.

3.4.2.5 Interaktionsgeräte und -techniken

Im Gegensatz zum Bürobereich, in dem die Tastatur und die Maus den Standard darstellen, hat sich in der Produktionstechnik bislang kein einheitlicher Standard für die Interaktion herauskristallisiert (s. Abb. 3.21). Die Ursachen hierfür sind:

- Die im Bürobereich etablierte Dialogform „WIMP“ (Windows, Icon, Mouse, Pointer) kann im industriellen Umfeld nicht eingesetzt werden, da Aktivitäten sowohl vom Nutzer als auch vom Prozess bzw. der Maschine ausgehen können.
- Die Interaktionsanforderungen unterscheiden sich sowohl vom Bürobereich als auch untereinander. Diese Interaktionsanforderungen, die aus der Summe der zur bedienenden Anwendungen resultieren, sind maßgeblich für die Auswahl

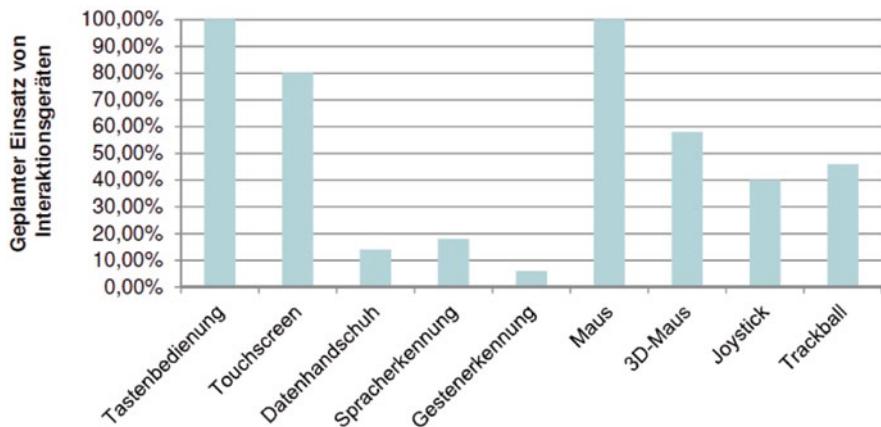


Abb. 3.22 Trends für den Einsatz von Interaktionsgeräten in der Produktionstechnik

der notwendigen Interaktionselemente heranzuziehen. Diese Interaktionsanforderungen sind implizit im Benutzungsmodell enthalten.

Bei einer vom Verfasser im Jahre 2009 durchgeführten Umfrage zum Thema „Entwicklungsrends bei Bediensystemen in der Produktionstechnik“ stand unter anderem die Frage nach den zukünftigen Interaktionselementen im Mittelpunkt (s. Abb. 3.22).

Die Ergebnisse belegen, dass auch zukünftig die Tastenbedienung (Funktions-tasten, Softkeys, Keyboard) als Interaktionstechnik bei Bediensystemen vorrangig verwendet werden. Die Nutzung des Eingabegerätes „Mouse“ hat im industriellen Umfeld, im Vergleich zur im Jahre 2000 durchgeführten Studie (Oortmann 2000), erheblich zugenommen.

Die Erwartungen, die im Jahre 2000 an den Einsatz von Touchscreen-Systemen gesetzt wurden (65 % aller Befragten erwarteten die Integration eines Touchscreens in industrielle Bediensysteme), sind Realität geworden (Oortmann 2000). Die im Jahre 2009 durchgeführte Studie zeigt, dass 80 % der befragten Personen den Einsatz von Touchscreen-Systemen planen bzw. im Einsatz haben. Dies zeigt, dass sich solche Systeme in der Industrie etabliert haben. Neuartige Interaktionsgeräte wie bspw. 3D-Maus, Datenhandschuh sowie Sprach- und Gestenerkennungssysteme zur multimodalen und 3-dimensionalen Interaktion befinden sich auf dem Vormarsch zur Integration in die Produktionstechnik von morgen.

Aufgrund der großen Relevanz von Touchscreen-Systemen wird in Kap. 6.3.3 näher auf die Touchscreen-Technologie eingegangen. Generell ist davon auszugehen, dass koordinatengebende Eingabegeräte immer häufiger auch in Steuerungen zu finden sein werden. Die Präferenzen für solche Eingabegeräte variieren bei den befragten Personen stark, sind in Summe aber von hoher Relevanz. Ein kurzer Überblick über koordinatengebende Eingabegeräte wird in Kap. 6.3.3 gegeben, ausführlichere Informationen zur Auswahl eines Gerätes finden sich in (Krauss 2003).

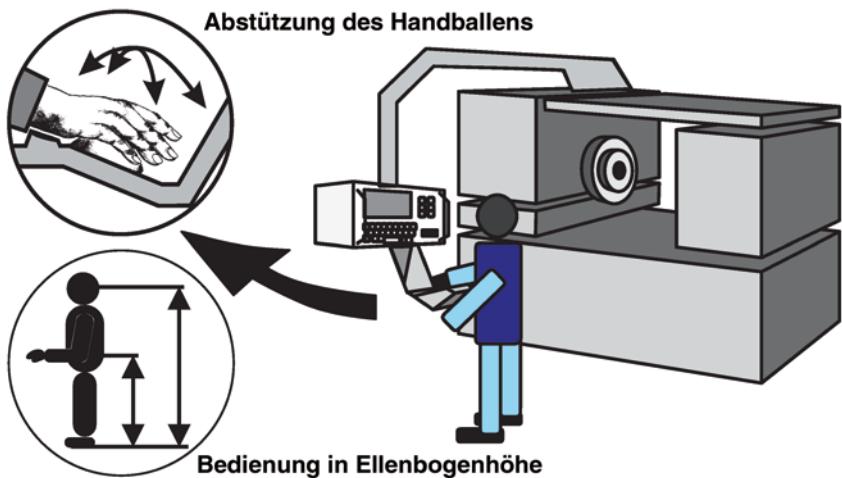


Abb. 3.23 Bedienhaltung an einer Werkzeugmaschine

3.4.2.6 Einbaubedingungen

Insbesondere bei Maschinensteuerungen, bei denen das Bedienfeld zumeist senkrecht und häufig auch in schwenkbaren Konsolen untergebracht wird, müssen die ergonomischen Anforderungen besonders bedacht werden. Ergonomische Anforderungen resultieren in einer ganzheitlichen Betrachtungsweise aus den Fachgebieten Anthropometrie, Physiologie und Psychologie. In Bezug auf den Menschen muss das Hardwarekonzept so gestaltet sein, dass der Mensch aufgabenorientiert, zweckentsprechend, ermüdungsfrei und problemlos – kurz benutzerfreundlich – mit dem System interagieren kann.

Bei allen Gerätegruppen ist die Benutzungsfreundlichkeit und die Präzision der Positionierung von individuellen Merkmalen wie herstellerspezifische Bauform und charakteristischer Anordnung des Eingabeelements im Handlungsräum des Menschen abhängig. Neben der konstruktiven Detailgestaltung der Bauform hinsichtlich einer ergonomischen Anordnung von Zeigeelement und Schaltelement ist die Auslegung der Eingabegeräte hinsichtlich der Betätigungskräfte und -momente (bzw. Dämpfung) von großer Bedeutung. So können Geräte von unterschiedlichen Herstellern stark unterschiedliche Ergebnisse bringen. Die Steuerung feinfühliger Bewegungen ist stark von der Hand- und Armhaltung abhängig. Die Bedienung sollte ideal in natürlicher Ellbogenhöhe erfolgen. Dabei kann eine Abstützung des Handballens zur feinfühligen Eingabe über die Finger beitragen (s. Abb. 3.23).

Befindet sich das Bedienfeld in einer schwenkbaren Konsole, so muss sicher gestellt sein, dass diese sich beim Bedienvorgang nicht bewegen kann. Bereits ein leichtes Schwingen macht feinmotorische Eingabeoperationen, wie sie für die Navigation erforderlich sind, unmöglich. Dieses fällt insbesondere bei den meist

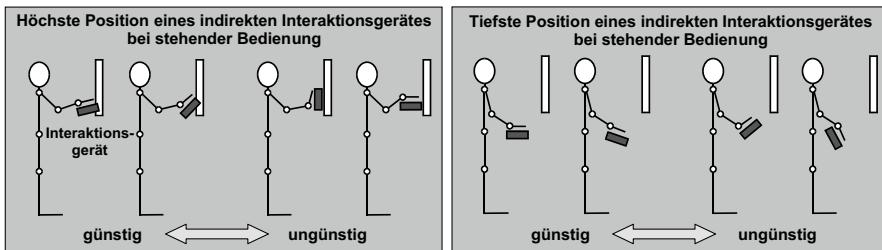


Abb. 3.24 Positionen für Bediensysteme und Interaktionsgeräte. (Krauss 2003)

kraftgesteuerten (isometrischen) Mouse-Buttons auf, die mittlerweile von einigen Steuerungsherstellern angeboten werden. Um bei dem recht hohen Kraftaufwand in zwei Freiheitsgraden entsprechend feinfühlig steuern zu können, bedarf es einer absolut steifen Konsolenkonstruktion.

Bei der Beurteilung der Gerätelocation sind neben der Gerätegestaltung Abstüttmöglichkeiten für das Hand-Arm-System, Höhe und Neigung des Gerätes sowie die Distanz zum Benutzer wichtige Dimensionen für die Gebrauchstauglichkeit des speziellen Gerätes (Abb. 3.24). Maße zur Anordnung der Gerätelocation können nur bedingt angegeben werden. Erst auf Grundlage detaillierter Angaben über die tätigkeitsspezifische Nutzung des Interaktionsgerätes können Entscheidungen darüber gefällt werden, welche Gewichtungen den verschiedenen Aspekten der Arbeitsplatzgestaltung beizumessen sind. Weiterhin können einzelne Faktoren sich gegenseitig beeinflussen.

Tiefergehende Erläuterung zu den Einbaubedingungen von Bedienfeldern finden sich in der Richtlinie VDI/VDE 3850 (vgl. Kap. 6.1.2).

Aus der Praxis: Zwei Beispiele sollen aufzeigen, welche Bedeutung die Hardwareauswahl und die projektbegleitende Evaluation besitzen:

1. Ein Maschinenhersteller wollte in seiner Neuentwicklung erstmals ein koordinatengebendes Eingabegerät einsetzen. Er beauftragte einen Berater mit der Auswahl eines geeigneten Gerätes. Die Eignung wurde dabei an einer vorhandenen pultförmigen Steuerung im Labor erprobt. Ein Gerät wurde ausgewählt und bzgl. Einbaulage und Handabstützung optimiert. Im weiteren Entwicklungsprozess entschied sich der Hersteller dann aber für eine senkrechte Gondelmontage, mit der Folge, dass die Maschinenbediener das neue Interaktionssystem als ungewohnt bis schmerhaft beschrieben. Zudem sorgte die geringe Steifigkeit der Gondelkonstruktion für ein hohes Maß an Fehlbedienungen durch die Schwingung bei der Betätigung.
2. Bei der Entwicklung eines neuen Flugzeugeckopits wurde als Bedienelement erstmalig ein grafikfähiges Display mit Touchpad vorgesehen. In den ersten Tests waren die Ergebnisse dieser Entwicklung sehr viel versprechend, so dass diese Lösung für die Serie vorgesehen war. Erst in den abschließenden, Anwendertests tauchten Probleme auf.

Hierbei wurde das neue Cockpit in einem Simulator realisiert, um auch die Bedienbarkeit unter realen Flugbedingungen (Flugbewegungen, Schwingungen) zu testen. Als Testpersonen wurden Piloten von verschiedenen Airlines eingeladen. Die Bedienbarkeit wurde dann von den meisten Piloten auch unter Flugbedingungen sehr positiv beurteilt. Nur die Piloten einiger asiatischer Airlines hatten Probleme mit dem Touchpad. Bei der Auswahl der Hardware hatte sich der Flugzeughersteller für ein kapazitives Touchpad entschieden, das nur auf direkten Hautkontakt reagiert. Es stellte sich heraus, dass die Piloten der asiatischen Airlines grundsätzlich in Baumwollhandschuhen fliegen, was eine Funktionsauslösung verhinderte. Kurzfristig löste man dieses Problem sehr pragmatisch, indem man die Fingerkuppen der Handschuhe abschnitt. Für die spätere Serie wurde das kapazitive Touchpad durch ein resistives ersetzt, welches auch eine Bedienung mit Handschuhen zulässt.

3.4.3 *Plattformspezifisches Benutzungsmodell*

In der Strukturgestaltung ist ein plattformübergreifendes Benutzungsmodell erstellt worden. Dieses leitet sich konsequent aus den Handlungsbeschreibungen bzw. dem Benutzungsmodell der Strukturgestaltungsphase ab, jedoch fließen in die Strukturierung der Navigation noch weitere Anforderungen an Funktionszugriffe, Zugriffshäufigkeiten etc. mit ein. Auch die Realisierungshardware und die gewählte Navigationsstrategie stellen Randbedingungen an die Navigationsstruktur.

Zwar wurden Anforderungen an die Hardware aus Sicht der Handlungen, Funktionen und Informationen bereits in der Strukturgestaltungsphase teilweise berücksichtigt und im Benutzungsmodell in Form einer „groben“ Bedienstruktur festgehalten, ihre genaue Betrachtung findet aber hier statt.

Eine Navigationsstruktur besteht aus Gruppen von Aktions- und Eigenschaftsrepräsentanten, die zur Strukturierung der Gesamtfunktionalität eines Systems in überschaubare Mengen dienen (Balzert 1996). Eine Navigationsstruktur bietet einen situationsbezogenen Zugriff auf Untermengen von Funktionen und Datenstrukturen.

Knoten der Navigation können hierarchisch oder als Netzwerk strukturiert werden (s. Abb. 3.25), wobei einfache Abfolgen als spezialisierte **Baumstrukturen** angesehen werden können. Prinzipielle Unterschiede bestehen dagegen zwischen Baumstrukturen und **Netzwerkstrukturen**. In Baumstrukturen sind die einzelnen Knotenstellen nur entlang der Baumzweige erreichbar, so dass man zu einem anderen Knoten der gleichen Ebene nur durch den Rücksprung auf die darüber liegende Ebene gelangen kann. Netzwerke besitzen nicht nur Verbindungen über die Zweige eines Baumes, sondern auch direkte Verbindungen zwischen verschiedenen Hauptästen der Baumstruktur unter Umgehung der gemeinsamen Wurzeln der Zweige. Zu jeder Knotenstelle kann es folglich mehrere Zugänge von beliebigen anderen Knotenstellen geben. Damit bieten Netzwerke zwar große Handlungsfreiheiten für

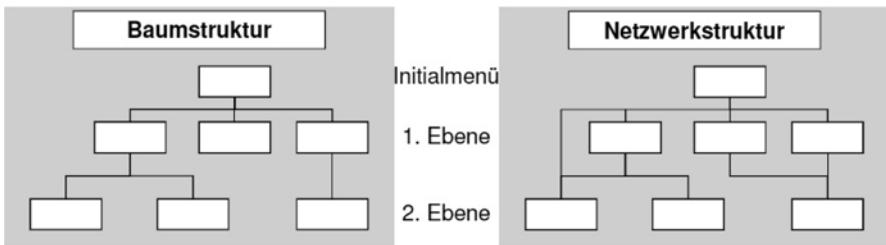


Abb. 3.25 Navigationsstrukturen

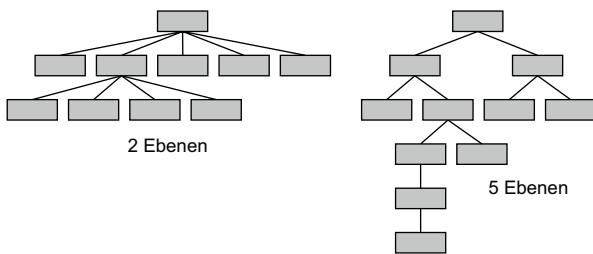
Bediener, allerdings sind sie auch schwerer durchschaubar als Baumstrukturen, so dass die Gefahr der Desorientierung besteht.

3.4.4 *Navigationskonzept*

Im **Benutzungsmodell** werden Funktionen und Anzeigeeinformationen gemäß ihrem Handlungsbezug und sicherheitstechnischer Anforderungen hierarchisch strukturiert. Die Entwicklung der Navigationsstruktur baut darauf auf, fasst Funktionen und Informationen zu einzelnen Ansichten einer Bedienschnittstelle zusammen und bestimmt durch die Beziehungen zwischen den Knoten die Navigationsstruktur. Unter Berücksichtigung der Hardwarerandbedingungen und der Bedienstruktur sind nun folgende Schritte zu Entwicklung der Navigationsstruktur durchzuführen (Wahl 2000):

- Bestimmung der globalen Informationen und Aktionen:
Informationen und Aktionen, die (fast) überall benötigt oder verwendet werden, können als global gekennzeichnet werden. Diese sind die späteren Inhalte des globalen Bereiches.
- Bestimmung der Hauptbedienbereiche der Navigationsstruktur:
Dieser Schritt besteht aus der Definition der Übersichtsmasken der Navigationsstruktur. Die Grobgliederung einer Navigationsstruktur für CNC-Maschinen und produktionstechnischen Anlagen sollte auf Bedienbereichen basieren, die den Hauptaufgabenbereichen der Nutzer entsprechen. Diese sind dem Benutzungsmodell direkt zu entnehmen oder aus ihr abzuleiten.
- Bestimmung der Inhalte der Startansichten:
Inhalte von Hauptbedienbereichen der Navigationsstruktur sind im nächsten Schritt zu bestimmen. Übersichtsmasken stehen an erster Stelle jedes Bedienbereiches und markieren den Einstieg in die entsprechenden Teilaufgaben. In diesen Ansichten sind üblicherweise diejenigen Informationen zusammengestellt, aus denen der Handlungsbedarf für zu Teilaufgaben gehörende Tätigkeiten abgeleitet wird.

Abb. 3.26 Breite und tiefe Navigationsstruktur



- Übertragung des Benutzungsmodells in die Navigationsstruktur:

Nachdem die Hauptbedienbereiche sowie die Inhalte der Einstiegsansichten festgelegt sind, kann die eigentliche Übertragung des Benutzungsmodells in die Navigationsstruktur erfolgen. Zunächst werden dazu die Knotenstellen des Benutzungsmodells, d. h. die Tätigkeiten und Handlungen den Bedienbereichen zugeordnet, dort jeweils als Knoten eingetragen und hierarchisch angeordnet. Die dadurch generierte Navigationsstruktur wird durch die Parameter Breite und Tiefe charakterisiert. Die Breite gibt in diesem Zusammenhang die Optionsanzahl je Knoten an. Die Tiefe beschreibt die maximale Anzahl der Ebenen. Breite und Tiefe beeinflussen die Geschwindigkeit der Optionssuche und die Übersichtlichkeit der Navigationsstruktur in erheblichem Maße. Unter Optionen ist die Summe aller Verzweigungsmöglichkeiten und Funktionen eines Knotens zu verstehen.

Eine große Tiefe erlaubt die Gliederung in kleine übersichtliche Knoten, aus denen leichter ausgewählt werden kann. Gleichzeitig verursacht sie jedoch eine größere Entscheidungs- und Bedienzeit zum Auslösen der gesuchten Option, da eine größere Folge von Knoten durchlaufen werden muss. Außerdem wächst die Gefahr der Unsicherheit darüber, in welchem Zweig eine gesuchte Option zu finden ist (Wandmacher 1993).

Eine große Breite reduziert dagegen zwar die Nebenzeiten zum Aufrufen und Aufbauen der Ansichten, allerdings steigt die Suchzeit, da nun mehr Optionen pro Knoten vom Bediener zu erfassen sind (Geiser 1990).

Als günstig hat sich eine Breite zwischen vier und sieben Optionen erwiesen. Die Tiefe sollte nicht über drei Ebenen hinausgehen (s. Abb. 3.26).

Die Auswahl der Navigationselemente sollte nach der Breite bzw. Tiefe der Hierarchie und dem auf dem Ausgabegerät zur Verfügung stehenden Platz erfolgen. Abbildung 3.28 zeigt den Platzbedarf von unterschiedlichen Navigationselementen.

Die Baumdarstellung, wie sie der WINDOWS-Benutzer vom EXPLORER her kennt, ist besonders gut zur Navigation geeignet, sofern entsprechender Platz vorhanden ist (Abb. 3.27). Der Grund hierfür ist einerseits die benutzerfreundliche Visualisierung selbst komplexer Hierarchien, andererseits die Tatsache, dass der Baum gleichzeitig für die Orientierung des Benutzers sorgt.

Liegen Platzverhältnisse vor, die den Einsatz der Baumdarstellung nicht zulassen, muss die Navigation über eine Schalterleiste erfolgen, die je nach Hierarchieebene

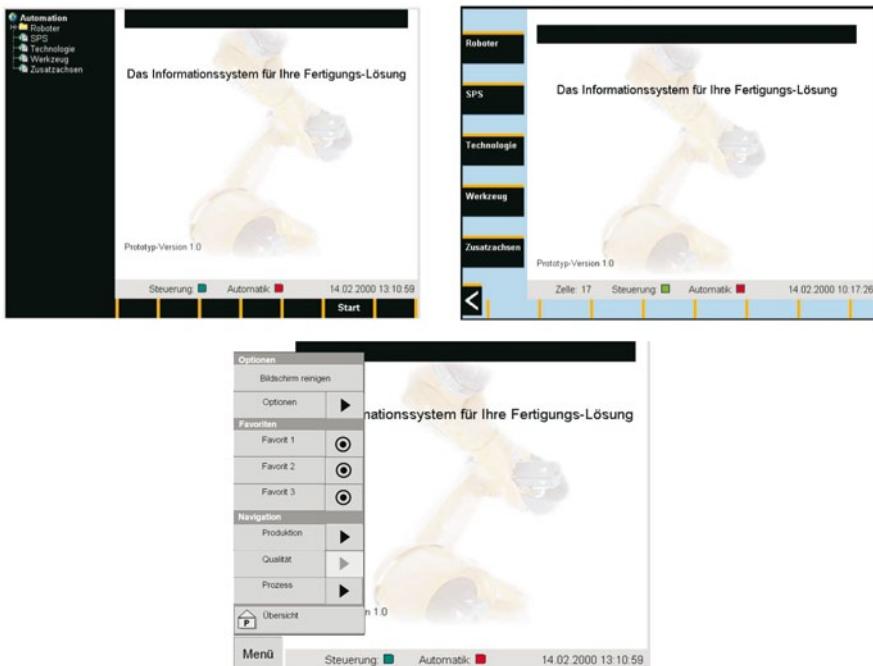
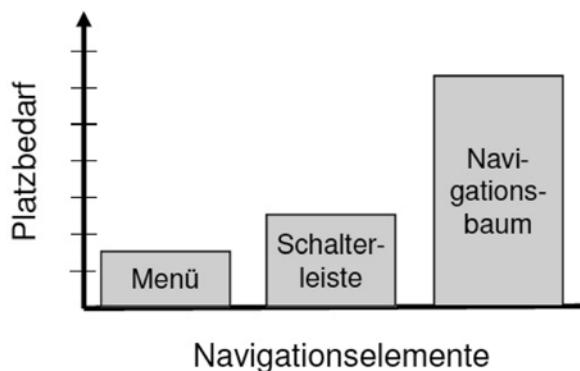


Abb. 3.27 Navigationsbaum vs. Schalterleiste vs. Fly-Out-Menü. (Quelle: KUKA ROBOTER)

Abb. 3.28 Visualisierungskomponenten für die Navigation



die in diesem Kontext möglichen Navigationsziele darstellt (Abb. 3.27). Neben der Schalterleiste kann zudem noch der Informationsbereich angeordnet werden.

Bei extrem knappen Platzverhältnissen, wie z. B. auf dem Display eines Kleingerätes muss die Navigation in Menüform den gesamten Bildschirmbereich beanspruchen, um noch lesbar zu sein und eine ausreichende Menübreite zu realisieren. Der Benutzer wählt somit das Sprungziel aus und daraufhin wird die entsprechende Information angezeigt, bis er erneut die Navigation aufruft. Diese Navigationsform

erfordert zudem jeweils die Visualisierung des Kontextes, da diesem aufgrund der Platzverhältnisse kein expliziter Bildschirm-Bereich zugewiesen werden kann.

Ein Bediensystem und damit auch das Navigationskonzept soll den Nutzer bei der Ausführung seiner Aufgaben unterstützen. Anfänger kommen besser mit der relativen Navigation zurecht. Die Vorgehensweise der relativen Navigation „vom Groben zum Detail“ ist für sie geeignet, um ihnen über den jeweiligen Sachverhalt zunächst einen Überblick zu geben. Dies unterstützt das „Erforschen“ des Systems. Der Experte benötigt häufig einen schnellen Zugriff auf die Details, die er im Rahmen seiner Arbeitsaufgabe manipulieren oder einsehen muss. Dies wird durch eine absolute Navigation besser unterstützt, da diese es ihm ermöglicht, direkt die Seiten aufzurufen, auf denen die benötigten Informationen zu finden sind. Beiden Nutzergruppen ist gemein, dass ihre Aufgaben vornehmlich in einem bestimmten Bedienbereich durchführen, weshalb die Bedienbereichswahl die oberste Navigationsebene darstellen sollte.

3.4.5 Strukturierung von Informationen

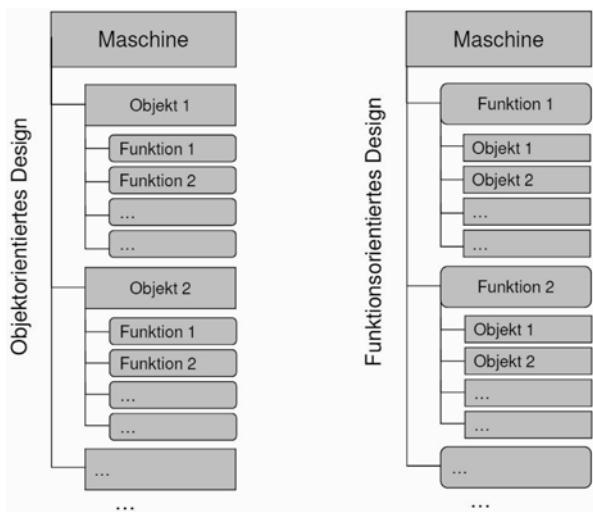
Die Definition von Navigationsstrukturen und Bildschirmhalten verfolgt im Wesentlichen die Ziele der Aufgabenangemessenheit und der Handlungsorientierung. Die Zusammenstellung und Anordnung der Maskeninhalte ist also abhängig von der Tätigkeit des Benutzers und dem dabei auftretenden Interaktions- und Informationsbedarf. Bei der Gestaltung der Masken ist anzustreben, dass der Benutzer die vorgesehenen Handlungen vollständig durchführen kann, d. h. die Maske sollte ihm erlauben, sowohl die Zielsetzung seiner Handlung festzulegen, als auch die Handlung durchzuführen und ihr Ergebnis zu kontrollieren.

Bei komplexen Maschinen und Anlagen tritt oftmals die Problematik auf, dass mehr Informationen und Funktionen für eine Tätigkeit benötigt werden, als auf einer Maske dargestellt werden können. Es ist zwar durchaus sinnvoll, möglichst viele zusammengehörende Informationen auf einer Maske anzubieten, dies muss aber abgewogen werden gegenüber der begrenzten Darstellungsfläche und der Anforderung, die benötigten Informationen noch schnell identifizieren zu können.

Masken von Bediensystemen lassen sich daher sowohl nach objekt- als auch nach funktionsorientierten Gesichtspunkten strukturieren (s. Abb. 3.29). Welcher der beiden Begriffe „Objekt“ oder „Funktion“ im Mittelpunkt der Betrachtung steht, stellt ein wesentliches Charakteristikum der Interaktionsstruktur dar.

- Beim objektorientierten Design identifiziert der Benutzer zuerst das Objekt und dann die Funktion(en), die mit diesem Objekt durchzuführen sind.
Dieses Prinzip bietet dann Vorteile, wenn viele (gleichartige) Objekte existieren, auf die nacheinander mehrere Funktionen angewendet werden müssen (wie z. B. das Einrichten einer Maschinenkomponente). In diesem Fall muss das Objekt also nicht bei jeder Funktionsbenutzung neu angegeben werden und die Anzahl der Bedienhandlungen wird reduziert.

Abb. 3.29 Objektorientiertes vs. Funktionsorientiertes Design



- Beim funktionsorientierten Design wählt der Benutzer zunächst die Funktion aus und dann die Objekte, auf die die Funktion anzuwenden ist.
Das funktionsorientierte Design ist immer dann vorzuziehen, wenn eine Vielzahl von unterschiedlichen Objekten existiert, auf die jeweils eine Teilmenge von Funktionen angewendet werden kann. In diesem Fall wird durch das Anwählen der Funktion die Menge der auswählbaren Objekte stark eingeschränkt, so dass die Objektauswahl übersichtlicher und damit effizienter wird. Weiterhin ist dieses Prinzip von Vorteil, wenn auf eine Vielzahl gleichartiger Objekte dieselbe Funktion angewendet werden soll (z. B. Heizungseinstellung an der gesamten Maschine). In diesem Fall verringert sich auch die Anzahl der Handlungen.

3.4.6 Klassifizierung von Informationen

Die Aufmerksamkeitslenkung sollte so ausgelegt sein, dass der Nutzer sehr schnell und einfach zwischen wichtigen und unwichtigen Informationen unterscheiden kann. Dazu lassen sich Informationen/Meldungen in verschiedene Informationsklassen einordnen, in denen Sie dann gemeinsame Merkmalskriterien definieren und welche insbesondere definieren, wie der Nutzer reagieren muss. Man unterscheidet zwischen reaktionspflichtigen, bestätigungs pflichtigen und nichtbestätigungs pflichtigen Informationen.

- Bei **reaktionspflichtigen Informationen** ist ein Eingriff des Nutzers (z. B. bei kritischen Maschinenzuständen) zwingend erforderlich, um den sicheren Betrieb der Anlage zu gewährleisten.
- Bei **bestätigungs pflichtigen Informationen** muss der Bediener lediglich bestätigen, dass er die Information zur Kenntnis genommen hat (z. B. Wartungsintervall überschritten).

- **Nichtbestätigungspflichtige Informationen** haben reinen Hinweischarakter. Die Informationen werden vom Bediensystem zur Verfügung gestellt und dem Bediener visualisiert. Es bleibt ihm aber freigestellt, ob er die Information (z. B. produzierte Stückzahl) zur Kenntnis nimmt oder nicht.

Die Zuordnung von Informationen zu Informationsklassen soll dem Benutzer helfen, die Priorität der jeweiligen Informationen im Hinblick auf ihre Wichtigkeit für den Betrieb des Systems eindeutig und schnell zu erkennen. Die Informationsklassifizierung ist aber nicht nur für den Nutzer von Bedeutung, auch der Entwickler wird so gezwungen, eine klare Strukturierung durchzuführen. In diesem Zusammenhang werden nach VDI 3850 die Klassen Alarm, Warnung, Statusinformation, Hinweisinformation, Bedienfehlerinformation und allgemeine Information unterschieden. In der Praxis sind bei einfachen Systemen auch weniger Klassen ausreichend.

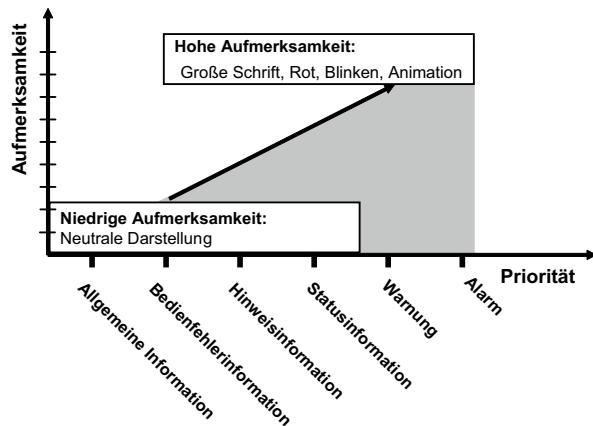
- **Alarne** haben die höchste Priorität, da sie auf einen kritischen Maschinenzustand verweisen.
- **Warnungen** sind Meldungen über mögliche oder sich anbahnende kritische Maschinenzustände.
- **Statusinformationen** beschreiben den aktuellen Zustand von Bearbeitungsprozess und Maschine.
- **Hinweisinformationen** geben dem Nutzer Hinweise zum aktuellen Geschehen und zum weiteren Vorgehen. (z. B. „Achsen mit Cursortasten verfahren“).
- **Bedienfehlerinformationen** machen den Nutzer auf eine fehlerhafte Bedienung aufmerksam.
- **Hilfeinformationen** sind Informationen zur Unterstützung des Nutzers, die auf Anforderung dargestellt werden.
- **Allgemeine Informationen** sind Informationen niedriger Priorität, die keiner der anderen Klassen zugeordnet werden können.

Die Informationsklassifizierung ist eine wichtige Basis für die konsistente Zuordnung von Darstellungsattributen. Deshalb sollte die Informationsklassifizierung einen eigenen Designschritt darstellen.

Der Zusammenhang zwischen Priorität der Informationen und der Aufmerksamkeit wird in (Abb. 3.30) verdeutlicht.

Die richtige Einordnung der Informationen in die entsprechenden Informationsklassen ist von hoher Bedeutung für den Nutzer des Bediensystems. Vor allem für den Automatikbetrieb einer Anlage, bei dem der Benutzer lediglich eine Überwachungsfunktion übernimmt, ist die Aufmerksamkeitssteuerung über die Informationscodierung von Bedeutung: Da es nicht möglich ist, die Aufmerksamkeit ständig auf höchstem Niveau zu halten, müssen wichtige Informationen einen Reiz beim Benutzer auslösen, der die Aufmerksamkeit erhöht.

Abb. 3.30 Zusammenhang von Priorität und Aufmerksamkeit



Die Informationsklassifizierung stellt eine wichtige Basis für die konsistente Zuordnung von Darstellungsattributen dar. Die im Folgenden aufgeführten Möglichkeiten der Informationscodierung sind nach dem Gesichtspunkt der zuverlässigen Informationserkennung gewählt. In der Praxis können weitere Gestaltungsaspekte wie z. B. Ästhetik oder Firmenfarben von Bedeutung sein. Um die Informationen unterschiedlicher Prioritäten voneinander abzugrenzen, stehen verschiedene Codierungsmöglichkeiten zur Verfügung:

- Farbe, Intensität
- Gestalt
 - Schriften
 - Bildzeichen, Symbole
 - Orientierung, Position
 - Linien und Schraffuren
 - Form
- Blinken, Animation
- Akustik (Warnton: Hupe, Sirene ...)

Durch die Verwendung verschiedener Möglichkeiten der Informationscodierung kann die Aufmerksamkeit des Nutzers gezielt gelenkt werden. Grundlage dafür ist allerdings ein konsistentes Konzept für die Zuordnung von Informationsklassen zu Darstellungsattributen (s. Kap. 6.1.2). Eine steigende Auffälligkeit von Meldungen kann z. B. mit folgender Verwendung der Informationscodierung erreicht werden:

- Informationen können durch die gezielte Verwendung von Farbe einer Klasse oder einer bestimmten Auffälligkeit verknüpft werden (z. B. Blau für Hinweise, Rot für Alarne, Orange für Warnungen).
- Durch das Steigern der Leuchtdichte kann eine höhere Auffälligkeit erzielt werden.
- Durch Blinken oder Animation wird die Auffälligkeit weiter gesteigert.
- Die höchste Aufmerksamkeit kann durch die Verwendung eines akustischen Signals erreicht werden. Dadurch kann auch die Aufmerksamkeit erlangt werden, wenn kein Sichtkontakt zum Bediensystem besteht.

Durch einen redundanten Einsatz verschiedener Attribute kann die Wirkung der Aufmerksamkeitslenkung weiter gesteigert werden (z. B. Blinken und Warnton gleichzeitig). Bei der Auslegung eines Stufenmodells in dieser Form sind allerdings einige Aspekte zu berücksichtigen:

- Bei einer Farbcodierung ergibt sich die Wirkung (d. h. Auffälligkeit) aus Farbfläche und Intensität.
- Bei dem Einsatz unterschiedlicher Leuchtdichten ist ein Kompromiss zwischen erkennbaren Unterschieden und übertriebenen Kontrasten zu finden.
- Die Beleuchtungsverhältnisse schränken den visuellen Gestaltungsspielraum ebenso ein wie Umgebungsgeräusche die Erkennbarkeit von Warntönen.
- Je größer die Anzahl der unterschiedlichen Informationsklassen und Auffälligkeitsstufen ist, desto sorgfältiger muss die Aufmerksamkeitslenkung gestaltet werden.

Bei der Auswahl der Darstellungsattribute müssen ggf. auch kulturspezifische Besonderheiten berücksichtigt werden. Wie in Kap. 4 dargestellt wird, unterliegt vor allem das Farbverständnis in anderen Kulturkreisen anderen Gesetzmäßigkeiten.

3.4.7 Dialoge

Das Prinzip des Mensch-Maschine-Dialogs bestimmt den Ablauf, mit dem ein Bediener zur Abwicklung einer Arbeitsaufgabe – in einem oder mehreren Schritten – Daten eingibt und Rückmeldungen über die Verarbeitung dieser Daten erhält (DIN EN ISO 9241). Dieser Ablauf kann in verschiedenen **Dialogformen** stattfinden. Je nach Interpretation und Definition können bis zu sieben Basiskonzepte für Dialogformen unterschieden werden (VDI 3850):

- *Frage-Antwort-Dialog*
Benutzer gibt Antwort auf Systemanfragen.
- *Formular-Dialog (auch Masken-Dialog genannt)*
Dialog besteht aus Formularen, die vom Benutzer auszufüllen sind.
- *Kommando-Dialog*
Der Benutzer verwendet einen vordefinierten Satz von Anweisungen zur Kommunikation mit dem System.
- *Natürlichsprachlicher Dialog*
Der Benutzer unterhält sich mit der Maschine in natürlicher Sprache als wäre sie ein Mensch.
- *Direkte Manipulation*
Aktion mit direkter Wirkung, z. B. Drag and Drop
- *Funktionstasten-Dialog*
Knöpfe mit permanenter Bedeutung zum Auslösen von Funktionen (Operationen)
- *Menü-Dialog*
Funktionen werden in Form eines Menüs strukturiert. Keine direkten Eingabe von Daten möglich.

Heutige grafische Benutzungsschnittstellen bestehen i. d. R. aus Kombinationen dieser Dialogformen. Beispielsweise kann ein Menü-Dialog für die Grobstrukturie-

rung der Handlungen sorgen, während Formular-Dialoge zur Einstellung von Parametern vorgesehen sind.

Neben den Dialogformen sind die **Dialogebenen** ein weiteres Merkmal des Dialogprinzips. Sie geben an, ob Dialoge die unmittelbare Aufgabenausführung betreffen (*Primärdialoge*) oder ob sie lediglich für situationsabhängige Tätigkeiten benötigt werden (*Sekundärdialoge*), um z. B. die Online-Hilfe abzufragen. Sekundärdialoge werden meist optional ausgeführt, sind von kurzer Bearbeitungsdauer und der Nutzer kann nach ihrer Beendigung im Primärdialog fortfahren (Zeidler und Zellner 1994).

Insofern werden in diesem Teilschritt Dialogformen und ihre entsprechenden Ebenen bestimmt. Zu Bestimmung der Dialogebene werden auf Basis der Navigationsstruktur diejenigen Knoten gekennzeichnet, die die zuvor beschriebenen Eigenschaften des Sekundärdialogs aufweisen, wie z. B. Hilfeinformationen oder Dokumentationstätigkeiten. Alle anderen Knoten werden dann als primär gekennzeichnet. Danach werden Dialogformen für die Navigationssteuerung bestimmt, wobei sich der Funktionstasten-Dialog und der Menü-Dialog für Maschinenbedien-
schnittstellen insbesondere als zweckmäßig erwiesen und durchgesetzt haben. Als weiteres sind Dialogformen für die Ein- und Ausgabe von Daten festzulegen. Hierfür bieten sich **Parametrieren** und **Auswählen aus Listen** für Dateneingabe sowie **Systemabfrage** für Datenausgabe gut an.

3.4.7.1 Dialogobjekte

Dialogobjekte sind die Bausteine eines Bediensystems; jeder Dialog wird aus Dialogobjekten aufgebaut. Mit Hilfe von Visualisierungstechniken können dann Darstellungsformen gewählt werden, damit sowohl eindeutig zwischen dem Primärdialog und den Sekundärdialogen als auch zwischen den einzelnen Ein- und Ausgaben-
dialogen differenziert werden kann. Zu diesem Zweck wird beschrieben,

- wie Informationseinheiten des Primärdialogs und der Sekundärdialoge auf dem Bildschirm geordnet bzw. strukturiert werden sollen,
- welche Fenstertypen für die zuvor ermittelten Dialogfunktionen benötigt werden und
- mit welchen Dialogmodi die Interaktionsfreiheiten zu begrenzen sind.

Die Visualisierung von Informationen und die Aktivierung von Aktionen erfolgen bei grafischen Bedienschnittstellen mit Hilfe sog. Dialogobjekte. Um die verschiedenen Typen von Dialogobjekten in ihrer Funktion besser einordnen zu können, lassen sie sich unterteilen in:

- **Toplevel-Dialogobjekte**
(Anwendungsfenster, Dialogfenster, Mitteilungsfenster etc.),
- **Basis-Dialogobjekte**
(Button, Eingabefelder, Aktionsknöpfe etc.) und
- **komplexe bzw. zusammengesetzte Dialogobjekte**
(Tabellen, Auswahlfelder etc.).

Auf Basis der Navigationsstruktur werden zusammengefasste Informationen in Top-level-Dialogobjekten – sog. Fenstern – zusammengefasst und dort strukturiert. Diese Fenster können dann nach dem Ordnungsprinzip als *überlappend* oder *nebeneinander liegend*, ähnlich einem Kachelmuster (disjunkt), definiert werden. Das vielfältige Spektrum, das die Fenstertechnik für Büro- und Heimanwendungen bietet, ist für Maschinenbedienschnittstellen einzuschränken. Aufgrund der Dynamik des zu steuernden und kontrollierenden Prozesses sind *überlappende Fenster* für Primärdialoge von Echtzeit-Anwendungen ungeeignet, da der Benutzer aus eigener Initiative evtl. verdeckte wichtigere Informationen in den Vordergrund holen müssen.

Als nächstes sind die Fenstertypen für die Dateneingabe zu bestimmen. Es kann zwischen Anwendungsfenstern, Dialogfenster und Mitteilungsfenster unterschieden werden.

Anwendungsfenster bleiben dauerhaft auf dem Bildschirm und gelten als Behälter anderer Fenstertypen. Demzufolge wird jedem Bedienbereich ein Anwendungsfenster zugeordnet.

Dialogfenster werden eingesetzt, wenn im Primär- oder Sekundärdialog Daten einzugeben sind. Damit möglichst wenig der darunter liegenden Fläche verdeckt wird, sollten sie kompakt sein und nur die wichtigsten Elemente enthalten. Typischerweise sind Dialogfenster nicht größenveränderlich, so dass bei ihnen entsprechende Schaltflächen zur Größenänderung entfallen.

Mitteilungsfenster sind spezifische Dialogfenster, die den Bediener über ein Ereignis informieren, auf das er reagieren muss. Sie sollten sparsam eingesetzt werden und nur auf sehr wichtige Dialoge bzw. Mitteilungen beschränkt bleiben, da ansonsten die Bedienung unnötig verlangsamt wird und vor allem die Aufmerksamkeit gegenüber den einzelnen Mitteilungen nachlässt.

Für die Dialog- und Mitteilungsfenster kann weiterhin ihr Modus definiert werden. Ein Fenster kann

- systemmodal, d. h. ohne Interaktionsmöglichkeit mit anderen Fenstern,
- anwendungsmodal, d. h. ohne die Möglichkeit eine andere Anwendung zu starten,
- oder nichtmodal, d. h. ohne Einschränkung, sein.

3.4.7.2 Fokus- und Zustandssteuerung

Mit der Beschreibung der **Fokus** - und **Zustandssteuerung** der Dialogobjekte wird die Dynamik der Bedienschnittstelle abgeschlossen. Hier werden die dynamischen Eigenschaften der Dialogobjekte einer Bedienschnittstelle, d. h. deren Zustände und die möglichen Interaktionen zum Einleiten der Zustandsübergänge definiert.

Der *Fokus* – auch als Eingabefokus oder Tastaturfokus bezeichnet – gibt das Ziel innerhalb einer Bedienoberfläche an, auf das sich alle fokusbezogenen Dateneingaben oder Aktionen auswirken. Durch Verschieben des Fokus zwischen Dialogobjekten wandert auch das Ziel mit, auf das sich nachfolgende Eingaben beziehen. Da Eingabeinstrumente wie die Maus für Produktionsbereiche in der Regel ungeeignet sind, erfolgt die Fokussteuerung meistens über Fokussteuerungstasten, wenn das

Gestaltungs-Dimensionen	Büro-Bereich	Industrie-Bereich
Tätigkeiten	<ul style="list-style-type: none"> - Generalisiert - Uneinheitlich in Inhalt und Reihenfolge - Multi Document Interface 	<ul style="list-style-type: none"> - Spezialisiert - Einheitlich in Inhalt und Reihenfolge - Single Document Interface
Logische Dialoggestaltung	<ul style="list-style-type: none"> - Nutzergesteuerter Dialog - Kaum modale Dialoge - Große Menütiefe 	<ul style="list-style-type: none"> - System- und nutzergesteuerter Dialog - Teilweise modale Dialoge - Geringe Menütiefe
Grafische Dialoggestaltung	<ul style="list-style-type: none"> - Undefinierte Layoutbereiche - Verdeckung von Informationen (Fenstertechnik) 	<ul style="list-style-type: none"> - Definierte Layoutbereiche - Keine Überdeckung von Informationen (Bildmasken-, Kacheltechnik)

Abb. 3.31 Industrie- vs. Büro-Bereich. (Krauss 2003)

Dialogpanel kein Touchscreen ist. Über diese Tasten kann der Fokus zwischen Dialogobjekten verschoben werden.

Durch die Fokussteuerung wechselt der Aktivitätszustand eines Dialogobjektes zwischen *fokussiert* und *nicht fokussiert*. Sobald ein Dialogobjekt fokussiert ist, bestehen abhängig von der Art des Dialogobjekts unterschiedlichste Interaktionsalternativen wie „Elemente markieren“, „Daten eingeben“, „Daten verwerfen“, „Daten in die Anwendung übernehmen“, „Dialog beenden“, „Dialog abbrechen“ etc. Bei der Fokussteuerung wird also bestimmt, welche Aktion bei der Betätigung des Dialogobjektes ausgelöst werden soll.

3.4.8 Statisches Bildschirmlayout

Die Gegenüberstellung von Büro- und Industriebereich (s. Abb. 3.31) zeigt die unterschiedlichen Anforderungen und Auswirkungen auf die Gestaltung des Bildschirmlayouts. In der Produktion können im Gegensatz zum Bürobereich Aktionen auch von der Maschine bzw. dem technischen System ausgehen. Aus diesem Grund spielt die Kontrolle über den Prozess eine wesentliche Rolle. Früher wurden Kommandodialoage an dieser Stelle eingesetzt, heute erfüllen Maskendialoge (s. Abb. 3.32) die gestellte Aufgabe am besten.

Unter WINDOWS wird demgegenüber ein freier Dialog eingesetzt, bei dem der Benutzer Objekte wie Fenster oder deren Inhalte beliebig manipulieren kann (z. B. überlappende Fenster). Der Vorteil von Masken gegenüber Fenstern ist die kontinuierliche Anzeige des Systemzustands. Als Masken werden zweidimensionale Anzeigen des Zustands von interaktiven Systemen bezeichnet, die die gesamte Bildschirmfläche belegen. Der Bildschirm kann in einzelne Bildbereiche aufgeteilt werden. Den jeweiligen Bildbereichen werden bestimmte eigene Funktionen und Aufgaben zugeordnet.

Dies erlaubt dem Nutzer eine bessere Orientierung bei dem Aufruf einer neuen Seite. Folgende Bildschirmbereiche sind typischerweise festzulegen:

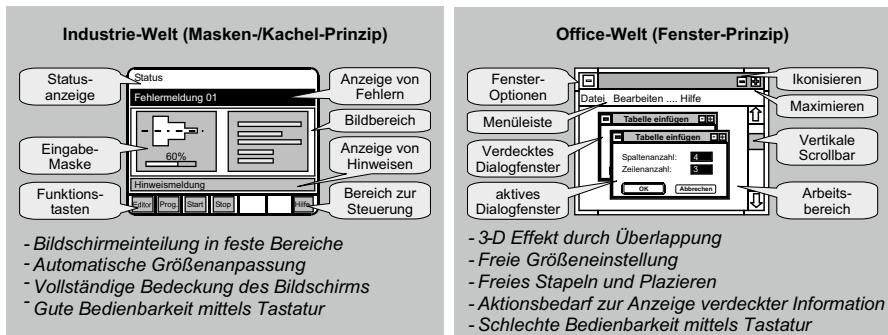


Abb. 3.32 Dialogformen im Industrie- und Bürobereich

- **Statusbereich**

Im Statusbereich erscheinen die asynchronen Maschinenmeldungen, sowie Datum und Uhrzeit.

- **Hinweisbereich**

Im Hinweisbereich erscheinen die kontextspezifischen Bedienhinweise.

- **Navigationsbereich**

Im Navigationsbereich werden die Bildschirmelemente dargestellt, die zur Navigation durch die Menüstruktur benötigt werden. Außerdem kann hier der Nutzer visualisiert werden, wo er sich in der Menüstruktur befindet und wohin er sich begeben kann.

- **Arbeitsbereich**

Der Arbeitsbereich ist im Allgemeinen der größte zusammenhängende Bereich und ist für die maskenspezifischen Informationen bestimmt.

Diese Bildschirmaufteilung ist auch für ein Plattformkonzept interessant, bei dem Bildschirme unterschiedlicher Größe eingesetzt werden. Abb. 3.33 zeigt ein beispielhaftes Bildschirmlayout für ein 18"-Touchscreensystem. Die gleiche Struktur ist auch in einem Konzept für ein 5,7"-System wieder zu finden (s. Abb. 3.34):

Ganz oben in dem Bereich der höchsten Aufmerksamkeit befindet sich der Meldungsbereich. Hier werden die Meldungen (Warn- bzw. Störmeldung mit der höchsten Priorität) der Maschine angezeigt. Dem Bereich ist ebenso die Statusübersicht über die gesamte Anlage (Linienübersicht) zugeordnet. Hier wird angezeigt, in welchem Status sich die anderen Maschinen der Linie befinden.

Darunter befindet sich ein Bereich, in dem die wesentlichen, für den Nutzer interessanten Stati von Prozess (z. B. akt. Geschwindigkeit) und Produkt angezeigt werden. Außerdem enthält der Status-Bereich allgemeine Informationen wie z. B. Maschinenname und Uhrzeit/Datum.

In dem Feld Bedienbereichsauswahl kann zwischen den in der Strukturgestaltung festgelegten Bedienbereichen umgeschaltet werden. Darunter befindet sich der Arbeitsbereich. Hier werden die aktuellen Arbeitsinhalte dargestellt.

Am rechten Bildrand befindet sich ein Bereich zur Auslösung von direkten Bedienfunktionen. Dieser Bereich enthält globale Aktionen wie z. B. Start/Stopp. Im

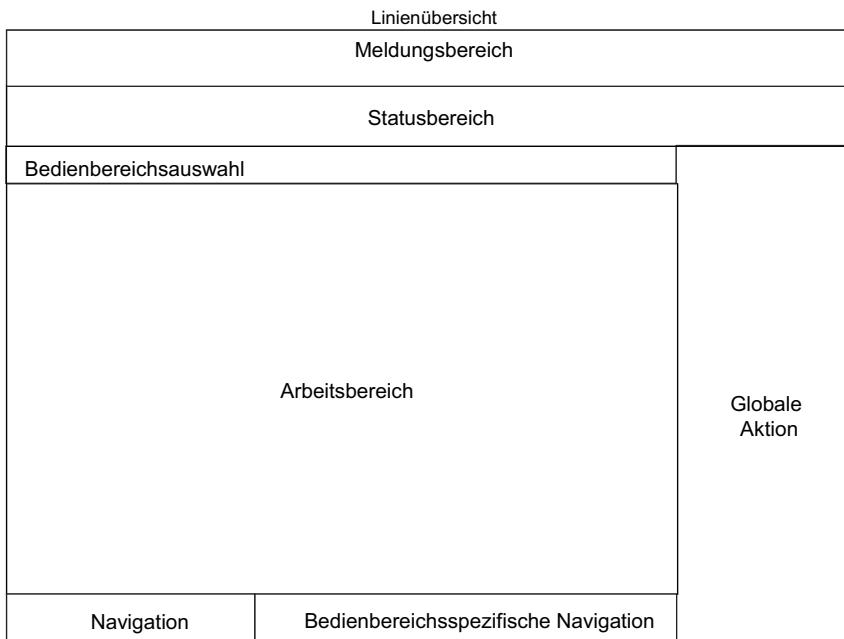


Abb. 3.33 Statisches Bildschirmlayout bei Systemen mit großen Bildschirmen

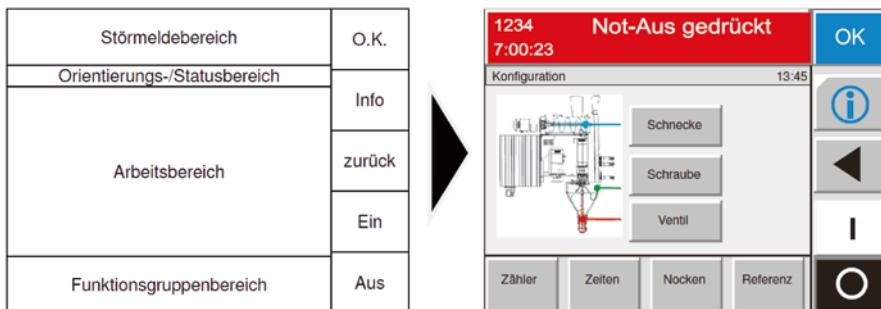


Abb. 3.34 Bildschirmlayout für kleine Bediensysteme

unteren Bildschirmbereich befindet sich der Navigationsbereich, der in übergreifende und bedienbereichsspezifische Navigation eingeteilt werden kann. Der übergreifende Teil des Navigationsbereichs ist in allen Kontexten gleich aufgebaut und ermöglicht den Zugriff auf die Navigationselemente der absoluten Navigation. Die Auswahl bzw. Umschaltung der Nutzergruppen und die Einstellung anderer Optionen (z. B. Sprachauswahl) wird mit Hilfe der hier angeordneten Interaktionselemente ausgeführt. Außerdem kann hier relativ in Richtung der Startseite navigiert werden.

Der bedienbereichsspezifische Navigationsbereich enthält die Elemente zur Navigation in tieferliegende Hierarchieebenen der Menüstruktur. Hier können Detailinformationen abgerufen werden.

In Abb. 3.34 ist dargestellt, wie der Bildschirm bei kleineren Systemen aufgebaut sein könnte. Leicht zu erkennen ist, dass der grundsätzliche Aufbau gleichartig zu den größeren Systemen ist, allerdings einige Bereiche verkleinert wurden. Dies ist im Hinblick auf den geringeren Umfang an Funktionalität der weniger komplexen Maschinen vertretbar. Dennoch ist für den Nutzer ein schnelles Identifizieren des „gewohnten“ Aufbaus möglich.

3.4.9 Dialogelemente

Aus Gründen der Konsistenz und der Erwartungskonformität sollten alle Masken aus einer begrenzten Anzahl von Basisdialogelementen aufgebaut sein. Hierzu können vordefinierte Basisdialogelemente verwendet werden, wie z. B. Eingabefelder, Tabellen, Radio-Buttons oder Slider, oder es können spezielle Basisdialogelemente selbst erstellt werden, wie z. B. Anzeigeelemente oder grafische Maschinenübersichten. Der Einsatz und die Auswahl dieser Elemente ist abhängig von der eingesetzten Hard- und Softwareplattform. So wird von den meisten Entwicklungsumgebungen bereits eine Bibliothek an Interaktionselementen zur Verfügung gestellt, aber der Entwurf eigener Interaktionselemente ist ebenso möglich.

Eine sinnvolle Vorgehensweise ist es, zunächst die vorhandenen Interaktionselemente auf die Eignung für den konkreten Einsatzzweck (z. B. Touchscreen) zu



Abb. 3.35 Anordnungsregeln nach DIN 33414

testen und eine Auswahl zu treffen. Weiterhin benötigte Interaktionselemente sind zu definieren und zu programmieren.

Ausgehend von der Bedienstruktur werden zu jeder Maske diejenigen Bedienfunktionen und Informationen ausgewählt, die zur Erledigung der jeweiligen Aufgabe notwendig sind. Die Anordnung der Bedienfunktionen und Informationen auf der Maske erfolgt auf Basis der ergonomischen Gestaltungsprinzipien (Konsistenz, Gruppierung, Kompatibilität) und unter Beachtung von Normen und Richtlinien.

3.4.10 Feingestaltung

Der Gestaltungsspielraum wird im Verlauf des Entwicklungsprozesses immer geringer und die Entscheidungen werden immer spezifischer. Während die Entwicklung bis zu diesem Punkt im Wesentlichen ein kreativer Prozess war, kann im Bereich der Feingestaltung sehr gut auf die umfangreichen Normen und Richtlinien verwiesen werden. An dieser Stelle sei eine kurze Zusammenfassung gegeben:

Bei der Grobkonzeption wird im plattformspezifischen Benutzungsmodell beschrieben, welche Inhalte welchen Knotenpunkten in der Navigationsstruktur zugeordnet sind und wie diese gruppiert werden. Das allein stellt aber noch keine ideale ergonomische Aufbereitung der Benutzungsoberfläche dar. Daher erfolgt im Rahmen der Feingestaltung die Optimierung des auf dem Benutzungsmodell basierenden Grobkonzepts hinsichtlich ergonomischer Details. Hierzu zählt beispielsweise die nutzerfreundliche Anordnung der Interaktionsobjekte auf der Benutzungsoberfläche nach den in Abb. 3.35 dargestellten Regeln zur ergonomischen Gestaltung (DIN 33414).

Bei der Feingestaltung des Maschinenbediensystems wird auch die Dynamik der Interaktion stärker berücksichtigt. Ausgehend von den Festlegungen des Grobkonzepts hinsichtlich Navigationsstruktur und Interaktionsobjekten wird das Verhalten

des Bediensystems auf Basis der ergonomischen Gestaltungsprinzipien definiert und unter Beachtung von Normen und Richtlinien überprüft und korrigiert.

Während in der Analyse- und Konzeptionsphase methodische Vorgehensweisen und deduktive Strategien eine große Bedeutung haben, muss im Rahmen der Feingestaltung auch die **hedonistische Qualität** des Interaktionssystems berücksichtigt werden (Burmester 2002). Diese beschreibt beispielsweise, wie innovativ, interessant und exklusiv der Nutzer das Bediensystem empfindet. Aufgrund der Subjektivität dieser Empfindungen sind auch bei der Feingestaltung die Nutzer einzubinden und hinsichtlich der genannten Aspekte zu befragen. Wertvolle Beiträge gerade zu diesem Aspekt leisten Designer, die in dieser Projektphase eingebunden werden sollten. Sie tragen wesentlich dazu bei, dass die Interaktion auch emotional wahrgenommene Qualitäten erhält.

Somit umfasst die Feingestaltungsphase einerseits die Herstellung der Gebrauchstauglichkeit des Maschinenbediensystems, andererseits aber auch die Erzeugung darüber hinausgehender subjektiver Qualitäten. Dies verdeutlicht die Bedeutung der manuellen Arbeitsschritte dieser Phase sowie der kontinuierlichen Evaluation. Die Strategie der automatischen Ableitung des lauffähigen Interaktionssystems aus den erhobenen Nutzeranforderungen, wie sie einige analytisch motivierte Ansätze verfolgen, kann höchstens gebrauchstaugliche, nie aber nutzerfreundliche Systeme herstellen.

3.4.10.1 Verwendung von Normen und Richtlinien

Bei der Entwicklung von Bediensystemen können Normen und Richtlinien hilfreich sein und Antworten auf spezielle Fragestellungen liefern. In diesen Normen und Richtlinien sind Erfahrungen in Form von Empfehlungen enthalten, die für die Gestaltung benutzerfreundlicher Systeme genutzt werden können und die zu einer Vereinheitlichung und Konsistenz der Anwendungen führen. Insbesondere im Usability-Umfeld beschreiben Normen und Richtlinien „State-of-the-Art“-Empfehlungen, die in der Regel heute noch nicht von Produkten am Markt eingehalten werden. In Kap. 6 sind die für diesen Bereich relevanten Normen und Richtlinien aufgeführt und erläutert.

Normen werden von Gremien bestehend aus Vertreter der Wissenschaft, der Industrie und der Abnehmer und Verwender von Industrieprodukten formuliert, die Berücksichtigungen dieser Normen ist zunächst freiwillig. Allerdings beschreiben Normen den Stand der Technik und werden als Verweis in Gesetzestexten und Sicherheitsvorschriften verwendet. Hierdurch werden sie rechtsverbindlich und wirken mithin normativ. Ein Verstoß gegen die Normen kann somit zu einer Verletzung von Rechtsvorschriften führen.

Während in technischen Normen genaue Vorschriften z. B. über Maße und Toleranzen enthalten sind und damit präzise Aussagen gemacht werden, sind ergonomische Normen meist sehr allgemein gehalten. Trotz der sonst üblichen Genauigkeit haben sich die ergonomischen Normen sowohl in Deutschland als auch international etabliert und sind heute weltweit anerkannt.

Die alleinige Berücksichtigung von Normen kann die Ergonomie eines Bediensystems nicht sicherstellen, ebenso wenig wie die Berücksichtigung weiterer Vorschriften und Richtlinien. Letztendlich ist die aus Sicht der Nutzer gelungene Anpassung an Ihre Aufgaben und die individuell vorliegenden Arbeitsverhältnisse der entscheidende Maßstab. Normen und Richtlinien liefern Hinweise und Anregungen, welche Gesichtspunkte beim Entwurf und der Gestaltung berücksichtigt werden sollen und welche Lösungen zur Verfügung stehen. Allerdings können sie zur Begründung von Haftungsansprüchen herangezogen werden, wenn sich der Gesetzgeber in Gesetzen und Verordnungen auf solche Normen bezieht.

3.4.10.2 Farbverwendung

Der Mensch empfindet Farben stark subjektiv. Grundsätzlich sollte Farbe sorgfältig und sparsam verwendet werden. Allgemein übliche Bedeutungen von Farben sollten bei der Informationscodierung auf dem Bildschirm erhalten bleiben. Es ist darauf zu achten, dass viele Menschen eine Farbsehschwäche aufweisen und somit eine Farbcodierung nicht ausschließlich verwendet werden sollte. Die im Folgenden dargestellten Regeln zur Farbverwendung beziehen sich auf den funktionalen und weniger auf den ästhetischen Aspekt.

Aus der Praxis: Welche Bedeutung die Umgebungsbedingungen auf die Farbauswahl haben, zeigt das folgende Beispiel:

Im Rahmen einer Nutzeranalyse wurde eine Produktionshalle besucht, die mit Natriumdampflampen beleuchtet wurde. Auffällig war hier die im Vergleich zu anderen Unternehmen sehr niedrige Produktivität verursacht durch eine hohe Anzahl an störungsbedingten Stillstandzeiten. Es stellte sich heraus, dass die gelben Warnhinweise an der Steuerung von den Maschinenbedienern aufgrund der Leuchtkarakteristik der Hallenbeleuchtung (hoher Gelbanteil) nur schlecht erkennbar waren. Dies führte zwangsläufig zu einer fehlenden Reaktion auf die Warnungen und damit zu erhöhten Maschinenstillstandszeiten.

Leitsätze zur Farbverwendung:

- **Anzahl der Farben**

Der normal farbsichtige Mensch kann bei ungünstigen Lichtverhältnissen und sequentieller Betrachtung nur eingeschränkt Farben unterscheiden. Deshalb sollten in industriellen Anwendungen maximal sechs Bunntöne + schwarz + weiß zur farblichen Bedeutungscodierung von Informationen verwendet werden. Beispiele sind rot, gelb, grün, blau, cyan, magenta. Außerdem ist zu beachten, dass die Umgebungsbeleuchtung die Wahrnehmbarkeit von Farben beeinflussen kann.

Tab. 3.4 Eignung von Farbkontrasten

Zeichenfarbe	Hintergrundfarbe							
	Schwarz	Weiß	Magenta	Blau	Cyan	Grün	Gelb	Rot
Schwarz	++	0	–	++	+	++	0	
Weiß	+	+	+	k. A.	k. A.	k. A.	+	
Magenta	0	+	–	0	0	++	–	
Blau	–	++	–	+	+	++	–	
Cyan	++	–	0	++	k. A.	–	0	
Grün	++	–	0	+	k. A.	–	0	
Gelb	++	k. A.	0	++	k. A.	k. A.	0	
Rot	0	++	–	–	+	+	+	

- **Farb- und Leuchtdichthekontrast**

Alle zum Kodieren eingesetzten Farben sollten hohen Farb- und Leuchtdichthekontrast bieten, um Erkennbarkeit zu gewährleisten. Sicherheitsbezogene Farben sollten hell, gesättigt und kontraststark sein. Für nicht vorrangige Informationen dürfen die verwendeten Farben dunkel und ungesättigt sein.

- **Absolute und relative Farbcodierung**

Es wird die absolute und die relative Farbcodierung unterschieden. Bei der absoluten Farbcodierung ist die Bedeutung der Information für sich alleine verständlich (z. B. Rot für Gefahr). Im Gegensatz dazu ergibt sich bei relativer Farbcodierung die Bedeutung der Codierung erst durch den Bezug zum Kontext (z. B. Kontrastabsenkung bei nicht anwählbaren Funktionen).

- Farbsättigung darf nicht zur absoluten Codierung verwendet werden
- Relative Codierung erfolgt über Helligkeit und Farbsättigung

- **Helligkeitsstufen**

Jeder Button sollte in maximal zwei Helligkeitsstufen verwendet werden.

- **Eignung von Farbkontrasten**

Für Kontraste sollten nur zwei aufeinander abgestimmte Farben für Zeichen (Vordergrund) und Zeichenhintergrund verwendet werden (s. Tab. 3.4). Bei mehrfarbiger Darstellung ist die Gefahr der Verwechslung umso geringer, je weniger Farben verwendet werden.

- **Bedeutung von Farben**

Durch die gleiche Verwendung von Farbe kann, auch über verschiedene Informationskategorien hinweg, Zugehörigkeit ausgedrückt werden. Welche Bedeutung den einzelnen Farben zugeordnet ist, ist Tab. 3.5 zu entnehmen. Die Farben Rot, Gelb, und Grün sind zwingend den in Tab. 3.5 genannten Bedeutungen zugeordnet (s. auch IEC 73 und DIN VDE 0199). Dabei darf Gelb auch als Orange oder Bernstein ausgeführt werden (IEC 73/3.1). Weist man der Farbe Blau eine bestimmte Bedeutung zu, so sollte dies durchgängig durch das gesamte System geschehen. Für die übrigen Farben ist keine systemübergreifende Verwendung notwendig, es dürfen jedoch keine Verwechslungsmöglichkeiten entstehen.

Tab. 3.5 Bedeutung von Farben

Herkunft	Bereich	Farbe	Blau	Grün	Gelb	Rot	Schwarz	Weiß	Allg. Information
DIN IEC 73	Leuchtmelder	Spezielle Information	Sicherheit	Vorsicht	Warnung oder Vorsicht	Alarm oder Gefahr	–	Allgemeine Information	
DIN VDE 0199	Sicherheit von Personen und Umwelt	Spezielle Information, Vorschrift	Sicherheit	Vorsicht	Gefahr		Allgemeine Information		
DIN VDE 0199	Prozesszustand	Spezielle Information	Normal	Anormal	Not		Allgemeine Information		
DIN IEC 16336	Zustand der Einrichtung	Spez. Bedeutung	Normal	Anormal	Fehlerhaft	Keine spez. Bedeutungen zugewiesen			
VDI 3850 DIN EN 60073	Prozesszustand	Zustand der ein Handeln erfordert	Normaler Zustand	Anormaler Zustand, Warnung eines bevorstehenden kritischen Zustands	Kritischer Zustand	Freie Bedeutung (allerdings unterschiedlich zu anderen Farben)			
Notwendige Bedienreaktivität		Eingreifen spez. Tätigkeit	Kein Handlungsbedarf	Beobachten und/oder Eingreifen zum Vorbeugen von Gefahr	Dringender Handlungsbedarf, sofortiges Reagieren	Überwachen			
DIN EN 60073	Beispiel	Anzeige einer vorgeschriebenen Tätigkeit	Anzeige eines sicheren Zustandes, sicheres Vorgehen	Störung, fehlerhafter Zustand, dauerhaftes oder vorübergehendes Risiko	Gefahr oder Befehl	Allgemeine Information			
DIN EN 60204	Drucktaster-Bedienteile	Bei einem Zustand betätigen, der eine zwingende Handlung erfordert	Betätigen, um übliche Zustände einzuleiten	Bei einem normalen Zustand betätigen	Bei gefährbringendem Zustand oder im Notfall betätigen	Allgemeine Einleitung von Funktionen			
DIN 4844	Sicherheitszeichen	Gebotszeichen, Hinweise	Gefährlosigkeit, Erste Hilfe	Vorsicht! mögliche Gefahr	Halt! Verbot	Allgemeine Information			

Bildliche Darstellung	Meldungskodierung	Kurzzeichen	Blinkfrequenz	Bedeutung Beispiel
	Dunkel	0	—	Ein zu meldender Betriebszustand steht nicht an. Prozeßvariable im Gutzustand, Anlage nicht gestört.
	Dauerlicht	1	—	Ein zu meldender Betriebszustand steht an. Prozeßvariable im Fehlzustand, Anlage gestört.
	Langsames Blinken	f_1	0,5 Hz	Übergang vom Fehl- in den Gutzustand.
	Blinken	f_2	1 Hz	Übergang vom Gut- in den Fehlzustand.
	Schnelles Blinken	f_3	2 Hz	Rückmeldung eines befohlenen Überganges. Eingeschalteter Stellantrieb läuft.

Abb. 3.36 Blinken

Aus der Praxis: Ein Unternehmen hat als „Hausfarbe“ gelb, sämtliche Maschinen sind dementsprechend in diesem Farbton lackiert. Auch auf dem Bildschirm soll gelb als Designelement eingesetzt werden. Damit verbietet sich diese Farbe allerdings für die Visualisierung von Warnungen. Eine alternative Darstellung könnte in diesem Fall z. B. die Aufblendung von Fenstern mit orangefarbenen Zeichen auf schwarzem Hintergrund sein.

Da das beschriebene Szenario nicht ungewöhnlich ist, sollte generell eine frühzeitige Abstimmung zwischen dem Designkonzept eines Unternehmens und den notwendigen Anforderungen der Informationsvisualisierung erfolgen.

3.4.10.3 Blinken

Eine weitere Möglichkeit neben der Farbcodierung ist das Blinken. Das Blinken kann gezielt zur Aufmerksamkeitssteuerung oder für ein Feedback vom Prozess eingesetzt werden. Blinken erschwert allerdings die Lesbarkeit von Schrift, weshalb Texte möglichst nicht blinkend dargestellt werden sollten. Abbildung 3.36 stellt Leitsätze aus der Prozesstechnik zum Einsatz von Blinken dar.

3.4.10.4 Schriften

Leitsätze

- Grundsätzlich ist seriflose Schrift zu verwenden.
- Proportionalschrift (Schrift mit konstantem Zeichenabstand) ist für fortlaufenden Text sowie für Uhrzeit und Datum am besten geeignet. Eine geeignete Schriftart ist z. B. Arial.
- Rasterschrift (Schrift mit konstantem Mittenabstand der Zeichen) ist für Anlagenkennzeichnungen und insbesondere bei Vergleich übereinander angeordneter Werte günstig. Eine geeignete Schriftart ist z. B. Courier.
- Zur Codierung von Information sollten maximal
 - 2 Schriftarten (Arial, Courier)
 - 2 Verzerrungen (normal, kursiv)
 - 3 Schriftgrade, die deutlich unterscheidbar sind
 - 2 Bildfette (normal, fett)
 - 3 Positionen (normal, hochgestellt, tiefgestellt)
 verwendet werden. Daraus ergibt sich eine Vielzahl an Möglichkeiten, und nur durch einen sparsamen Umgang mit Codierungsmöglichkeiten kann der Nutzer den Überblick behalten.
- Unterstreichung sollte zur Informationscodierung nicht verwendet werden, da hierdurch die Lesbarkeit reduziert wird.
- Fortlaufender Text wird schneller und mit weniger Fehlern erfasst, wenn Groß-/ Kleinschreibung angewandt wird. Texte ausschließlich in Großschrift sind schlecht lesbar, lediglich kurze und verbreitete Begriffe werden in Großbuchstaben besser erkannt (STOP, EIN, AUS).

3.4.10.5 Linien und Schraffuren

Linien können zur Gestaltung von Pfeilen und von Rahmen für Gruppierungen eingesetzt werden. Bei der Verwendung von Linien zur Informationsdarstellung sollten nicht mehr als drei Linienbreiten und vier Linienarten (durchgezogen, strichpunktiert, gestrichelt, usw.) verwendet werden. Bei Bildschirmanzeigen muss auf die durch das Raster bestimmte untere Darstellungsgrenze geachtet werden.

Schraffuren sollten zur Informationscodierung nicht verwendet werden wegen der damit verbundenen Interferenzen mit Zeichen im Vordergrund und wegen des Moiré-Effekts (Beim Betrachten einer schraffierten Fläche entsteht der Eindruck, als würde sie sich bewegen und vibrieren).

3.4.10.6 Bildzeichen

Bildzeichen werden eingesetzt, um Beschriftungen aller Art vorzunehmen und können außerdem eingesetzt werden, um Maschinenfunktionen zu visualisieren. Grundsätzlich lassen sich Bildzeichen in grafische Symbole, Piktogramme und

Abb. 3.37 Beispiele grafischer Symbole



Icons einteilen. Zur Visualisierung von Maschinenkomponenten können Maschinengrafiken eingesetzt werden (Charwat 1994).

Grafische Symbole

Grundsätzlich steht das Symbol für ein Bild oder ein Wort, das mehr aussagt, als auf den ersten Blick zu erkennen ist. Es steht für eine Sache oder einen Inhalt.

So steht beispielsweise das Posthorn-Symbol für das Unternehmen „Post“ und die damit verbundenen Dienstleistungen. Wie auch für viele andere Symbole typisch, wurde es aus einem realen Abbild durch zunehmende Abstraktion entwickelt (Abb. 3.37).

- Ein grafisches Symbol muss einfach sein, um das Wahrnehmen und Reproduzieren zu erleichtern.
- Es muss deutlich unterscheidbar von allen anderen Symbolen sein, mit denen es zusammen angewendet wird.
- Die Bedeutung soll leicht zu erlernen oder aber selbsterklärend sein.
- Für Buchstaben, Ziffern und mathematische Zeichen als einfache Symbolelemente soll eine einfache Schriftform angewendet werden.
- Die Anwendung von Farbe soll ein zusätzliches Element, nicht ein grundsätzlich informatives Element für die Bedeutung des Symbols sein.
- Das grafische Symbol soll innerhalb der z. B. in Richtlinien definierten Grundfigur entworfen werden.

Piktogramme

Piktogramme oder auch Sachbilder sind Bildzeichen, die reale Gegenstände oder beobachtbare Tätigkeiten (wie z. B. das Ausüben einer best. Sportart) als stilisierte Abbilder wiedergeben (Abb. 3.38).

Piktogramme sind für Personen, die das Abgebildete erkennen, selbsterklärend. Dadurch sind sie unabhängig von verbaler Sprache und demzufolge bleiben sie leichter und länger im Gedächtnis haften.

- Die Darstellung soll beim Betrachter eine möglichst starke Assoziation hervorrufen.
- Die Darstellung soll möglichst einfach sein. Dazu kann das Stilmittel der Abstraktion verwendet werden. Details können vernachlässigt werden.

Abb. 3.38 Beispiele von Piktogrammen

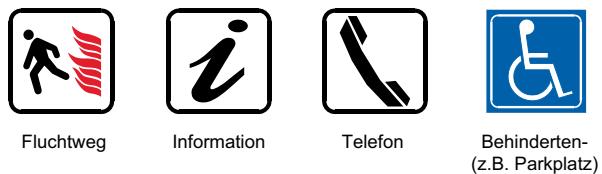
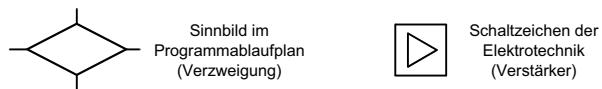


Abb. 3.39 Beispiele von Icons



- Beobachtbare Tätigkeiten sollen möglichst einen statisch instabilen Zustand wiedergeben. Dieser erweckt beim Betrachter den Eindruck einer Momentaufnahme der Bewegung.
- Jedes Piktogramm soll sich in möglichst vielen Merkmalen von anderen Piktogrammen unterscheiden.
- Die Anzahl an Piktogrammen in einem Alphabet ist möglichst gering zu halten. Bei einer großen Anzahl kann es zu Verwechslungen kommen und die Anzahl an Unbekannten steigt.
- Kann ein Begriff nicht eindeutig dargestellt werden, ist die verbale kurze Information dem Piktogramm vorzuziehen (z. B. STOP).

Icons/Ideogramme

Icons bzw. Ideogramme sind Zeichen, die für einen ganzen Begriff stehen, also nicht so elementar wie Buchstaben oder Ziffern sind (Abb. 3.39). Sie erhalten ihren Sinn (daher auch Sinnbilder), indem einer bestimmten geometrischen Figur eine bestimmte Bedeutung fest zugeordnet wird. Als weiteres Kennzeichen gilt, dass Icons abstrakter Natur sind. Es existiert weder bildliche Ähnlichkeit noch gibt es einen realen Bezug zwischen Icon und seiner Bedeutung, weshalb seine Interpretation erlernt werden muss.

In der Mensch-Maschine-Interaktion werden Icons verwendet, um abstrakte Sachverhalte mit einem einzigen Zeichen kenntlich zu machen. Dabei werden sie gelegentlich mit Piktogrammen kombiniert. Ist dem Betrachter die Bedeutung eines Icons geläufig, so erfasst er dieses schneller als die entsprechende Kette aus Schrift-Zeichen.

- Die Objekte im Icon sollen so realistisch und detailgetreu wie erforderlich gezeichnet werden. Dabei können die Objekte symbolisch für einen Begriff stehen. Folgende Stile, die konsistent beibehalten werden sollten, stehen zur Auswahl: Fotografischer Realismus, vereinfachte Zeichnung, Karikatur, Umriss, Silhouette.
- Die Objekte sind erkennbar zu zeichnen. Dazu soll der charakteristische Standpunkt für das Objekt beachtet werden, wichtige, aber kleine Komponenten eines

Objektes sollen übertrieben dargestellt werden. Tiefeneffekte (Fluchtenlinien, Schatten) – falls verwendet – sollen konsequent angewendet werden.

- Die Entwürfe für Icons sollen sich an der Technologie des Bildschirms orientieren (Pixeldarstellung): Gezackte Kanten und Linien vermeiden (Stufen- oder Treppenbildung bei Bögen und kleinen schiefen Winkeln), Diagonale Linien breiter zeichnen.
- Bei der Anordnung der Objekte soll auf Ausgewogenheit, Symmetrie und Stabilität geachtet werden.

Gestaltungsprinzipien von Bildzeichen

- Bildzeichen sind so zu gestalten, dass sie nach Möglichkeit der Alltagswelt des Nutzers entsprechen. Durch die Vertrautheit verbessern sich Lernen, Behalten und Erinnern.
- Die Gestaltung ist auf den angestrebten Anwendungsbereich (Darstellung auf Schildern, Tasten, Bildschirmoberflächen usw.) und deren Umgebungsbedingungen (Büro- bzw. Fabrikumgebung), sowie auf die Lesebedingungen (Entfernung des Beobachters, Beleuchtungssituation, usw.) abzustimmen.
- Bildzeichen, die in einem gemeinsamen Benutzungskontext stehen, müssen ein durchgängiges Erscheinungsbild aufweisen; sie sollten so genannte Bildzeichenfamilien bilden.
- Neue Bildzeichen können zur Beschreibung von Funktionszusammenhängen durch Kombination bestehender Bildzeichen bzw. aus Teilen derselben erstellt werden.
- Bildzeichen sind in ihrer Darstellung auf das Wesentliche zu reduzieren.
- Bei internationaler Verwendung ist auf die Verständlichkeit sowie auch Missverständlichkeit in den jeweiligen Nutzer- bzw. Kulturreihen zu achten.

Bildzeichenfamilien

Bildzeichenfamilien basieren auf der Kombination elementarer Bildelemente zur Codierung der Information (s. Abb. 3.40):

Bei einem internationalen Einsatz von Bildzeichen muss die kulturelle Bedeutung der Symbole abgeklärt werden. Insbesondere in den asiatischen Ländern, in denen Bildzeichen als Sprachzeichen gelten, werden Bildzeichen zum Teil falsch interpretiert oder erst gar nicht erkannt. Auf die kulturspezifische Gestaltung wird explizit in Kap. 4 eingegangen.

Bei der Bewertung von Bildzeichen wird zwischen objektiver Unterscheidbarkeit und subjektiver Erkennbarkeit differenziert. Die objektive Unterscheidbarkeit kann mit Hilfe der Hamming-Distanz ausgewertet werden. Die Hamming-Distanz liefert einen Kennwert für die Ähnlichkeit zweier Bildzeichen und kann somit zur Erkennung einer möglichen Verwechslungsgefahr herangezogen werden (Hamming 1987).

Die subjektive Erkennbarkeit steht für die Erkennung der Bedeutung eines Bildzeichens durch einen realen Nutzer und kann deshalb nur in Anwendertests ermittelt werden.

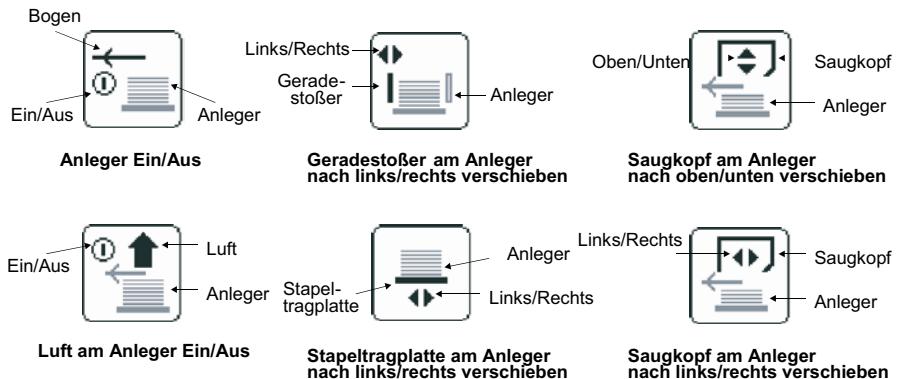


Abb. 3.40 Beispiel einer Bildzeichenfamilie

Maschinengrafiken

Maschinengrafiken als Abbild der realen Maschine werden eingesetzt, um dem Nutzer einerseits einen visuellen Überblick über das zu bedienende System zu bieten und andererseits die ortsortorientierte Bedienung zu ermöglichen. Orientiert sich die Navigation mit Hilfe der Funktionsgruppen eher an den Handlungen des Nutzers, so kann er über die Maschinengrafik Informationen komponentenbezogen abrufen (Abb. 3.41).

Grundsätzlich sollten Maschinengrafiken eher schematisch als bildhaft dargestellt werden. Der Einsatz von Fotografien hilft dem Nutzer eher bei der Identifikation bestimmter Detailkomponenten. Schematische, perspektivische 3D-Grafiken mit wenig Farbanteil haben sich als geeignete Mittel zur Visualisierung herausgestellt.

Wichtig beim Einsatz einer Maschinengrafik ist es, Komponenten, die an der spezifischen Maschine nicht zu finden sind, auch nicht darzustellen. D. h. es ist eine maschinenspezifische Darstellung erforderlich. Bedienbare Komponenten werden durch einen Rahmen hervorgehoben. Die Farbwahl des Rahmens sollte einen möglichst großen Kontrast zur Farbgebung der Maschinengrafik bilden. Über die Farbgebung kann zusätzlich der Störstatus der entsprechenden Maschinenkomponente codiert werden. In diesem Fall dürfen die Signalfarben Rot, Gelb, Grün nur zur Störmelde-Codierung verwendet werden.

3.4.10.7 Abkürzungen

Grundsätzliche Regeln

- Innerhalb desselben Anwendungsbereichs sollte für einen bestimmten Begriff nur **eine** Abkürzung festgelegt werden. Eine Abkürzung sollte in demselben Anwendungsbereich nur eine Bedeutung haben.
- Eine Abkürzung, die nicht wie ein selbständiges Wort gebraucht wird (Akronym), gilt für alle Flexionsformen (Abwandlungsformen) der Langform.



Abb. 3.41 Beispiel für eine bedienbare Maschinengrafik

- Bei Abkürzungen sollten Groß- oder Kleinschreibung nicht dazu dienen, Bedeutungsunterschiede auszudrücken.
- Fachsprachliche Benennungen und Namen sind in der Regel ohne Punkt abzukürzen. Allgemeinsprachliche Wörter, die in Fachtexten verwendet werden, sind nach den Regeln des DUDEN abzukürzen. Für besondere fachsprachliche Belege kann hiervon abgewichen werden.

Kürzungstechnische Regeln

- Der Anfangsbuchstabe einer Abkürzung sollte mit dem Anfangsbuchstaben der Langform des Begriffs übereinstimmen.
- Die Zeichenfolge einer Abkürzung sollte von der Zeichenfolge der Langform abgeleitet sein.

- Eine Abkürzung, die einem Wort oder einem Warenzeichen gleicht, sollte vermieden werden.
- Um von der Abkürzung leichter auf die Langform schließen zu können, sollten bei der Bildung von Abkürzungen worttypische Buchstabenkombinationen der Langform nach Möglichkeit erhalten bleiben.
- Bei Abkürzungen, die nach dem „Baukastenprinzip“ gebildet werden, beginnt jeder Baustein mit einem Großbuchstaben, dem ein oder mehrere Kleinbuchstaben folgen. Der Großbuchstabe bleibt in solchen Fällen auch im Innern der zusammengesetzten Abkürzung erhalten, um die Wortfuge zu verdeutlichen.

Sonstige Regeln

- Werden Abkürzungen verwendet, deren Bedeutung nicht als bekannt vorausgesetzt werden kann, sind sie zusammen mit der Langform einzuführen (z. B. in Bedienungsanleitungen oder Hilfesystemen).
- Die Verwendung von Sonderzeichen ist zugelassen (z. B. §).
- Innerhalb einer konkreten Anwendung sollten nicht mehrere Abkürzungstechniken parallel benutzt werden.
- In der Praxis bewährte Abkürzungen sollten auch dann benutzt werden, wenn sie von der gewählten Abkürzungstechnik abweichen, z. B. **CAD** für Computer-Aided-Design.
- Desgleichen sind Abweichungen von der gewählten Abkürzungstechnik erforderlich, wenn die Abkürzungen anderen zwingenden Vorschriften unterliegen (z. B. Einheiten gemäß SI-System).

Beispiele zur Anwendung der Regeln

- Abbrechkürzungen:

Eine Abbrechkürzung besteht aus mindestens zwei Buchstaben, die mit dem Anfang der Langform eines Wortes übereinstimmen.

Akku Akkumulator

Diag	Diagnose
Prog	Programmieren

- Initialkürzungen:

Es wird der erste Buchstabe der Wort-Langform verwendet.

S	Sekunde
l	Länge

- Klammerkürzungen:

Eine Klammerkürzung besteht aus dem ersten sowie weiteren Buchstaben der Langform eines Begriffes.

Spdl	Spindel
Strg	Steuerung

- Mischkürzungen:

Mischkürzungen liegen vor, wenn nicht alle Teile einer mehrteiligen Langform mit der gleichen Abkürztechnik bearbeitet werden.

VDI-Richtl.	Verein Deutscher Ingenieure Richtlinie
6 % ig	sechsprozentig

- Abkürzungen nach dem Baukastenprinzip:

StrgEin Steuerung Ein

StrgZust Steuerungszustand

Falls die Abkürzungen der einzelnen Bausteine aus Großbuchstaben bestehen, sollte zur Verdeutlichung der Wortfuge ein Punkt oder ein Bindestrich verwendet werden.

WZ-NP Werkzeugnullpunkt

WS-NP Werkstücknullpunkt

- Abkürzungen unter Verwendung von SI-Einheiten:

v-Schneid Schneidgeschwindigkeit

v-Vorschub Vorschubgeschwindigkeit

n-max Maximaldrehzahl

3.4.11 Erstellung eines Style-Guides

Im Hinblick auf die Realisierung bietet es sich an, die Ergebnisse des Gestaltungsprozesses in einem zentralen Dokument, dem sog. Style-Guide zu dokumentieren. In diesem Regelwerk werden die wichtigsten Grundsätze zur ergonomischen Gestaltung einer benutzer- und aufgabengerechten sowie einheitlichen Bedienoberfläche zusammengestellt. Dabei werden die besonderen Gestaltungsmerkmale für das betrachtete Produkt und softwareergonomischen Details der Gestaltungsphase dokumentiert und für andere Softwareentwickler zur Verfügung gestellt. Ein wesentliches Ziel beim Einsatz von Style-Guides ist die Vereinheitlichung der Bedienoberfläche und die konsistente Verwendung von Oberflächenbausteinen. Je präziser ein Style-Guide formuliert ist, desto weniger Fragen bleiben offen und desto weniger Interpretationsspielraum bleibt dem Entwickler bei der Umsetzung der Bedienoberfläche. Aufgrund des geringeren Interpretationsspielraumes können Arbeits- bzw. Entwicklungsschritte sicherer und schnellerer durchgeführt werden.

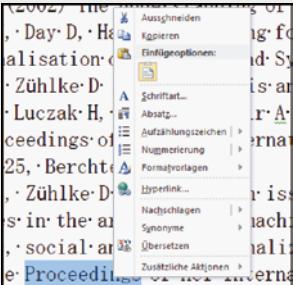
Die Wichtigkeit eines Style-Guides zeigt sich insbesondere, wenn nach bspw. einigen Monaten Veränderungen bzw. Ergänzungen an einer Bedienoberfläche durchgeführt werden müssen. Da die wichtigsten Regeln bzgl. der Bedienoberfläche im Style-Guide festgehalten wurden ist es, selbst für neue Mitarbeiter, einfach möglich Veränderungen bzw. Ergänzungen konsistent umzusetzen. Des Weiteren kann auf Basis eines Style-Guides die umgesetzte Bedienoberfläche auf Konformität überprüft und bewertet werden.

Der Aufwand zur Erstellung eines Style-Guides ist zwar initial mit Kosten verbunden, die allerdings aufgrund des Einsparungspotentials bei Änderung bzw. Erweiterung der Bedienoberfläche, des konsistenten Erscheinungsbildes und der gesteigerten Usability sich amortisieren können.

Style-Guides sind eine Ergänzung zu den sog. Design-Guides, in denen ausschließlich gestalterische Aspekte wie z. B. Farbkonzept, Buttongestaltung geregelt werden. Im Gegensatz zu Normen und Richtlinien, welche einen hohen Grad an

Pop-Up-Menü

Ein kontextsensitives Menü, das durch Druck auf die rechte Maustaste erscheint



Sinnvoll für:

- Erfahrene Benutzer als Shortcuts.
- Die Minimierung von Mausbewegungen durch Funktionsauswahl an der aktuellen Cursorposition.
- Die Anwahl der Eigenschaften von markierten Objekten.

Nicht sinnvoll für:

- Funktionen, die an keiner anderen Stelle im System angewählt werden können.
- Anfänger, die nicht wissen, dass in einem bestimmten Kontext Pop-Up-Menüs existieren.

Abb. 3.42 Beispiel für Vorgaben zur Auswahl von Dialogelementen

Allgemeingültigkeit aufweisen, vermitteln Style-Guides konkrete produktspezifische Vorgaben. Style-Guides sind damit als praxisnahe Leitfäden für die Softwareentwickler zu verstehen.

Ein Style-Guide besteht in der Regel aus den folgenden drei Hauptteilen:

- Teil I: Allgemeine Grundprinzipien zur ergonomischen Gestaltung
- Teil II: Grundaufbau des Bediensystems sowie Anweisungen zur Verwendung gestalterischer oder struktureller Elemente
- Teil III: Konkrete Beschreibung der Dialogelemente (Layout und Verhalten).

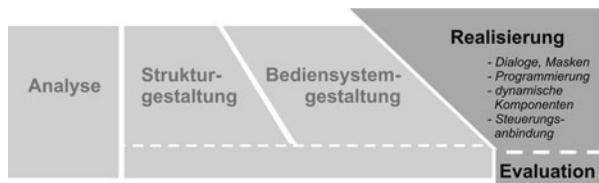
Die allgemeinen Grundprinzipien zur ergonomischen Gestaltung sind zwar fester Bestandteil jedes Style-Guides, der Schwerpunkt liegt jedoch auf den produktspezifischen Teilen II und III.

Teil II beinhaltet die Vorgaben zum statischen Bildschirmlayout, der Strukturierung von Inhalten sowie Vorgaben zum Einsatz von Dialogelementen. Diese Inhalte stellen die Grundlage für die Entwicklung des Systems dar und sollen zum Basiswissen des Entwicklers werden. Denn um eine gestalterische Problemstellung zu lösen, müssen dem Entwickler bereits alle möglichen Lösungen bekannt sein, damit er durch ihre Überprüfung im Style-Guide die geeignete auswählen kann.

So verdeutlicht das Beispiel aus Abb. 3.42, dass der Entwickler bereits wissen muss, dass Pop-Up-Menüs zum Aufrufen der Objekteigenschaften geeignet sind, um diese bei seiner Suche nach geeigneten Gestaltungsmöglichkeiten überhaupt zu berücksichtigen.

Hat der Entwickler sich nach den Vorgaben aus Teil II für ein Interaktionselement entschieden, kann er sich im **Teil III** über die genaue Spezifikation des Elementes informieren. Dieser Abschnitt des Style-Guides ist als Nachschlagewerk für den Entwickler zu verstehen, der die Objekteigenschaften wie Layout und Verhalten der Elemente genau beschreibt.

Abb. 3.43 Inhalte der Realisierungsphase



Auch bei Verwendung eines Style-Guides lassen sich heute bei der Realisierung zwei Probleme beobachten:

- Die Entwickler nutzen nur einen Teil der ihnen zur Verfügung stehenden Elemente. Alternativen kennen sie nicht und können sie nur durch Zufall oder intensive Recherche im Style-Guide kennen lernen.
- Die Entwickler setzen die Elemente falsch ein, da es zu umständlich ist, für einen neuen Sachverhalt geeignete neue Elemente zu finden.

Um eine konsistente Verwendung von Gestaltungselementen zu erzielen, muss ein Style-Guide sich daher an abstrakten Gestaltungszielen orientieren und dazu anleiten, wie diese in konkrete Gestaltungs- und Strukturmerkmale umgesetzt werden können. Mit Hilfe von Style-Guides kann diesen Problemen nur begegnet werden, wenn sie einfach zu nutzen sind und die Einhaltung der Regeln kontrollierbar ist. Wichtig ist daher, dass der Style-Guide nicht zu abstrakt, aber auch nicht zu detailliert ist, so dass das Dokument praktisch anwendbar ist. Durch konkrete und praxisnahe Vorgaben zeigt sich der Nutzen eines Style-Guides bei der Entwicklung aber recht schnell und unmittelbar in einer einheitlichen Gestaltung.

3.5 Realisierung

Aus Abb. 3.43 geht hervor, dass die Gestaltung und Realisierung des Maschinenbediensystems zu einem großen Teil parallel erfolgen. Im Rahmen der Gestaltung werden die Details des Interaktionssystems wie beispielsweise Interaktionssobjekte oder die Inhalte der Bildschirmmasken festgelegt. Deren Umsetzung in Form eines lauffähigen Programms erfolgt in der Realisierung. Hierbei werden auch die dynamischen Aspekte des Maschinenbediensystems umgesetzt und die Anbindung an die Maschinensteuerung vorgenommen.

Für die Realisierung werden i. d. R. gängige Entwicklungsumgebungen wie bspw. MICROSOFT VISUAL STUDIO herangezogen, mit denen die Entwickler auch die Maschinenfunktionalitäten umsetzen. Bisher bieten diese Werkzeuge keine Möglichkeit, Daten aus den frühen Phasen des Entwicklungsprozesses zu importieren. Daraus resultieren zwei alternative Vorgehensweisen, wie die erarbeiteten Entwicklungsergebnisse in ein lauffähiges Interaktionssystem überführt werden können:

Einerseits existiert ein nahezu unüberschaubares Angebot an Software-Entwicklungsgeräten, andererseits benötigen bestimmte Useware-Plattformen proprietäre Entwicklungswerkzeuge. In vielen Fällen ist davon auszugehen, dass deshalb ein Medienbruch zwischen Struktur- und Fein gestaltung erfolgen wird. Die Entwickler werden dabei den vom Nutzer evaluierten und in Form von Prototypen vorliegenden Useware-Entwurf mit dem jeweiligen Entwicklungswerkzeug realisieren müssen, ohne dabei die vorliegenden Daten importieren zu können.

Auch in der Realisierung werden **begleitende Nutzerevaluationen** durchgeführt, um so sicherzustellen, dass die Umsetzung der dynamischen Aspekte des Bediensystems den Nutzer-Anforderungen gerecht wird. Mit Abschluss der Entwicklungsarbeiten kann das realisierte Bediensystem direkt an der Maschine eingesetzt werden.

Zukünftig ist anzustreben, auch die Feingestaltung und Realisierung in gleicher Weise mit entsprechenden **modellbasierten Entwicklungswerkzeugen** zu unterstützen. Die bestehenden Ansätze hierzu sind aber aufgrund der Vielzahl von Programmiersprachen und Entwicklungssystemen immer auf bestimmte Domänen begrenzt (Myers 1996). Die zunehmende Verbreitung der XML-Technologie bringt allerdings bereits erste offene Konzepte hervor.

3.6 Evaluation

Phasenorientierte Prozessmodelle wie das Wasserfallmodell überprüfen das Entwicklungsergebnis im Rahmen einer abschließenden Validierung. Allerdings wird diese, dem Entwicklungsprozess nachgelagerte Evaluation – die häufig auch als Usability-Testing bezeichnet und angeboten wird – in den wenigsten Unternehmen wirklich durchgeführt (Zühlke 2010). Die Gründe liegen in der Projektorganisation, die nach der (funktionalen) Fertigstellung des Produkts keine Entwicklungressourcen mehr für die Evaluation selbst oder eventuelle Nacharbeiten bietet. Weitere Gründe können bspw. in der Geheimhaltung des Produktes vor Markteinführung oder manchmal auch im Desinteresse der Beteiligten liegen. Außerdem besteht die Gefahr, dass der Aufwand für Änderungen an der bereits fertig gestellten Useware zu hoch wird und die Evaluationsergebnisse erst beim nächsten Modellwechsel berücksichtigt werden können. Mittlerweile hat sich in den meisten Unternehmen auch die Einsicht durchgesetzt, dass Qualität im Prozess entstehen muss und nicht nach der Fertigstellung eines Produkts „hineingeprüft“ werden kann. Daher ist die **Evaluation als integrierte Querschnittsaufgabe** im gesamten Projektverlauf durchzuführen (Ortlieb und Holz auf der Heide 1993; Burmester und Komischke 1999).

Um die Nutzer frühzeitig in den Entwicklungsprozess einzubinden, müssen diese sich mit abstrakten Strukturen oder mit noch unvollständigen Prototypen beschäftigen. Der Mensch ist in dieser Hinsicht zwar sehr leistungsfähig, vor allem, wenn es um die Überprüfung und Beurteilung übergeordneter Strukturen und Zusammenhänge geht (Weidenmann 1999). Damit das Abstraktionsvermögen der Pro-

banden dennoch nicht überfordert wird, müssen übersichtliche Darstellungen der Strukturen aus dem Benutzungsmodell (automatisiert) generiert werden. Außerdem ist es wichtig, dass die Probanden den Einfluss ihrer Anregungen und Vorschläge auf das Projekt erkennen können. Auf Ablehnung bei den befragten Personen stoßen erfahrungsgemäß Prototypen, die schon zu perfekt ausformuliert und in allen Details umgesetzt sind. Sie erwecken den Anschein, dass keine Einflussmöglichkeiten mehr bestehen und die Befragung nur eine Alibi-Funktion hat.

Auch im Hinblick auf die Entscheidungsfindung ist das Prototyping wichtig. So können mit Hilfe der bei der Evaluation erhobenen Aussagen Entscheidungskriterien bewertet werden (Ortlieb 1993). Ebenso können sich Entscheidungsträger durch die Prototypen konkrete Vorstellungen von den zu erwartenden Endergebnissen machen und schneller die richtigen Entscheidungen treffen. Weiterhin leisten die Prototypen wertvolle Beiträge für Lieferantengespräche, da sie frühzeitig eine gute Vorstellung vom gewünschten System vermitteln.

Aber wie ist nun bei der projektbegleitenden Evaluation vorzugehen? Es gibt nicht die „beste“ Evaluationsmethode. Im Gegenteil ein Einsatz von Evaluationsmethoden ist immer aufgaben-, kontext- und projektspezifisch zu variieren bzw. anzupassen. In einer bestimmten Situation ist zur Erhebung einer bestimmten Information eine Evaluationsmethode mal mehr und mal weniger geeignet. Um einzelne Phasen der Prototypenentwicklung – im Sinne des parallel-iterativen Entwicklungsansatzes – evaluieren und dokumentieren zu können, muss eine flexible Methodenauswahl erfolgen, die eine prozessbegleitende Evaluation unterstützt.

3.6.1 *Evaluationsverfahren*

Eine Evaluation kann mehrere Ziele verfolgen (Holz auf der Heide 1993). Diese Ziele lassen sich wie folgt beschreiben.

- **Vergleichende Evaluation** (*Welches System ist besser?*)

Dazu werden mindestens zwei unterschiedliche Systeme miteinander verglichen. Die vergleichende Evaluation dient zur Bestimmung des besseren Systems für einen vorher festgelegten Anwendungsfall. Es können bei der Durchführung einer Arbeitsaufgabe die subjektive Zufriedenheit der Nutzer oder die objektive Leistungsmessung als Kriterien herangezogenen werden. Bei der Bearbeitung werden Fehler und Zeiten der Nutzer protokolliert und anschließend ausgewertet.

- **Bewertende Evaluation** (*Wie gut ist das System?*)

Hierbei sollen die Vorteile eines Systems ermittelt werden. Es wird genau untersucht, welche Ausprägung von Systemeigenschaften besonders gut gelungen ist. Kriterien hierfür sind die bei der Anforderungsanalyse erhobenen Systemeigenschaften selbst, die es zu überprüfen gilt. Als Kriterium können auch Heuristiken herangezogen werden (Nielsen 1994). Dieses Vorgehen erfolgt üblicherweise durch Experten. Aber auch Nutzer haben die Möglichkeiten ihr persönliches Urteil abzugeben. Außerdem kommt hier die interaktionsbezogene Analyse von

Nutzerfehlern zum Einsatz. Demnach ist das Erreichen von bestimmten vorher definierten Zielen zu überprüfen. Die dabei erzielten Ergebnisse werden vorher festgelegten Eigenschaften gegenüber gestellt und ermöglichen eine Aussage über die Güte des evaluierten Systems.

- **Analysierende Evaluation** (*Warum ist ein System schlecht?*)

Dieses Evaluationsziel wird besonders häufig während der Entwicklungsphase verfolgt. Es sollen Schwachpunkte herausgefunden werden, um diese dann, verbunden mit entsprechenden Vorschlägen für ein Redesign, verbessern zu können. Dazu müssen potentielle Nutzer die Möglichkeit bekommen, das System in einem festzulegenden Einsatzfall anzuwenden. Die Ergebnisse können z. B. durch Interviews oder Videoaufzeichnung festgehalten und nach Auswertung entsprechende Verbesserungspotentiale identifiziert werden.

Eine weitere Unterteilung der Evaluationsverfahren kann anhand der eingesetzten Methoden vorgenommen werden (Hom 2011). Es wird dabei zwischen Testmethoden (Testing), Kontrollmethoden (Inspection) und Erkundungsmethoden (Inquiry) differenziert.

Bei den **Testmethoden** erledigen repräsentative Nutzer typische Aufgaben mit dem System (oder Prototyp). Die Experten benutzen die Ergebnisse, um zu sehen, ob der Nutzer bestimmte Aufgaben mit dem System erfüllen kann. Bei den **Kontrollmethoden** versuchen die Experten etwas über Wünsche, Probleme, Anforderungen und Verständnis der Nutzer herauszubekommen, indem sie mit ihnen reden, sie bei der Arbeit beobachten, oder indem sie den Nutzern Fragen stellen (schriftlich oder mündlich). Die **Erkundungsmethoden** sind dadurch gekennzeichnet, dass Ergonomie-Experten (manchmal auch Programmierer und andere Experten) bestimmte Benutzbarkeitsprobleme der Schnittstelle selbst untersuchen (Hom 2011).

Evaluationsverfahren können aber auch nach dem Kriterium der Vorgehensweise und der dabei eingesetzten Methoden bei der Evaluation in die drei Gruppen unterteilt werden: formale, heuristische und empirische Evaluation (Shneiderman 2009).

Die **formale Evaluation** basiert auf dem GOMS-Modell (Card 1983). Dabei wird die Bedienung interaktiver Systeme in Ziele (Goals), Operationen (Operations), Methoden (Methods) und Selektionsregeln (Selection rules) unterteilt.

Bei der **heuristischen Evaluation** handelt es sich weniger um ein messbares Verfahren. Vielmehr ist es ein methodisches Vorgehen, um Probleme bei der Nutzung des Systems zu Identifizieren. Die Methodik liegt darin, Usability-Probleme mit Hilfe von Richtlinien zu erkennen. Grundlage für die Durchführbarkeit einer heuristischen Evaluation ist ein bestehender Prototyp. Je nach Entwicklungsstand des Prototyps spricht man von einer *kalten* Einschätzung, wenn das System nur auf dem Papier existiert; einer *warmen* Einschätzung, wenn nur eine Teilfunktion besteht; einer *heißen* Einschätzung, wenn es sich bei dem System um ein nahezu vollendetes Produkt handelt (Preim 1999).

Mit der **empirischen Evaluation** bezeichnet man die Tests von Endnutzern an einem Prototyp. Der Unterschied zur heuristischen Evaluation ist, dass der Test in einer späteren Phase der Entwicklung stattfindet. Der empirischen Evaluation ist somit oft eine heuristische vorausgegangen.

Im Folgenden werden einige wichtige Evaluationsmethoden vorgestellt. Dafür wird die Einteilung nach Shneiderman (Shneiderman 2009) verwendet, die Zielstellungen der Evaluation werden integriert (Hom 2011).

3.6.2 Evaluationsmethoden

3.6.2.1 Formale Evaluationsmethoden

An dieser Stelle soll das bereits erwähnte GOMS-Modell näher erläutert werden, ausgehend von den vier Elementen: Ziel, Operation, Methode und Selektionsregel.

Das Ziel (Goal) beschreibt, was der Nutzer erreichen will. Dies könnte z. B. das Bewegen eines Objektes sein. Um dieses Ziel erreichen zu können, muss der Nutzer verschiedene Operationen (Operations) durchführen. Diese Operationen sind also die elementaren Interaktionstechniken, zu denen die Bewegung des Cursors, die Betätigung der Mausbuttons und die Verwendung der Tastatur gehört. Die Methoden (Methods) setzen sich aus einzelnen Operationen zusammen, die aufeinander folgend ausgeführt werden. Auf das Beispiel bezogen besteht eine Methode aus der Bewegung der Maus und der Betätigung der Maustaste. Wenn nun mehrere Methoden zur Zielerreichung möglich sind, wird deren Auswahl durch die Selektionsregeln (Selection rules) bestimmt. Für die jeweiligen Alternativen können die Zeiten für jede einzelne Teilaktion (Operation) bestimmt werden, die zur Zielerreichung notwendig sind.

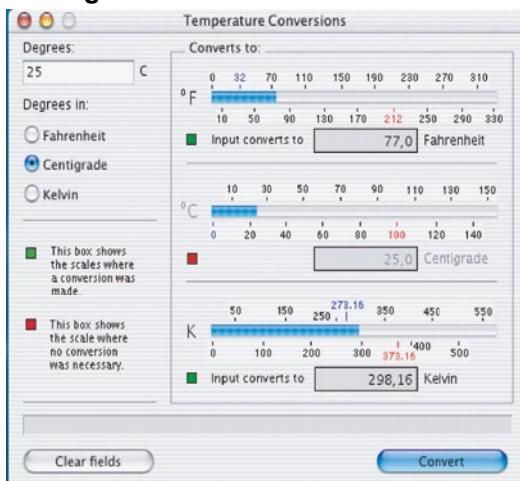
Daraus ergibt sich dann die jeweilige Gesamtzeit pro Alternative für alle zur Verfügung stehenden Möglichkeiten (Wege) der Handlungsdurchführung zur Zielerreichung (s. Abb. 3.44). Anhand dieser Zeiten kann das GOMS-Modell zur Effizienzanalyse verwendet werden (Card 1983; Preim 1999; Wandmacher 1993).

3.6.2.2 Heuristische Evaluationsmethoden

Eine **heuristische Evaluation** liegt vor, wenn ein Untersuchungsgegenstand von einem oder mehreren Evaluatoren stellvertretend für die Endnutzer beurteilt wird (Heuer 2003). Charakteristisch ist außerdem, dass heuristische Methoden überwiegend in frühen Phasen der Entwicklung eingesetzt werden.

Durch die **iterative heuristische Expertenevaluation** wird ein bestehender Untersuchungsgegenstand, auf der Basis von Heuristiken, auf mögliche Probleme für die Endnutzer untersucht. Ein beispielhaftes Ergebnis einer solchen Evaluation zeigt Tab. 3.6. Die Heuristiken ergeben sich aus Erfahrungswerten und Richtlinien, die sich für die entsprechende Art des Untersuchungsgegenstandes als wichtig und ergonomisch sinnvoll erwiesen haben. Diese heuristische Methode wird von mindestens einem Tester durchgeführt. Die Anzahl der gefundenen Usability-Probleme steigt nach Nielsen (Nielsen 1993) degressiv mit der Anzahl der Evaluatoren. Diesen Zusammenhang gibt Abb. 3.45 wieder.

Dialogfelder



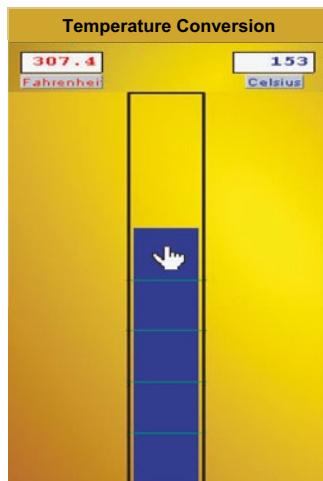
H+M+Z+Z+T = 4,85sec

Taste: (T = 0.3sec)
Zeit für das Drücken einer Taste auf der Tastatur.

Zeigen: (Z = 1,3sec)
Zeit zum Zeigen auf eine Position am Bildschirm.

Hand: (H = 0,5sec)
Zeit, um die Hand von der Tastatur oder GUI zu bewegen

Schieber



H+M+Z+Z+Z+Z+T = 7,45sec

Mentale Vorbereitung: (M = 1.45sec)
Zeit zur mentalen Vorbereitung auf den nächsten Schritt.

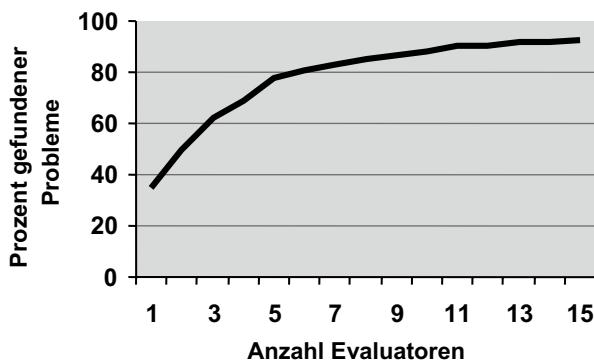
Reaktion: (R = x sec)
Zeit bis das System antwortet

Abb. 3.44 Beispiel für eine GOMS-Analyse – Anwendungsfall „Temperaturumrechnung“. (Röse und Ziegeler 2003)

Tab. 3.6 Beispielhafte Ergebnisse einer iterativen heuristischen Expertenevaluation im Projekt „MMI-Portal“. (DFN 2003)

Bereich	Problem oder Verbesserungsvorschlag
Portal	Links werden „gelöscht“, Kategorien hingegen „entfernt“. Für die Funktionen sollten einheitliche Begriffe verwendet werden
Portal	Es existiert eine Anzeige, in der bereits abgegebene Bewertungen zu einem Wert zusammengefasst sind. Handelt es sich hierbei um einen Mittelwert? Der User sollte diese Information auch bekommen können
CSCW	Woher weiß der User, welche Mitglieder er einladen kann?
Portal und CSCW	Der „Abbrechen“-Button sollte immer an derselben Stelle vorhanden sein. Bisher steht er bei einigen Eingabeformularen unten links und bei anderen rechts

Abb. 3.45 Verhältnis zwischen Anzahl Evaluatoren und dem Anteil gefundener Probleme. (Nielsen 1993)



Die iterative heuristische Expertenevaluation kann somit bei fünf Evaluatoren bereits ca. 80 % der Usability-Probleme aufdecken. Steigt die Anzahl über zehn, so sind nur noch marginale Verbesserungen zu erwarten. Der erste Schritt dieses Evaluationsverfahrens besteht darin, die Heuristiken festzulegen, nach denen das Produkt untersucht werden soll. In der Literatur existiert eine Vielzahl verschiedener Heuristiken. Nachfolgend werden die zehn Heuristiken von Nielsen (Nielsen 1994) dargestellt, da diese die allgemeinen Fragestellungen der Entwickler abdecken und allgemein bereits gute Erfahrungen mit ihnen gemacht wurden. Im Folgenden werden diese Heuristiken genannt und kurz erläutert.

1. Sichtbarkeit des Systemzustandes

Das System sollte zu jedem Zeitpunkt den Nutzer darüber informieren, wo er sich gerade befindet und wohin er als nächstes navigieren kann. Es müssen also Links und generelle Verknüpfungen deutlich markiert sein. Die Struktur muss an jeder Stelle klar ersichtlich sein.

2. Übereinstimmung zwischen System und der realen Welt

Das System sollte die Sprache des Anwenders sprechen; mit Worten, Phrasen und Konzepten, welche dem Nutzer vertraut sind. Es sollte auf die Verwendung von Systemorientierten Meldungen -insbesondere Codes- verzichtet werden. Die Informationen bzw. Meldungen müssen in natürlicher und logischer Reihenfolge angezeigt werden.

3. Nutzerkontrolle und Freiheit

Bei der Nutzung des Systems wird es immer wieder zu Fehlern von den Nutzern kommen. Für diesen Fall muss an jeder Stelle im System ein deutlich markierter ‚Notausgang‘ existieren, um unerwünschte Aktionen rückgängig machen zu können. Dies sollte mit kurzen Dialogen realisiert sein. Funktionen, wie ‚Rückgängig‘ und ‚Wiederholen‘ sollten von einer Software unterstützt werden. Wichtig ist auch ein genereller Link zum Ausgangspunkt der Navigation.

4. Konsistenz und Standards

Bezeichnungen für Inhalte, Situationen und Aktivitäten müssen konsistent sein. Ebenso sollten die Buttons konsistent bezüglich Gestalt und Platzierung sein. Eine Anwendung sollte gleiche Sachverhalte einheitlich darstellen.

5. Fehlerprävention

Ein umsichtiges Design, welches Fehlern vorbeugt, ist immer besser als eine gute Fehlermeldung. Deshalb sollten Mehrdeutigkeit und Unübersichtlichkeit vermieden werden sowie unwichtige Informationen verborgen bleiben. Es sollten Eingabeüberprüfungen in Dialogen und Formularen stattfinden.

6. Wieder erkennen ist besser als Erinnern

Objekte, Aktionen und Optionen sollten sichtbar sein. Es ist zu vermeiden, dass sich der Anwender an jede Dialogreihenfolge erinnern muss, um Instruktionen für das Benutzen des Systems erreichen zu können bzw. das System selbst bedienen zu können. Dialoge und Instruktionen sollten erreichbar sein, wann immer sie gebraucht werden.

7. Flexibilität und Effizienz der Nutzung

Beschleunigungen oder Abkürzungen (zum Beispiel Toolbars mit Icons oder Tastaturkürzel, wie Strg+C), werden vom unerfahrenen Anwender nicht gesehen und nicht gebraucht. Sie können jedoch häufig die Geschwindigkeit der Benutzung für erfahrene Anwender erhöhen. Sie sollten daher für die am häufigsten genutzten Funktionen angeboten werden.

8. Ästhetik und minimales Design

Dialoge mit unwichtigen oder ungewöhnlichen Informationen sollten vermieden werden. Jede zusätzliche Information in einem Dialog steht im Wettbewerb mit den relevanten Informationen und vermindert somit ihre relative Sichtbarkeit. Informationen sollten stets so kurz und gehaltvoll wie möglich sein, d. h. jegliche unnötige Information ist zu vermeiden.

9. Hilfe für den Anwender beim Erkennen, Diagnostizieren und Rückgängigmachen von Fehlern

Fehlermeldungen sollten in einfacher Sprache erfolgen. Es sollte auf Codes bzw. dem Anwender unverständliche Meldungen verzichtet werden. Fehlermeldungen sollten die Ursache des Problems präzise beschreiben und konstruktiv zeigen, wie das Problem zukünftig vermieden werden kann.

10. Hilfe und Dokumentation

In jedem Fall ist es anzustreben, dass ein System ohne Dokumentation verwendet werden kann. Oft sind Hilfe und Dokumentation trotzdem notwendig. In diesem Fall sollte jede Information leicht zu finden sein. Die Hilfe sollte auf den Kontext bzw. die Aufgabe, welche der Anwender gerade bewältigt, fokussiert sein. Es sollte eine konkrete Liste von Schritten ausgegeben werden, welche einen Lösungsweg für die Aufgabe beschreibt und den Anwender möglichst schnell zum Ziel führt. Links zur Hilfe sollten stets verfügbar sein.

Diese heuristische Evaluation ist meist nur der Einstieg für die weiteren Evaluationschritte. Dadurch können sich die Beteiligten der Evaluation ein umfassendes Bild von den bestehenden Funktionalitäten und Elementen des Produktes machen. Insbesondere für ein späteres Aufstellen von Task-Szenarios bringt diese Methode große Vorteile, denn durch die Szenarios kann überprüft werden, ob die von den Experten prognostizierten Probleme auch tatsächlich bei den Endnutzern auftreten.

Focus-Group-Research sind moderierte Diskussionsrunden, die zu einem bestimmten Thema geführt werden. Dabei werden die Teilnehmer mit den verschiedenen Sichtweisen, Ideen und Wahrnehmungen der anderen Gruppenteilnehmer konfrontiert und tauschen sich untereinander aus. Typisch für die Durchführung von Fokusgruppen ist es, dass sie in einer frühen Phase des Entwicklungsprozesses stattfinden. Ziel ist es, erste Konzepte durch repräsentative Nutzer zu evaluieren. Das Hauptunterscheidungsmerkmal zu anderen Techniken ist die simultane Zusammenarbeit von mindestens drei Nutzern oder drei Experten (Rubin 1994).

Fokusgruppen dienen so der qualitativen Datenerhebung bezüglich zentralen Ansprüchen und Nutzererwartungen. Sie eignen sich dann, wenn es gilt, Verhalten begründende Motivationen kennen zu lernen oder bestehende Kundenprobleme zu entdecken. Insgesamt kann eine Fokusgruppe aber nur eine Ergänzung zu quantitativen Datenerhebungen sein. Wenn sie vor der empirischen Datenerhebung durchgeführt wird, können die Dimensionen der Datenerhebung geklärt werden. Danach ist die Fokusgruppe geeignet, um offen gebliebene Bereiche abzudecken (Wandke 2003).

Das Untersuchungsobjekt einer Fokusgruppe kann durchaus zunächst aus einem Papier- oder Plastikmodell bestehen oder ein virtueller Prototyp am Computer sein. Es ist zu überprüfen, ob dem Untersuchungsobjekt ein akzeptables Konzept zugrunde liegt, bzw. an welchen Stellen es nicht akzeptabel ist und wie die Akzeptanz gesteigert werden kann. Der große Vorteil des Focus-Group-Research ist dessen Fähigkeit, durch die Urteile und Gefühle einiger weniger Testpersonen, die sich intensiv mit dem Untersuchungsobjekt befassen, einen Einblick in das Gedankenmodell der Endnutzer zu bekommen.

3.6.2.3 Empirische Evaluationsmethoden

Die **empirische Evaluation** ist im Gegensatz zur heuristischen Evaluation dadurch gekennzeichnet, dass hierbei die Beurteilung durch echte Endnutzer erfolgt (Heuer 2003; Shneiderman 2009). Der zugrunde liegende Untersuchungsgegenstand kann sich – ähnlich wie bei den heuristischen Evaluationen – in verschiedenen Stadien der Entwicklung befinden. Generell werden empirische Methoden aber in späteren Phasen der Entwicklung eingesetzt (Preim 1999).

Die **Thinking-Aloud-Methode** (Methode des lauten Denkens) ist laut Nielsen (Nielsen 1994) die wertvollste Usability-Engineering-Methode. Dabei werden die Testpersonen dazu aufgefordert, während des gesamten Tests laut zu denken. Ziel ist es, Schwierigkeiten des Nutzers zu identifizieren und nach Möglichkeit die Struktur seiner mentalen Modelle zum Handlungsvorgehen zu erkennen.

Bei der Thinking-Aloud-Methode arbeitet die Testperson in Gegenwart eines Evaluators oder Prüfers am System und wird ständig dazu aufgefordert auszusprechen, was ihr bei der Arbeit mit dem System in den Sinn kommt (Thimbleby 1990). Ziel ist es, die Vorstellungen, Gedanken und unbewussten Haltungen der Nutzer zum Vorschein zu bringen und Hinweise darüber zu erhalten, an welchen Stellen Probleme bei der Mensch-Maschine-Interaktion auftreten.

Thinking-Aloud-Methoden lassen sich unterscheiden in die Concurrent-Thinking-Aloud- und die Retrospective-Thinking-Aloud-Methode (Nielsen 1993). Bei der **Concurrent-Thinking-Aloud-Methode** werden die Testpersonen (Nutzer) gebeten, ihre Gedanken zu verbalisieren, während sie versuchen, eine Aufgabe zu lösen. Bei der **Retrospective-Thinking-Aloud-Methode** sollen die Testpersonen (Nutzer) erst nach Beantwortung einer Frage ihre Äußerungen tätigen. Bei der Retrospective-Testing-Methode wird der Test mitgeschnitten. Dieser Mitschnitt geht der Tester (Evaluator) mit der Testperson durch und beide besprechen gemeinsam die aufgetretenen Probleme. Der Vorteil dieser Methode ist es, dass der Tester die Aufnahme jederzeit anhalten und gezielt Fragen stellen kann. Dadurch dauert der Test meist doppelt so lange wie bei die Concurrent-Thinking-Aloud-Methode, was den Nachteil dieser Methode ausmacht. In einigen Untersuchungen werden beide Varianten der Thinking-Aloud-Methode miteinander verbunden. Dies geschieht indem die Testpersonen aufgefordert werden während des Lösen einer Aufgabe ihre Gedanken zu verbalisieren und zusätzlich nach der Bearbeitung der letzten Frage Gelegenheit haben, ihre Eindrücke zu schildern und Fragen zu stellen. Die Variation beider Verfahren wird gewählt, weil nicht nur Gedanken, die direkt mit der Lösung einer Aufgabe zusammenhängen erfasst werden können, sondern darüber hinaus auch Äußerungen der Zufriedenheit, Begeisterung und Frustration. Ein retrospektives gemeinsames Ansehen des Testes ermöglicht der Testperson eigenes Verhalten zu reflektieren und zu kommentieren.

Eine weitere empirische Evaluationsmethode ist die Durchführung von **Interviews**. Bei einem Interview werden Nutzern, die mit einem System vertraut sind oder die an einem System Testaufgaben erledigt haben, Fragen zu diesem System gestellt. So werden Erfahrungen durch den Befragten mitgeteilt. Ein Problem bei diesem Evaluationsverfahren ist es, dass viele Befragte – insbesondere im Werkstattumfeld – ihre Erfahrungen schlecht in Worte fassen können. Insgesamt gehen Details zwischen Test und Befragung verloren.

Eine Alternative zu Interviews stellen **Fragebögen** dar. Deren Vorteil liegt in einer schnelleren Auswertung, die oft auch automatisch durchgeführt werden kann. Es besteht bei diesem Verfahren darüber hinaus eine geringere Hemmschwelle für die Testpersonen. Ein Nachteil liegt darin, dass Fragen falsch verstanden werden können und durch die Formulierung der Frage das Ergebnis bzw. die Antwort stark beeinflusst werden kann. Die Absicht eines **Background-Fragebogens** ist es, Hintergrundinformationen über die Testpersonen zu erhalten. Diese Informationen sollen helfen, die Verhaltensweisen und auftretenden Probleme der Testpersonen besser zu verstehen. Es wird also nach den bisherigen Erfahrungen in Bezug auf das Aufgabengebiet gefragt. Typischerweise wird der Background-Fragebogen vor dem eigentlichen Test ausgefüllt (Rubin 1994).

Ziel des **Post-Test-Fragebogen** ist es, Informationen über die Stärken und Schwächen des Untersuchungsgegenstandes zu bekommen. Dabei sollen die Fragen im speziellen die Aspekte „Joy-of-use“ und „Ease-of-Use“ hinterfragen, deren Umsetzung ein wichtiges Ziel einer ergonomischen Produktentwicklung ist. Somit könnte sich der Post-Test-Fragebogen aus zwei Teilen zusammensetzen. Der erste Teil wäre dann ein semantisches Differential zur Erfassung des „Joy-of-use“, nach

Abb. 3.46 Semantisches Differenzial zur Erfassung des „Joy-of-use“. (Hassenzahl 2001)

Hervorragend	<input type="checkbox"/>	Extrem Schlecht				
Exklusiv	<input type="checkbox"/>	Standardmäßig				
Imponierend	<input type="checkbox"/>	Ausdruckslos				
Einmalig	<input type="checkbox"/>	Gewöhnlich				
Innovativ	<input type="checkbox"/>	Konservativ				
Interessant	<input type="checkbox"/>	Langweilig				

Abb. 3.47 Likert Skala zur Erfassung des „Ease-of-use“.
(DFN 2003)

	Volle Zustimmung	Zustimmung	Unentschieden	Ablehnung	Volle Ablehnung
Die Navigation war übersichtlich.	<input type="checkbox"/>				
Die Funktionen sind verständlich.	<input type="checkbox"/>				
Ich wusste stets wo ich mich im Portal befand.	<input type="checkbox"/>				
Das Feedbackverhalten war zufriedenstellend.	<input type="checkbox"/>				

Hasenzahl (Hassenzahl 2001). Allgemein handelt es sich dabei um Skalen, mit welchen die Testpersonen ihre Bewertung abgeben und ihre Meinung auf einer zweipoligen Skala mit semantischen Dipolen vermerken. Ein mögliches semantisches Differenzial zeigt Abb. 3.46.

Die ausgewählte Skala besteht wie oben dargestellt aus fünf Feldern. Diese Skala wird meist einer Sieben-Felder bzw. Drei-Felder-Skala vorgezogen, da sich in der Praxis die fünf Felder als einfach benutzbar erwiesen haben, denn durch die bekannte Assoziation mit Schulnoten ist eine Erfassung von Tendenzen problemlos möglich. Der zweite Teil des Post-Test-Fragebogens könnte dann aus einer Beurteilung des „Ease-of-Use“ bestehen. Hierzu können die Testpersonen mit verschiedenen Aussagen konfrontiert werden. Zu diesen Aussagen geben die Testpersonen den Grad ihrer Zustimmung oder Ablehnung in einer Likert-Skala an (Rubin 1994) (s. Abb. 3.47). In beiden Teilen des Post-Test-Fragebogens ist hinter jeder Frage Platz für einen Kommentar, um der Testperson die Möglichkeit zu geben, ihre Entscheidung zu begründen.

Mit **Observation** wird die Beobachtung der Testpersonen durch den Tester bezeichnet. Nach der Art der Durchführung dieser Methode unterscheidet man zwischen direkter Observation, wenn der Tester die Testperson bei der Bearbeitung des Tests beobachtet und indirekter Observation durch eine Software, die automatisch Logfiles oder Video- bzw. Audioprotokolle über die Aktionen der User erstellt (Preece 1994). Die direkte Observation erfolgt über schriftliche Dokumentation von Problemen und Besonderheiten. Mit Hilfe von Video-Software (z. B. CAMTASIA) kann eine indirekte Observation umgesetzt werden (s. Abb. 3.48). Als Ergebnis steht ein Problempunktkoll, in dem jeder vermerkten Testsituation die entsprechende Aufzeichnungssequenz der CAMTASIA-Software beigelegt ist.

Task-Szenarien beschreiben eine typische Folge von Aktionen, die bei der Arbeit mit dem Produkt zukünftig auftreten werden (Preece 1994). Sie sind mehr als eine bloße Aneinanderreihung von Aufgaben. In den Task-Szenarien wird dar-

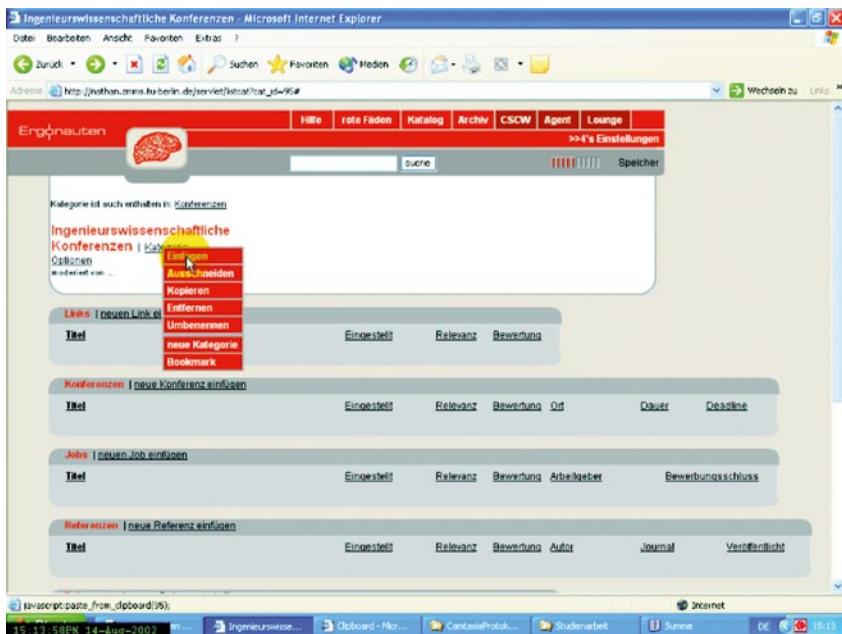


Abb. 3.48 Mit CAMTASIA erfasste Problematik der mangelnden Übersichtlichkeit bei einem Alpha-Prototypen. (DFN 2003)

Tab. 3.7 Beispiel einer Aufgabenliste beim Pilot-Test aus dem Projekt „MMI-Portal“. (DFN 2003)

Aufg.-Nr.	Aufgabe
Aufgabe 1	Link bewerten
Aufgabe 2	Kommentar erstellen
Aufgabe 3	Eigene Kategorie anlegen
Aufgabe 4	Dokument hinzufügen
Aufgabe 5	Link zur eigenen Homepage in der Kategorie „Personen“ hinzufügen
Aufgabe 6	Logout

über hinaus das Endresultat beschrieben, das die Testperson erreichen soll. Außerdem werden die Motive dargestellt, die hinter den Aufgaben stehen (Rubin 1994).

Vor der Durchführung der Testaufgaben bzw. Task-Szenarien mit den eigentlichen Testpersonen sollte unbedingt ein **Pilot-Test** durchgeführt werden. Dieser Pilot-Test dient dazu, versteckte Probleme aufzudecken, die bei der Bearbeitung der Aufgaben auftreten. Außerdem kann auf diese Weise eine bessere Einschätzung über die Dauer der Tests erfolgen (Rubin 1994b). Die Ergebnisse des Pilot-Tests sind Ausgangspunkt für die eine evtl. Veränderung der Testunterlagen und somit Bestandteil der Testvorbereitung. Für diesen Pilot-Test ist meist eine Anzahl von zwei Testpersonen ausreichend.

In Tab. 3.7 ist zu sehen, dass die Aufgaben nicht ausformuliert sind und lediglich eine allgemeine Handlungsanweisung geben. Die Aufgaben ergeben sich aus den Evaluationszielen der Entwickler und den aus der mittels Task-Analyse ermittelten

Abb. 3.49 Beispiel für Task-Szenarien. (DFN 2003)

Sie haben von einem Kollegen erfahren, dass die Tagung „Useware2002“ stattfindet. Außerdem hat er Ihnen die URL des MMI-Portals genannt und Sie wollen nun im Portal die genauen Konferenzdaten anschauen.

Aufgabe 1:

Suchen Sie im Katalog die Kategorie Konferenzen und lassen Sie sich die Informationen zur „Useware2002“ anzeigen.

Aufgabe 2:

Machen Sie ebenfalls von der Bewertungsmöglichkeit Gebrauch und geben Sie eine Bewertung für diese Konferenz ab.

wichtigsten Funktionen und Aufgaben (Röse 2003b). Wie die Aufgaben des Pilot-Tests zeigen, handelte es sich zu diesem Zeitpunkt noch nicht um ein Task-Szenario. Es ist lediglich eine Hintereinanderreihung von Test-Tasks, zwischen denen kein inhaltlicher Zusammenhang besteht (Nielsen 1993). Zur Steigerung der Motivation bei den Testpersonen werden die Aufgaben so umgestaltet, dass sich ein schlüssiges Task-Szenario ergibt. Abbildung 3.49 zeigt Aufgaben eines Testszenarios.

Als **Ergebnis** der empirischen Evaluation steht eine Tabelle (s. Tab. 3.8) mit allen wahrgenommenen Problemen, dem Vermerk, um welche Testperson es sich handelt und – wenn möglich – dem entsprechenden Ausschnitt aus der Video-Aufzeichnung zur Verfügung. In einer weiteren Spalte wird das Problem kategorisiert. Dabei wird zwischen funktionalen, begrifflichen und strukturelle Problemen unterschieden.

3.6.2.4 Entwicklungsbegleitende Evaluation

Ein Evaluationszyklus, der sich an die einzelnen Produktentwicklungsschritte anpasst und deren Ergebnisse und somit die Nutzersicht und -integration in das jeweilige Redesign des Produktes einfließen, nennt man entwicklungsbegleitende bzw. iterative Evaluation. Eine **iterative Evaluation** ist eng mit den einzelnen Produkt-

Tab. 3.8 Beispiel Ergebnisliste nach empirischer Evaluation aus dem Projekt „MMI-Portal“. (DFN 2003)

Nr.	Lokation	Problembeschreibung	Notizen und Beschlüsse
10	Home-Button	<ul style="list-style-type: none"> – Button „Home“ wird vermisst – Navigation ist nicht klar ersichtlich, Nutzer findet keinen „Home“ Button – Button „Katalog“ wurde nicht als „Home“ Button oder Funktion erkannt – Home-Button-Funktion wurde auf Logo „Brain“ vermutet – Home-Button wurde gesucht 	Beschluss 1: Schriftzug + Logo = Homebutton Beschluss 2: „Home“ = als Home wird die Hauptkatalogansicht festgelegt;
39	Drop-Down-Menue	– Optionen-Popup verschwindet zu schnell	Beschluss 3: Abgleich von Portal und CSCW, Anzeigedauer wie bei Windows-Kontextmenüs

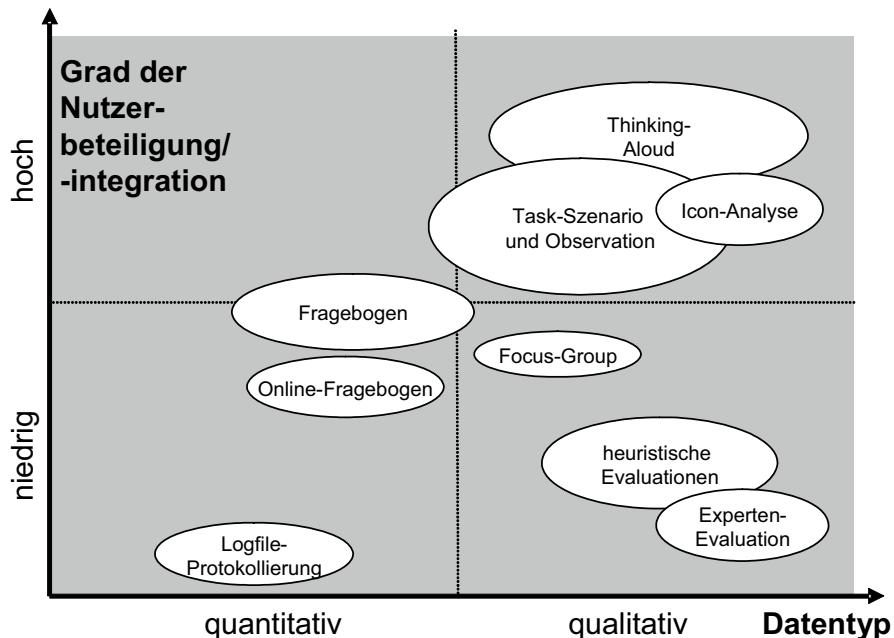


Abb. 3.50 Einordnung der Evaluationsmethoden. (DFN 2003)

entwicklungsphasen verknüpft und gleichzeitig Ausdruck einer hohen Nutzerintegration in die Produktentwicklung. Nur mit einem interdisziplinären Projektteam ist eine erfolgreiche Einbindung einer solchen iterativen Evaluation in den Produktentwicklungsprozess möglich. Die iterative Evaluation ist somit ein notwendiges Instrument für einen parallel-iterativen Entwicklungsprozess (Leuchter 2003). Die Evaluationsverfahren und -methoden werden dabei gezielt zur Ermittlung der qualitativen Güte eingesetzt, um den Entwicklern hinreichende Informationen für Redesigns und Releases zu liefern. Durch die iterative Evaluation ist es somit möglich, eine durchgehende Einbindung der Nutzer in die Produktentwicklung zu gewährleisten und einen ‚user-centered‘ Ansatz zu realisieren.

Wenn die Daten der Evaluation zügig in ein Produktentwicklerteam einfließen sollen und konkrete Aufgabenstellungen zum Redesign von der Evaluation erwartet werden, dann ist generell ein qualitatives Evaluationsverfahren zu bevorzugen. Abbildung 3.50 gibt einen Überblick zu den im DFN-Projekt ‚Useworld.net‘ (DFN) verwendeten Evaluationsmethoden. Es wurden fast alle hier vorgestellten Verfahren eingesetzt. Ebenfalls zeigt die Abbildung eine Einordnung der Methode hinsichtlich qualitativ oder quantitativ und den Grad der Nutzerbeteiligung.

Der überwiegende Anteil qualitativer Verfahren ist deutlich zu erkennen. Die dargestellte Größe der einzelnen Methoden soll den Aufwand widerspiegeln, sowohl zeitlich als auch kostenmäßig. Diese Einteilung ergibt sich aus der Praxiserfahrung in mehreren Projekten.

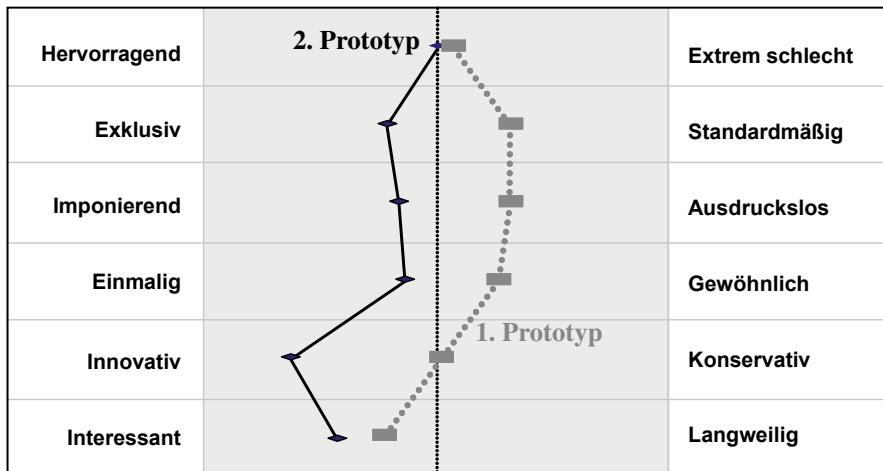


Abb. 3.51 Ergebnisse iterativer Evaluation von Prototypen am Beispiel des von den Nutzern wahrgenommenen „Joy-of-use“. (DFN 2003)

Die Schwierigkeit in der iterativen Evaluation besteht in der ständigen Weiterentwicklung und Veränderung der zu bewertenden Prototypen. Genau dies ist jedoch das Kennzeichen eines parallel-iterativen Entwicklungsansatzes. Damit sind häufig kurzfristige qualitative Evaluationen mit einer überschaubaren Anzahl von Testpersonen und einem vertretbaren Aufwand notwendig.

Die Vorteile einer iterativen Evaluation verdeutlicht Abb. 3.51. Durch die Bewertung einzelner Prototypenphasen ist ein deutlicher Fortschritt zu sehen und vorhandenes Verbesserungspotential erkennbar. Der Vorteil der iterativen Evaluation liegt ganz klar in der präventiven Wirkung. Durch die stetige Evaluation in einzelnen Entwicklungsphasen können teure Nachbesserungen des Produktes vermieden sowie kundenorientierte Anpassungen bereits in frühen Entwicklungsphasen eingebunden werden. Dabei ist darauf zu achten, dass die Evaluation nach Möglichkeit durch objektiv erfolgt. Dies meint, dass nicht der Entwickler selbst mit Kollegen eine Evaluation vornimmt (was für kleinere Aspekte durchaus vertretbar ist), sondern eine neutrale Evaluation mit potentiellen Nutzern durchgeführt wird (gemeint ist, nach Abschluss einer jeweiligen Entwicklungs – bzw. Prototypenphase).

3.6.2.5 Hinweise für die Durchführung

Bei der Durchführung einer Evaluation findet sich der Durchführende in verschiedenen Rollen wieder. Dies ist meist die Rolle des Evaluators, der Stärken und Schwächen eines Produktes aufzeigt; kann jedoch auch die Rolle des Beraters sein, der Hinweise zu Produktänderungen oder – Verbesserungen gibt oder die Rolle des Advokaten, der zu einer bestimmten Aussage Stellung nehmen muss und die Daten zur Untermauerung seiner Aussage benötigt. Auch denkbar ist die Rolle eines

Interpreters, der Evaluationsergebnisse interpretiert und Empfehlungen – im Sinne eines Agenten – ausspricht (Moser 2003). Bei der Ergebnisaufbereitung empirischer Daten sollte man sich stets diese unterschiedlichen Rollen ins Bewusstsein rufen und bei der Dateninterpretation und Formulierung von Empfehlungen darauf reflektieren. Dies ermöglicht eine umfassende Aufbereitung von Evaluationsdaten und die Zufriedenstellung unterschiedlichster Anforderungen an eine Evaluation.

Bei der Formulierung entsprechender Daten ist mit Bedacht vorzugehen. Allzu schnell sind Generalisierungen formuliert, die nur auf eine ganz spezifische Stichprobe zutreffen. Im Folgenden wird eine kurze Auflistung der häufigsten Fehler vorgestellt, die häufig bei der Durchführung einer Evaluation gemacht werden (Moser 2003).

1. *Einfach mal fragen*

Es erscheint trivial, Nutzer mittels Fragebogen oder Interview einfach einmal zu befragen. Dies ist jedoch nur möglich, wenn vorab genug Wissen über den Befragungsgegenstand zur Verfügung steht. Über den zur Diskussion stehenden Sachverhalt „nichts zu wissen“ dürfte andernfalls am Beginn und Ende einer Befragung stehen, da diese Methoden nur bedingt zur Aufklärung geeignet sind. So können z. B. geschlossene Fragen zum Sachverhalt nur mit vorher generierten Hypothesen gestellt werden. Dies gilt auch für Interviews und Beobachtungen. Die Zielrichtung des Methodeneinsatzes muss vorab bekannt sein, um eine seriöse Interpretation der ermittelten Daten vornehmen zu können.

2. *Fragebogen als Wundermittel*

Fast unabhängig von der Aufgabenstellung gibt es den so genannten Fragebogen-Reflex. Es herrscht die allgemeine Auffassung vor, dass ein Fragebogen mit geschlossenen Fragen leicht zu erstellen und auszuwerten sei. Einfach erstellt und verschickt bzw. Online gestellt, dann müsse man nur auf die Antworten warten. Probleme hierbei sind jedoch: meist relativ geringe Rücklaufquoten (meist um 20 %) und das ‚Kratzen an der Oberfläche‘ mit geschlossenen Fragen bzw. standardisierten Fragebögen. Die quantitative Datenauswertung ist zwar hilfreich zum Erstellen von Überblicksgrafiken etc., aber eine Interpretation der Ergebnisse ist stets mit Vorsicht zu genießen, da keine erläuternden Beschreibungen etc. zu den Antworten vorliegen. Letzteres ist jedoch ein wichtiges Indiz, um Fehlinterpretationen zu vermeiden. Dies ist auch mit ein Grund, warum in der Nutzeranalyse und – evaluation meist qualitative Methoden bevorzugt werden. Außerdem ist zur Erstellung und Auswertung geschlossener Fragebögen ein detailliertes Wissen zum Sachverhalt notwendig sowie die Generierung von entsprechenden Hypothesen. Ferner ist eine Beschränkung des Befragungsgegenstandes sowie der daraus resultierenden Aussagefähigkeit der erhobenen Daten zu beachten.

3. *Formulierung von Fragen*

Die Kunst der Frageformulierung will gelernt sein. Allzu oft schleichen sich Suggestivfragen ein, z. B. „Finden Sie, dass die Informationsdarstellung des Hauptmenüs zu verwirrend ist?“. Eine Frage sollte nicht gleich die Bewertung vorweg nehmen, wie im Beispiel dargestellt. Der gleiche Sachverhalt lässt sich auch neutral erfragen, z. B. „Bitte beurteilen Sie die Informationsdarstellung

des Hauptmenüs!“. Einfachheit der Frageformulierung und Verständlichkeit sind oberstes Gebot. Ansonsten sollte auch auf die Transparenz des Frageanliegens für den Befragten geachtet werden.

4. *Kausalitäten und Korrelationen*

Berechnete Korrelationen sind zwar Aussagen über Zusammenhänge, jedoch nicht automatisch über vorhandene Kausalitäten. In der Fachliteratur oft zitiert: „.... dass die Geburtenrate immer dann am höchsten ist, wenn die Störche aus ihrem Winterquartier nach Hause kommen“ (Moser 2003). Heutzutage würde niemand auf die Idee kommen, dass damit der Kausalschluss gegeben ist: Störche bringen die Kinder. Hier sind gleiche Randfaktoren für unterschiedliche Phänomene zuständig, die ihre Effekte zufällig zum gleichen Zeitpunkt zeigen. Eine Korrelation von Störfällen zu einer ganz bestimmten Tageszeit kann durchaus mehrere Ursachen haben, z. B.: Stromschwankungen, Schichtwechsel, Überhitzung nach 8 Stunden, etc. Meist wirken auch mehrere Faktoren zusammen.

5. *Macht der Zahlen*

Nichts wirkt so überzeugend wie Zahlen und Prozente. „80 % der Befragten lehnen neue Informationstechnologien ab.“ Eine Aussage, die auf den ersten Blick recht überzeugend klingt. Angenommen, diese 80 % beziehen sich auf 4 von 5 befragten Nutzern; und weiterhin angenommen, diese 5 Befragten von 1000 Betroffenen sind aus einer Abteilung, ... So oder so ähnlich sind viele Untersuchungen im Detail analysierbar. Von generalisierten Aussagen ist daher Abstand zu nehmen und die reale Anzahl der evaluierten Nutzer sollte stets angegeben werden sowie eine Anmerkung zur Repräsentativität der Stichprobe (Studenten vs. Maschinenbediener).

6. *Jäger und Sammler*

Bei der qualitativen Forschung kommt schnell eine kaum zu bewältigende Datensammlung zusammen. Insbesondere Anfänger unterliegen einem so genannten ‚Jäger – und Sammler-Trieb‘. Eine Beherrschung der Datenmengen und eine sinnvolle Datenauswertung ist jedoch nur möglich, wenn vorab geklärt wurde: welche Informationen für das Projekt benötigt werden, mit welcher Methode die Informationen erhoben und wie die ermittelten Daten ausgewertet werden.

7. *Dabei sein ist nicht alles*

Eine Konzentration auf die Aussagen, die einem speziell bei Interviews als erster Eindruck zurück geblieben sind, bildet meist den Kern einer Analyse. Es werden entsprechende Aussagen verglichen und ad hoc-Hypothesen aufgestellt. Damit werden jedoch viele Potenziale, die insbesondere im Interviewmaterial stecken, verschenkt. Neue und interessante Ansätze können nur gefunden werden, wenn die Informationen systematisch, nach vorher festgelegten Hypothesen bzw. Schemata ausgewertet werden. Dabei auffallende Unterschiede bieten meist interessante Ansätze für Innovationen. Eine Anwesenheit des Studienleiters oder Verantwortlichen für die Datenauswertung und – interpretation bei allen Interviews ist somit nicht der Garant zur Findung innovativer Ideen und ersetzt auch nicht die systematische Datenauswertung.

8. Generalisierung und Verallgemeinerung

Ziel der meisten Untersuchungen ist es, allgemeingültige Aussagen zu finden. Vor zu schnellen Verallgemeinerungen oder Generalisierungen ist jedoch zu warnen. Insbesondere Evaluationen im industriellen Umfeld sind auf ganz spezifische Produkte oder ganz bestimmte Produktions- bzw. Fertigungsverfahren bezogen. Dies sind wichtige Einflussfaktoren im Rahmen der ergonomischen Produktgestaltung. Auch die Randfaktoren der Arbeitsgestaltung spielen eine wichtige Rolle und weisen eine große Varianz auf. Im Kontext der industriell-empirischen Evaluation im Produktentwicklungsprozess sollten die Aussagen immer den spezifischen Einsatz bzw. Verwendungszweck der Produkte und die daraus für die Nutzer resultierenden Einschränkungen berücksichtigen.

9. Epische Breite von Berichten

In der Kürze und Einfachheit der Ergebnisdarstellung liegt die Stärke. Evaluationsberichte sollten nicht den Anspruch haben, einen Nobelpreis für Literatur gewinnen zu wollen; hier kommt es nicht auf die epische Breite der Darstellungen an. Eine an der Entwicklungsaufgabe bzw. konkreten Fragestellung orientierte Aufbereitung der Ergebnisse, eine sachliche Darstellung der Faktenlage und die Bezugnahme auf Originaldaten sind wichtig. So sollten z. B. lieber Originalzitate der Nutzer verwendet werden, statt allgemeiner zusammenfassender Aussagen. Wichtig ist es, dass der Leser des Evaluationsberichtes einen möglichst „plastischen Eindruck“ von den Nutzungsproblemen erhält. Zusätzliche Kurzzusammenfassungen mit den wichtigsten Highlights der Probleme sollten ebenfalls angeboten werden. Hierin sollten nach Möglichkeit nur ganz dringende Probleme angesprochen werden, die z. B. bei Prototypenevaluation unbedingt notwendige Produktänderungen betreffen oder die zum Erfolg der Serienreife des Produktes notwendig sind.

10. Der Computer wird es schon richten

Zu verlockend ist die Vorstellung, dass alle Daten nur in den Computer eingegeben werden müssen und entsprechende Programme die Datenanalyse vornehmen und dann fertige Ergebnisse und Interpretationen ausgeben. Heutige Systeme helfen jedoch nur bei der Aufbereitung des Datenmaterials. Allein die Eingabe des falschen Datenformats, die Nutzung einer falschen Berechnungsmethode etc. kann entscheidenden Einfluss auf die Ergebnisse haben. Für diese Dinge bleibt jedoch der Mensch zuständig. Statistische Auswertungsprogramme und computerbasierte Auswertungen sind somit immer nur so gut, wie der Dateninput und die Fachkenntnis des Durchführenden bzw. auswertenden es erlauben.

Dies sind nur einige Hinweise zur Durchführung und Auswertung von Evaluationen, speziell der entwicklungsbegleitenden Evaluation. Zusammenfassend bleibt anzumerken, dass zu einer erfolgreichen Evaluation eine gute Vorbereitung, professionelle Durchführung und konzentrierte Auswertung gehören. Dies ist in der Regel garantiert, wenn Usability-Experten in das Entwicklungsteam integriert sind, egal ob firmeninterne oder -externe.

Kapitel 4

Useware-Systeme für internationale Märkte

In den letzten Jahren ist ein deutlicher Trend in Richtung einer zunehmenden Globalisierung zu erkennen. Hiermit eng verbunden ist der Anstieg der Exporte deutscher Maschinenbauhersteller. Ausgehend vom Customizing als Service für alle Kunden, ist durch die Globalisierung eine immer stärkere Notwendigkeit gegeben, die Produkte auch auf internationale Kundenbedürfnisse anzupassen. Dies ist eine Herausforderung, denn Export in unterschiedliche Länder heißt auch Export in unterschiedliche Kulturen.

Wie eine Umfrage unter 250 Unternehmen des Maschinen- und Anlagenbaus belegte, entwickeln nur 11 %¹ der befragten Unternehmen landesspezifische Bediensystemversionen, die meisten Firmen belassen es hingegen bei der englischen (19 %) oder landesspezifischen (30 %) Übersetzung.

Aus der Praxis: Ein international tätiger Textilmaschinenhersteller verkaufte einen großen Teil seiner Produkte nach Thailand. Mit steigenden Löhnen wurde die Produktion und damit auch die Maschinen allerdings mehr und mehr nach China verlagert. Im Gegensatz zu den gut ausgebildeten und durchaus englisch sprechenden Thailändern verstanden die chinesischen Nutzer nahezu kein Englisch. Nachdem die Maschinenbedienung zu immer mehr Problemen führte, entschloss sich der Hersteller, eine chinesische Übersetzung bei einem in Deutschland ansässigen Übersetzungsbüro in Auftrag zu geben. Die Ergebnisse waren ernüchternd. Viele Maschinenbediener konnten mit der chinesischen Bedienoberfläche nichts anfangen. Daraufhin beauftragte das Unternehmen ein erfahrenes Beratungsinstitut, welches in Zusammenarbeit mit einem chinesischen Ingenieur und Kundenvertretern das gesamte Bediensystem überarbeitete. Das daraus entstandene System wurde dann vom Kunden sofort angenommen und die Stillstandszeiten sanken drastisch.

¹ Kritisch betrachtet kommt dieser recht hohe Wert dadurch zustande, dass die meisten Firmen bereits kleine Abwandlungen ihres deutschen Systems als landesspezifische Gestaltung ausgeben. Ein realistischer Wert dürfte wesentlich niedriger liegen.

Um erfolgreiche Exportprodukte anbieten zu können, die den Bedürfnissen der Nutzer der jeweiligen Exportkultur entsprechen, müssen kulturelle Anpassungen vorgenommen werden. Die Nutzersysteme müssen auf ihren lokalen Einsatzort hin optimiert werden, sie müssen einem lokalisierten Gestaltungsprozess unterzogen werden. Dazu zählen zum einen die technische Lokalisierung (z. B. instabile Stromversorgung oder klimatisch bedingte Arbeitsraumbesonderheiten u. a.) als auch die kulturelle Lokalisierung (Berücksichtigung anderer Denk- und Handlungsmodelle oder formeller Interaktionsformen bei den Nutzern der technischen Geräte u. a.). Diese Anpassung an lokale (regionale) Besonderheiten ist die Grundvoraussetzung, um kundengerechte Produkte anbieten zu können. Erste kulturvergleichende Untersuchungen im Bereich der Produktionstechnik haben Nutzerunterschiede zwischen der europäischen und der asiatisch-pazifischen Region aufgezeigt und diesen Ansatz bestätigt (INTOPS 1, Zühlke 1998). Folgeuntersuchungen bestätigten aber auch die Vermutung, dass selbst im europäischen Kulturraum deutliche Unterschiede festzustellen sind. In Zeiten der Globalisierung muss somit nutzerorientiertes Design auch Aspekte des kulturorientierten Designs berücksichtigen.

Das Ziel dieses Kapitels ist es, Elemente des interkulturellen Produktdesigns für Mensch-Maschine-Systeme aufzuzeigen. Dazu werden einzelne Aspekte der kulturorientierten Gestaltung vorgestellt. Es wird dabei eingegangen auf:

- Kultur als Nutzervariable (s. Berufserfahrung o. ä.)
- Interkulturelles Design und Engineering von Useware
- Empirische Ergebnisse einer Nutzeranalyse zur Gestaltung lokalisierter chinesischer Mensch-Maschine-Systeme.

Es wird der Fragestellung nachgegangen werden, wie im Rahmen der Produktentwicklung neben einem „user-oriented“ Design, ein „culture-oriented“ Design umgesetzt werden kann.

4.1 Interkulturelles Design

Schon seit Jahren geht man davon aus, dass es langfristig eine globale Kommunikation geben wird und mit ihr verbunden einen gemeinsamen Sprachraum. Dies ist heute schon im Bereich der Computerwissenschaften mit den neuen Begrifflichkeiten des Computerzeitalters zu bemerken. Z. B. Email, Upgrade, Server, etc. sind Begriffe, die international weitgehend einheitlich für gleiche Funktionalitäten benutzt werden: eine so genannte globale Technosprache. Aber dennoch bleibt die **kulturelle Identität jedes Menschen** und somit Kommunikationspartners wesentlicher Bestandteil seiner Person und nimmt Einfluss auf die eigentliche Kommunikation, auch auf die Mensch-Technik-Interaktion.

Der Ansatz des interkulturellen Designs verfolgt die Zielstellung der nutzer- und kulturorientierten Gestaltung interaktiver Systeme. Er basiert auf ethnologischen, psychologischen (insbesondere wahrnehmungspsychologischen) und

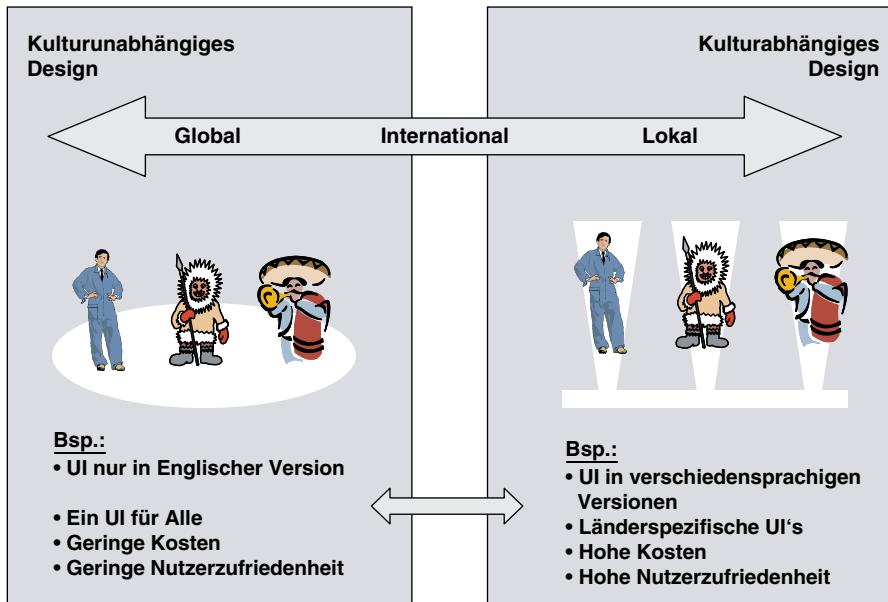


Abb. 4.1 Ansatz des interkulturellen Designs. (Röse 2002)

ergonomischen Studien und Erkenntnissen zu kulturellen Besonderheiten und Unterschieden sowie zahlreichen anthropologischen und kulturvergleichenden Untersuchungen. Diese Erkenntnisse werden genutzt, um ein Design zu unterstützen und zu entwickeln, welches auch die kulturellen Erfahrungen und Prägungen eines Produkt-/Systemnutzers berücksichtigt (Röse 2001). Dabei wird davon ausgegangen, dass die Erfahrungen von Nutzern grundsätzlich kulturell geprägt sind. Dies betrifft die Lern- und Verhaltensmuster, das Wissen und die darauf aufbauenden Interpretationen von Information und Problemlösestrategien. Die Kultur, in der ein Mensch aufwächst und sozialisiert wird, prägt ihn und hat somit Einfluss auf die Interaktion dieses Menschen als Nutzer technischer Geräte und das Vorwissen sowie die Interaktionsstile, die der Nutzer in die Interaktion einbringt.

In Abb. 4.1 ist der Grundansatz des interkulturellen Designs dargestellt. So wird zwischen kulturabhängigem und kulturunabhängigem Design unterschieden. Kulturunabhängiges Design zeichnet sich im Wesentlichen dadurch aus, dass ein Produkt/System in mehreren Ländern – ohne Änderungen – eingesetzt werden kann. Ein kulturabhängiges Design zeichnet sich durch seine Anpassung an die jeweilige Nutzerkultur aus. Zwischen diesen beiden „Polen“ des interkulturellen Designs gibt es unterschiedliche Varianten, Ansätze und Realisierungen, die sich alle auf der Dimension – in Abhängigkeit ihrer Intensität der kulturellen Anpassung – abbilden lassen.

Eine Unterteilung der Dimension des interkulturellen Designs in drei Grade hat Day 1996 für Softwareprodukte vorgenommen (Day 1996):

- **Globalisierung:** eine Art „culture-less“ internationaler Standard für die Nutzung in allen Märkten.
- **Internationalisierung:** eine Basisstruktur mit der Absicht einer späteren (kulturellen) Kundenanpassung, bei der die strukturellen und technischen Möglichkeiten dafür bereits geschaffen wurden.
- **Lokalisierung:** Entwicklung „culture specific packages“ für einen speziellen Zielmarkt.

Globale Produkte gelten somit als kulturunabhängig – in Bezug auf die Nutzerkultur (jedoch nicht kulturlos) – und werden nicht zum Bereich interkultureller Designprodukte gerechnet, jedoch im Ansatz des interkulturellen Designs berücksichtigt.

Je interaktiver ein Produkt ist, desto größer ist die Wahrscheinlichkeit, dass es eine Nutzerorientierung anstrebt und somit auch die kulturellen Aspekte berücksichtigen muss. Mit zunehmenden Interaktionsgrad eines Produktes steigt somit auch die Notwendigkeit einer kulturorientierten Produktgestaltung, denn Produkte der globalen Zukunft werden die Kultur als eine Variable des Nutzers berücksichtigen müssen, um nutzerfreundlich sein zu können.

Eine nutzergerechte und aufgabenorientierte Systemgestaltung ist somit zukünftig nur gegeben, wenn die kulturellen Unterschiede hinsichtlich Arbeitsrahmenbedingungen sowie Informationsstrukturierung und -austausch bei der Gestaltung neuer Mensch-Maschine-Systeme bzw. User-Interfaces integriert werden. Interkulturelles Design lässt sich somit wie folgt definieren (Röse 2002):

Interkulturelles Design beschreibt eine nutzer- *und* kulturoorientierte Gestaltung von interaktiven Systemen und Produkten, welche den kulturellen Kontext des Nutzers -in Abhängigkeit von der Produktnutzung und der zu erfüllenden Arbeitsaufgabe- berücksichtigt. Von einem interkulturellen Designprodukt wird nur dann gesprochen, wenn es die kulturellen Nutzerbedürfnisse in Form eines internationalen oder lokalisierter Designansatzes berücksichtigt.

Dabei wird davon ausgegangen, dass die Erfahrungen von Nutzern kulturell geprägt sind. Dies betrifft die Lern- und Verhaltensmuster, das Wissen und die darauf aufbauenden Interpretationen von Information und Problemlösestrategien. Die Kultur, in der ein Mensch aufwächst und sozialisiert wird, prägt ihn und hat somit Einfluss auf die Interaktion dieses Menschen als Nutzer technischer Geräte und das Vorwissen sowie die Interaktionsstile, die der Nutzer in die Interaktion einbringt.

Bei der ergonomischen Gestaltung von interaktiven Mensch-Maschine-Systemen und somit von Useware, sind daher auch interkulturelle Aspekte zu berücksichtigen.

4.2 Interkulturelles Useware-Engineering

4.2.1 Interkultureller Entwicklungsprozess

Bei der Entwicklung von softwarebasierten Produkten lassen sich allgemein vier Hauptphasen ausmachen (Wahl 2000). Diese sind:

- die Planungsphase (Produktidee, Hauptfunktionen),
- die Definitionsphase (Anforderungsspezifikation),
- die Entwurfsphase (Systemarchitektur, Funktionen, Schnittstellen) und
- die Implementierungsphase (Algorithmen, Programmstruktur, Implementierung). Diese Unterteilung ist auch für moderne Useware-Systeme nutzbar.

Bei den folgenden Ausführungen zum interkulturellen Useware-Engineering geht es primär um die Details in der **Entwurfsphase**. Damit sind die Teilbereiche: Datenbasis, Struktur, Interaktion und Präsentation gemeint. Das folgende Modell von Röse (Röse 2001) wurde in Anlehnung an das Modell von Rasmussen (Rasmussen 2000) für das Management des Entwicklungsprozesses von Useware – in der Entwurfsphase – erstellt. Es berücksichtigt die einzelnen Teilbereiche innerhalb der Entwurfsphase und ist mit seinen Wechselwirkungen im Sinne eines soziotechnischen Ansatzes konzipiert.

Die vier Bereiche: Daten(basis), Struktur, Interaktion und Präsentation bilden den Kern des Modells. Sie stehen in einem hierarchischen Bedingungsgefüge zueinander und repräsentieren die unterschiedlichen Ebenen eines Mensch-Maschine-Systems. Das Modell berücksichtigt dabei bewusst die Anwendung heterarchischer Strategien durch den Entwickler. Objektorientierte Entwurfsansätze des Software-Engineering hatten ebenfalls Einfluss.

Beim Engineering von ergonomischer und nutzerzentrierter Useware wird mit einer Nutzeranalyse begonnen und die dabei ermittelten Nutzeranforderungen werden als Ausgangsbasis für die weitere Entwicklung genutzt. Somit gehen die kulturbedingten Nutzeranforderungen sehr frühzeitig in den Entwicklungsprozess ein. Die Umsetzung der kulturbedingten Nutzeranforderungen ist dann im nächsten Schritt, der Gestaltung, zu finden. Das Resultat kulturbedingter Nutzeranforderungen ist ein kulturspezifisches User-Interface. Bei der Evaluation eines solchen User-Interfaces werden die kulturbedingten Nutzeranforderungen als Maßstab genommen und es wird bewertet, inwiefern die Realisierung den Anforderungen entspricht. Die Empfehlungen für ein Redesign sind somit ebenfalls kulturbedingt. Durch die Zentrierung der Systementwicklung auf den Nutzer, der durch seinen sozio-kulturellen Hintergrund geprägt ist, ziehen sich die kulturellen Einflüsse durch alle Entwicklungsphasen des Useware-Engineering. Somit sind die Planungs-, Definitions-, Entwurfs- und Implementierungsphase der Gestaltung von Useware gleichermaßen betroffen. Der im Folgenden vorgestellte Ansatz „interkultureller Usewaresysteme“ konzentriert sich auf die Entwurfsphase, auf eine Darstellung aller Engineeringphasen wird verzichtet.

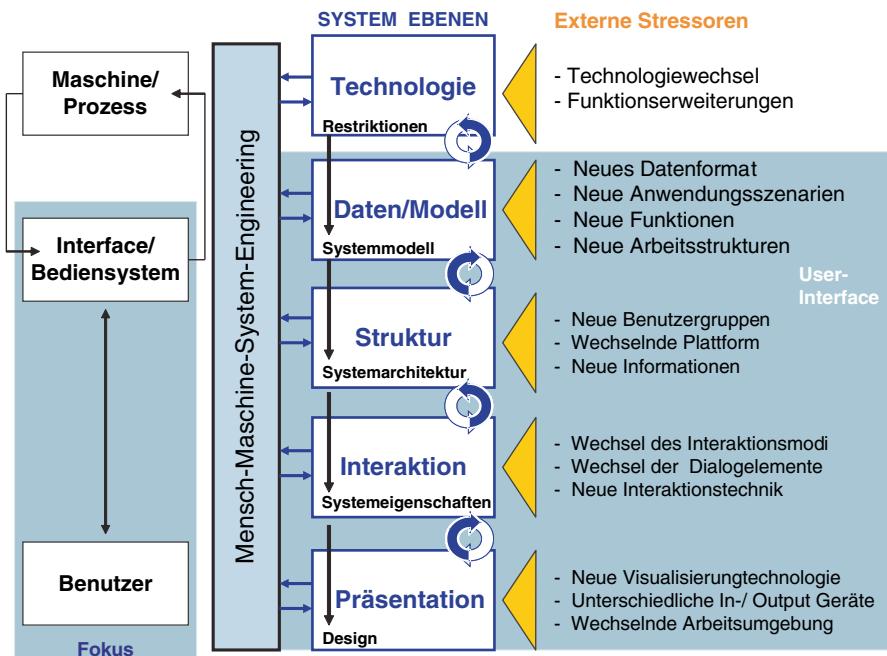


Abb. 4.2 Management der Entwurfsphase mit externen Stressoren. (Röse 2001, 2002)

Die Analyse des kulturell geprägten Nutzungskontextes und die daraus abgeleiteten Spezifikationen der kulturbedingten Nutzeranforderungen haben beim Useware-Engineering einen Einfluss auf die Systemebene Technologie (z. B. Funktionalitäten), Daten (z. B. Arbeitsstrukturen), Struktur (z. B. kulturbedingte Hierarchieunterschiede bedingen mehr Nutzergruppen als vorgesehen), Interaktion (z. B. eingeschränkte vs. unzählige Navigationsmöglichkeiten) und Präsentation (kulturbedingte Unterschiede der Informationscodierung). Diese Einflüsse müssen spezifiziert werden. Dazu wird erneut das Modell für die Entwurfsphase des Useware-Engineering aus Abb. 4.2 benutzt. Die im Modell dargestellten externen Stressoren, stellen die variabel einwirkenden Änderungsanforderungen an den Engineering-Prozess der Entwurfsphase dar. Diese externen Stressoren werden für den vorliegenden Fall der Entwurfsphase des interkulturellen Useware-Engineering als **kulturelle Stressoren** definiert. Durch die Darstellung der kulturellen Einflüsse für jede Systemebene werden die kulturellen Stressoren spezifiziert (s. Tab. 4.1).

Die dargestellten kulturellen Einflussfaktoren sind zu berücksichtigen, wenn interkulturelle Mensch-Maschine-Systeme entwickelt werden sollen. Sie geben jedoch nur einen Beispielhaften Überblick zu den möglicherweise vorfindbaren kulturellen Unterschieden.

Den einzelnen System-Ebenen der Entwurfphase des Useware-Engineering lassen sich statt Stressoren auch interkulturelle Variablen zuordnen, die sich unter-

Tab. 4.1 Kulturelle Stressoren für das Mensch-Maschine-System-Engineering (Röse 2001, 2002)

System Ebenen	Externe Stressoren	Kulturelle Stressoren
Technologie	Technologiewchsel	Andersartige Einsatzbedingungen (z. B. starke Stromschwankungen, klimatische Einflüsse (Luftfeuchtigkeit bei Steuerungen))
Daten/Modell	Funktionserweiterungen Neue Anwendungs-szenarien	Spezifische Funktionalitäten Einsatzzweck bestimmter Maschinentyphen kann kulturell variieren (z. B. Key-Production-Machine vs. Quality-Production-Machine), dementsprechende Anpassung des Systemmodells ist notwendig
Struktur	Neue Arbeitsstrukturen und -aufgaben Wechsel von Formaten und Standardformat-einstellungen Neue Nutzergruppen	Kulturelle Unterschiede der Nutzerstruktur und der Systemstruktur müssen beim Systemmodell berücksichtigt werden
Mental Modelle der Nutzer	Mental Modelle der Nutzer berücksichtigten	Kulturbedingte Unterschiede bei den Formaten, wichtig für Releases bei internationaler und lokalisierter Software
Interaktion	Wechselnde bzw. neue Funktionen Wechsel des Interaktionsmodus Integration/Änderung von Assistenzfunktionalitäten Wechsel der Dialogelemente	Kulturbedingte Unterschiede bei den Arbeitskonzepten, die sich oft auch auf ein kulturbedingtes eigenständiges Nutzergruppensystem auswirken (z. B. 10 statt 2 Nutzerhierarchien) Kulturelle Varianten bei den Nutzermodellen, oft einhergehend mit großen Differenzen des Nutzerwissens in verschiedenen Kulturen. Kulturspezifische Anforderungen und Änderungen hinsichtlich der Strukturen eruieren, integrieren, evaluieren
Präsentation	Neue Visualisierungstechnik	Integration der Übersetzung und Dokumentation in allen Versionen bzw. Dokumenten Kulturbedingte Präferenzen für bestimmte Interaktionsstile (z. B. direkt vs. indirekt, anonym vs. persönlich) Berücksichtigung kultureller Tabus und Präferenzen (z. B. bei Avataren)
		Analyse und Bewertung der Popularität und somit Akzeptanz sowie der Benutzbarkeit in der jeweiligen Kultur
		Kulturspezifische Vorlieben, Voraussetzungen und Tabus (z. B. für Darstellungen: Farben, Symbole, etc.)
		Anthropometrische Besonderheiten
		Kulturspezifische Benutzung von Darstellungsattributen (z. B. Farbe, Symbole) sowie Tabus
		Kulturspezifische Produktions- und Arbeitsbedingungen sowie Umweltfaktoren (z. B. Klimaeinfluss)

Tab. 4.2 Andockstellen interkultureller Variablen an Mensch-Maschine-Systeme. (Röse 2002, 2003a)

System-Ebenen der Entwurfsphase	Interkulturelle Variablen	Variablenart		
		Direkte Variable	Indirekte Variable	Randvariable
<i>Mensch-Maschine-System direkt betreffend</i>				
Technologie	Maschinenfunktionalitäten	■		
	Technischer Standard		■	
Daten/Modell	Maschinenfunktionalitäten	■		
	Arbeitsstrukturen		■	
Struktur	Bildungssystem		■	
	Arbeitsstrukturen		■	
	Service		■	
Interaktion	Interaktionsdesign	■		
	Dialogdesign	■		
	Technische Dokumentation		■	
Präsentation	Informationspräsentation	■		
	Sprache	■		
<i>Mensch-Maschine-System indirekt betreffend, (extern); Arbeits-/Umgebungs- und Rahmenbedingungen</i>				
	Allgemeines Maschinendesign	■		
	Technische Dokumentation	■		
	Service/Wartung	■		
	Bildungssystem		■	
	Umweltfaktoren		■	
	Politisches System	■		

teilen in direkte und indirekte Designvariablen sowie Randvariablen (Röse 2002). Die direkten interkulturellen Variablen haben einen unmittelbaren Einfluss auf die Gestaltung der Mensch-Maschine-Interaktion. Sie bestimmen ganz wesentlich die Benutzbarkeit und Akzeptanz des fertigen Useware-Systems im Sinne der DIN ISO 66234 Teil 8. Indirekte interkulturelle Variablen haben einen mittelbaren Einfluss auf die Gestaltung des Mensch-Maschine-Systems. Sie bestimmen im Wesentlichen die Anwendbarkeit des Systems und die Realisierbarkeit der Arbeitsaufgabe im Hinblick auf Nutzerorientierung und funktionaler Verfügbarkeit. Zum anderen sind sie wichtig, im Hinblick auf eine Kundenorientiertheit und -zufriedenheit. Die Randvariablen sind wichtig im Hinblick auf die Einsatzfähigkeit eines Systems. Dies gilt sowohl für physische Komponenten (z. B. Klimaschwankungen), als auch für strukturelle Änderungen (z. B. bei Arbeitssystemen und -strukturen).

Tabelle 4.2 gibt einen Überblick über die Integration der einzelnen interkulturellen Variablen in die Systemebenen der Entwurfsphase von Useware-Systemen. Die Methodik zur systematischen Integration interkultureller Varianten und Nutzeranforderungen ist in (Röse 2002) ausführlich beschrieben.

Diese theoretischen Ausführungen zu interkulturellen Variablen und interkulturellem Design sollen nun mit Anwendungsspekten ergänzt werden. Daher werden

im folgenden Kapitel anhand des Beispiels ‚Mainland China‘ praktische Aspekte des interkulturellen Useware-Engineerings beispielhaft dargestellt.

4.2.2 Useware-Design für den chinesischen Markt

Für den deutschen Maschinenbau ist China als Absatzregion mittlerweile immens wichtig. Kulturelle Unterschiede zwischen China und Deutschland sowie daraus resultierende kulturspezifische Nutzeranforderungen auf dem chinesischen Markt sind daher ein erkanntes Problem beim Export deutscher Maschinen nach China. Bereits im Projekt INTOPS 1², welches vergleichende Untersuchungen in den Ländern China, Korea, Indonesien, Indien und USA zum Inhalt hatte, konnten zum Teil erhebliche kulturelle Unterschiede nachgewiesen werden. Die Breite dieser Untersuchungen sowie die Tatsache, dass ein deutsches Untersuchungsteam die Arbeiten durchführte, ließ allerdings eine Reihe von Detailfragen unbeantwortet. Aufgrund dieser Situation wurde das Projekt „INTOPS 2 – Nutzersysteme für den chinesischen Markt“³ vom Zentrum für Mensch-Maschine-Interaktion (ZMMI) an der TU Kaiserslautern durchgeführt.

Aus der Praxis: Die Schwierigkeiten, denen sich ein deutsches Untersuchungsteam bei Kulturstudien gegenüber sieht, zeigt folgendes Beispiel aus der INTOPS 1 Studie:

In vielen asiatischen Kulturen gehört es sich nicht, mit „der Tür ins Haus zu fallen“. Es ist somit absolut unhöflich, einen Untersuchungstermin zu vereinbaren, sich nach einer kurzen Begrüßung zu den Maschinenbedienern leiten zu lassen und dort ein protokoliertes Interview zu führen. Der Kunde erwartet dort erst einmal ein persönliches Kennenlernen des Teams z. B. am Tag vorher bei einem ungezwungenen Abendessen, bei dem das notwendige Vertrauen zwischen den Seiten aufgebaut wird. Am nächsten Tag kann man dann mit ihm über die Studie und den Sinn und Zweck der Befragungen reden. Auch wenn es nicht gewünscht ist, so kann man es bei der Befragung häufig nicht vermeiden, dass der Chef bzw. Vorgesetzte ständig bei der Befragung anwesend ist und häufig auch in Absprache mit dem Werker die Antworten gibt. Ebenso ist es Usus, dass sich mehrere Maschinenbediener um den exotischen Befrager scharen und gemeinsam versuchen, „gute“ Antworten zu geben. Ein weiteres Problem stellen die oftmals notwendigen Übersetzer dar. Die Erfahrung zeigt hier, dass Übersetzer dazu neigen, Antworten in gebote-

² Gefördert durch das BMBF.

³ Gefördert durch einen Industriearbeitskreis, dem die Firmen MAN ROLAND (Offenbach), BÜHLER AG (Amriswil-CH), IWKA AG (Karlsruhe) und RITTAL GMBH & Co. KG (Herborn) angehörten.

ner Höflichkeit zu schönen und technische Details, die sie selbst nicht verstehen, mit Allgemeinfloskeln zu umgehen.

Das Projekt INTOPS 2 hat sehr deutlich gemacht, dass die Untersuchungsergebnisse deutlich besser ausfallen, wenn die Untersuchungen von Einheimischen geschult auf die Methoden sowie möglichst mit Kenntnis der englischen oder deutschen Sprache durchgeführt werden. Einen Beleg für diese Erfahrungen liefern auch viele international tätige Großunternehmen, die lokale Usability-Labs in den Hauptzielmärkten betreiben.

Das Ziel dieses Projekts war es, die kulturspezifischen Anforderungen chinesischer Nutzer an deutsche Maschinen herauszufinden und dabei Erkenntnisse aus INTOPS 1 verbessernd einfließen zu lassen. Im Folgenden werden einige Untersuchungsergebnisse dargestellt.

4.2.3 Das Projekt INTOPS 2

Im Projekt INTOPS 2 ging es um die Ermittlung von Nutzer- und Produktanforderungen in China (nur Mainland China). Die Studie wurde von 2000–2002 durchgeführt. Ergebnisse der Studie sind repräsentativ für den Bereich der Mensch-Maschine-Systeme, speziell im Bereich Produktionstechnik. Die Durchführung der Studie in China erfolgte durch einen chinesischen PhD-Studenten, denn bei der Ermittlung kulturspezifischer Nutzeranforderungen ist es unbedingt notwendig ‚native Speaker‘ zu integrieren, die sich in den ‚Feinheiten der Kultur‘ (Kommunikation, Mimik & Gestik, Verhaltensregeln, etc.) auskennen. Nur so können ‚reale Kulturspezifika‘ ermittelt werden.

Im Zusammenhang mit dem Projekt wurde ein Untersuchungsansatz zu kulturellen Einflussfaktoren entworfen. Dieser Ansatz ordnet die zu betrachtenden kulturellen Aspekte in zwei Hauptfaktoren ein: die kulturelle Mentalität und die kulturelle Umgebung (s. Abb. 4.3).

Die kulturelle Mentalität bezieht sich auf die psychologischen und Verhaltensbedingten Eigenschaften der Nutzergruppe, während die kulturelle Umgebung sich auf die physikalischen und organisatorischen Merkmale der Kultur bezieht. Beide Faktoren haben einen bedeutsamen Einfluss auf das Maschinendesign. Zur Ermittlung der kulturspezifischen Designmerkmale des chinesischen Marktes war eine Analyse chinesischer Nutzer vor Ort notwendig. Diese Analyse betrachtete sowohl den Faktor der *kulturellen Mentalität*, als auch der *kulturellen Umgebung*. Dadurch war eine ganzheitliche Erfassung der Kulturspezifika möglich, die Aussagen zu allen Aspekten des Maschinendesigns (Interface & Rahmenbedingungen) erlaubte.

In einer Vorbereitungsphase, die der empirischen Untersuchung vorausging, wurden Informationen zu: Studien der chinesischen Kultur, Eigenschaften des



Abb. 4.3 Kulturelle Einflussfaktoren

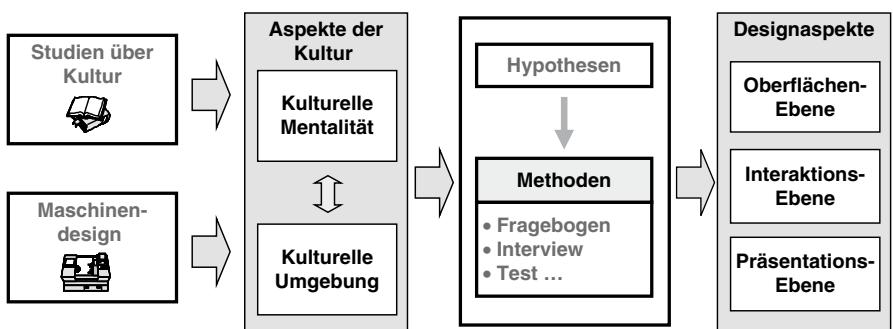


Abb. 4.4 Vorgehensweise im Projekt INTOPS 2

chinesischen Markts, chinesischen Technologienormen, Maschinendesignrichtlinien und aktuellen Maschinen-Designbeispielen gesammelt und ausgewertet. Basierend auf diesen Voruntersuchungen und Informationsrecherchen konnten Hypothesen zu Aspekten wie: allgemeines Maschinendesign, Interaktionskonzept, Menüstruktur, Navigation, Informationskodierung, Benutzungsoberflächenlayout, Nutzerunterstützungssystem und technische Dokumentation aufgestellt werden. Diese Hypothesen wurden durch verschiedene Untersuchungsmethoden, wie z. B. Test, Interviews und Fragebogen verifiziert. Mit den durch diese Methoden ermittelten Daten und den verifizierten Untersuchungshypothesen konnten kulturspezifische Designanforderungen abgeleitet werden (s. Abb. 4.4)

Die Durchführung der Vor-Ort-Untersuchungen in China erfolgte Anfang des Jahres 2001. Im Rahmen der Untersuchung wurden insgesamt 25 chinesische Indus-

Tab. 4.3 Kulturspezifische Aspekte des Maschinendesigns. (Röse 2002)

Erkennbarkeit/Erfassbarkeit	Design Ebene	Design-Aspekte
Sichtbar (leicht erfassbar)	Funktion	Maschinenfunktionalität Technische Merkmale Serviceaspekte etc.
	Oberfläche	Informationspräsentation Allgemeines Interface-Layout Sprachaspekte etc.
Versteckt (schwer erfassbar)	Interaktion	Allgemeine Strukturaspekte und Layout Dialogdesign Nutzerunterstützung, etc.

triebetriebe und Jointventures verschiedener Branchen wie: Werkzeugmaschinenbau, Motorenfertigung, Automontage, Steuerungstechnik und Verpackungstechnik besucht. Dort wurden Befragungen, Interviews und Tests mit 42 chinesischen Maschinennutzern und über 40 Service-Technikern, Ingenieuren, Designern und Betriebsverwaltern durchgeführt. Zahlreiche Untersuchungsdaten wurden gesammelt und analysiert (Zühlke 1998).

Einige interessante kulturelle Besonderheiten hinsichtlich der Maschinendesignmerkmale konnten ermittelt werden. Die weitergehende Bewertung und Interpretation der Untersuchungsergebnisse führte zur Formulierung wichtiger bezüglich des kulturorientierten Maschinendesigns für China, welche für die deutschen Maschinenhersteller bedeutsam sind, um ihre Maschinen an die Anforderungen chinesischen Nutzer anzupassen.

4.3 Kulturorientierte Gestaltung für den chinesischen Markt

4.3.1 Designaspekte

Beim kulturspezifischen Maschinendesign müssen verschiedene Designaspekte, die durch kulturelle Faktoren beeinflusst werden, umfassend berücksichtigt werden (s. Tab. 4.3).

Bisherige Ansätze konzentrierten sich auf die Designaspekte der Benutzeroberflächen (Piamonte 1999). Diese umfassten: die Informationspräsentation in Hinblick von Farbe, Symbole, Layout, Format, etc. sowie sprachbedingte Aspekte. Die praktische Erfahrung zeigt jedoch, dass im Bereich der Produktionsautomatisierung, insbesondere beim Maschinendesign für den asiatischen Markt, die Designaspekte jenseits der Benutzeroberfläche, z. B. die Maschinenfunktionen, technische Merkmale, Serviceorganisation, etc. für die Akzeptanz der Maschine durch den Nutzer ebenso entscheidend sind (Zühlke 1998). Daher wurden im Pro-

jekt INTOPS 2 nicht nur die Designaspekte der Benutzungsoberfläche, sondern auch allgemeine Nutzungsaspekte berücksichtigt. Tabelle 4.3 zeigt einen Überblick zu den Designaspekten und eine Einteilung in unterschiedliche Ebenen, basierend auf den Einflüssen hinsichtlich des Maschinendesigns und der Erfassbarkeit ihrer kulturellen Ausprägung.

Nachfolgend sind die wichtigsten Ergebnisse zur Gestaltung von Maschinen für den chinesischen Markt zusammengestellt. Dabei wurde eine Unterteilung nach den in Tab. 4.3 aufgeführten Designebenen: Funktion, Oberfläche und Interaktion unternommen.

4.3.2 Hinweise für die Funktions-Ebene

Maschinenqualität und Service sind die wesentlichsten Anforderungen auf der Funktions-Ebene. Der Vergleich chinesischer und importierter Maschinen zeigt einen entscheidenden Vorteil für deutsche Maschinen, der durch die hohe Qualität – als wesentlichstes Merkmal deutscher Maschinen – bestimmt wird. Die allseits geschätzte Qualität deutscher Werkzeugmaschinen ist bei den meisten Kaufentscheidungen ausschlaggebend. Damit sind primär die Fertigungsqualität, Genauigkeit und stabilen Laufleistungen gemeint. Also auf das zu fertigende Produkt und den Fertigungsprozess bezogene Merkmale, bei denen sich viele High-Tech-Anbieter heutzutage kaum mehr unterscheiden lassen. Eine wesentliche einflussnehmende und zu berücksichtigende Randvariable sind hierbei die chinesischen Importbestimmungen. In den Interviews wurde immer wieder darauf hingewiesen, dass die Maschinenqualität wichtiger ist als der Preis, denn nicht selten werden deutsche High-Tech-Maschinen auch als Prestige-Objekt angesehen und angeschafft. Die Untersuchungsergebnisse lassen ebenfalls die Aussage zu, dass die Vermarktung der deutschen Maschinenqualität eine dauerhafte Wettbewerbsfähigkeit auf dem chinesischen Markt erwarten lässt. Daher muss die Strategie einer Qualitätsminde rung zugunsten eines günstigeren und somit wettbewerbsfähigeren Preises hinterfragt werden.

Alle chinesischen Maschinennutzer stufen den Service als sehr wichtigen Be standteil des „Gesamtpaketes Maschine“ ein. Leider ist der Service der schwache Punkt der deutschen Maschinenhersteller. Die Erwartungen chinesischer Nutzer an den Service beinhalten: schnelle Reaktion auf Serviceanforderungen, umgehende Lieferung von Ersatzteilen, effiziente Schulung sowohl von Maschinennutzern als auch von Servicetechnikern, etc. Der Großteil der deutschen Maschinen hersteller wird diesen Anforderungen auf dem chinesischen Markt – insbesondere im Hinblick auf die beiden erstgenannten Punkte – noch nicht ausreichend gerecht. Es müssen daher Maßnahmen ergriffen werden, um das Service-System (wirksame Serviceverfahren, mehr Einweisung des Servicepersonals) und die Maschinengestaltung (einfache Wartung, Fernwartungsmöglichkeit, hohe Standardisierung der Ersatzteile) den Anforderungen anzupassen. Dies ist notwendig,

um einen konkurrenzfähigen Service auf dem chinesischen Markt anbieten zu können.

Eine andere Anforderung auf der Funktions-Ebene ist die Modularisierung der Maschinenfunktionalität. Da die Nutzertypen, Maßstäbe, Umfang der Produktion, Arbeitsorganisation, Mitarbeiterqualifikation etc. in China sehr unterschiedlich ausgeprägt sind, sind weder das „all-in-one“-Maschinendesign, welches die Maschine zu komplex und zu teuer für die Chinesen macht, noch die stark spezialisierten Maschinenfunktionalitäten, die wenig Raum für die Weiterentwicklung der Funktionen lassen, zum chinesischen Markt passend. Nur eine modularisierte Gestaltung der Maschinen bietet die Flexibilität, um die einzelne Maschine den Kunden- und Nutzerbedürfnissen anzupassen und gleichzeitig Platz für die Weiterentwicklung zu lassen. Die Anforderungen an die Modularisierung der Maschinenfunktionalität in China wurde von den Nutzern dahingehend spezifiziert, dass die Kontroll-/Überwachungsfunktionen der Maschine in die Basisfunktionen integriert werden sollten, während die Materialbearbeitungsfunktionen modularisiert sein sollten.

4.3.3 Hinweise für die Oberflächen-Ebene

Die wichtigste Anforderung der chinesischen Nutzer an die Designaspekte der Oberflächen-Ebene ist die Sprache. Dies beinhaltet sowohl die Sprache auf der Benutzungsoberfläche, als auch in der technische Dokumentation. Sofern möglich, sollten alle textuellen Beschreibungen und Anzeigen ins Chinesische übersetzt werden. Die Interviews haben gezeigt, dass in vielen Fällen die Verfügbarkeit chinesischsprachiger Benutzungsüberflächen und Dokumente die Kaufentscheidung deutlich beeinflussen würde. Falls die optimale Variante – eine Übersetzung aller Bildschirminhalte – nicht möglich ist, sollten zumindest die Beschriftungstexte von wichtigen Funktionen/Softkeys übersetzt werden. Diese einfache Lösung ist ein erster wichtiger Schritt zur Akzeptanz der deutschen Maschinen durch die chinesischen Nutzer.

Tests zum Verständnis englischsprachiger Begriffe haben gezeigt, dass Bezeichnungen wie ESC und PIN von chinesischen Maschinennutzern zu max. 20 % erkannt wurden (s. Abb. 4.5). Hier konnten Fehlerquellen für Fehlbenutzungen und daraus resultierende Stillstandzeiten sowie erhöhten Servicebedarf identifiziert werden. Bei der Übersetzung müssen allerdings auch die Unterschiede der Sprachgewohnheiten bzw. Dialekte in Hongkong, Taiwan oder Mainland China berücksichtigt werden. Durchgeführte Tests zur Farbkodierung haben gezeigt, dass – durch einen in den letzten Jahren vollzogenen Lernprozess der Maschinennutzer – die in der IEC 73 angegebenen allgemeinen Farbkodierungen eingesetzt werden können. Insbesondere die Farben Rot, Grün und Gelb werden von den meisten chinesischen Nutzern richtig interpretiert (Tab. 4.4).

Hingegen ist die Anwendung der international standardisierten Icons problematisch. Dies gilt speziell für sehr abstrakte und mehrfach interpretierbare (generalisierte)

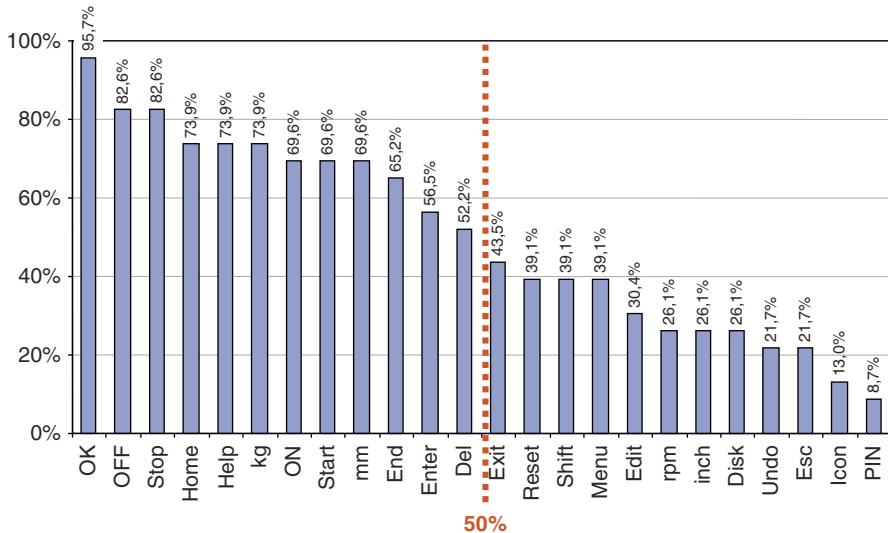


Abb. 4.5 Verständnis englischer Begriffe in Mainland China. (Röse und Liu 2002)

sierte) Icons. Die Anwendung von Icons sollte daher auf unkritische Maschineninformationen beschränkt werden. Falls die Anwendung von Icons zur Kodierung wichtiger Informationen nicht vermeidbar ist, sollte sie zusammen mit anderen Kodierungsarten – am besten jedoch mit einem Beschriftungstext – erfolgen. Auch bei der Verwendung von bildhaften Icons wurden Verständnisprobleme ermittelt, da bildhafte Darstellungen oft eng mit kulturellen Besonderheiten und Tabus verbunden sind. Schlussfolgernd bleibt festzuhalten, dass die Interpretation und das Verständnis der Icons für den jeweiligen Anwendungsfall – im Kulturkreis – zu testen ist.

4.3.4 Hinweise für die Interaktions-Ebene

Die Anforderungen der chinesischen Maschinennutzer an Designaspekte der Interaktions-Ebene wurden nicht direkt untersucht. Aus der Analyse der Untersuchungsergebnisse lassen sich jedoch folgende Aussagen ableiten. In China ist die Aufgabenverteilung unter verschiedenen Nutzern sehr ausgeprägt. Deswegen ist die Konfigurierbarkeit der Menüstruktur – in Abhängigkeit vom jeweiligen Nutzerprofil – ein ganz wichtiger Aspekt für die chinesischen Maschinennutzer. Jeder Nutzer sollte in seinem Profil nur die für ihn relevanten Informationen dargestellt bekommen. Dadurch soll eine schnellere Auftragserledigung unterstützt werden, da jeder Nutzer sein Profil aufgabenoptimiert anpassen kann.

Beispiele zur nutzerfreundlichen Strukturierung des Interaktionsbereiches zeigt Abb. 4.6 als ausgewähltes Ergebnis einer Befragung von chinesischen Maschinenbedienern (Röse et al. 2002). Dieses Beispiel zeigt, dass die Leserichtung durchaus

Tab. 4.4 Prozentrate der Erkennung von Farbkodierungen für unterschiedliche Kontexte und Konzepte. (Röse et al. 2003)

Kontext/Konzept	Farbe						Gelb
	Schwarz	Grün	Cyan	Blau	Magenta	Rot	
<i>Privat</i>	Glück	—	7,4	3,7	7,4	70,4	3,7
	Angst	59,3	11,1	3,7	14,8	3,7	7,4
	Hoffnung	—	40,7	11,1	3,7	7,4	29,6
	Stille	3,7	14,8	55,6	7,4	3,7	14,8
	Macht	25,9	7,4	3,7	25,9	11,1	14,8
	Gefahr	—	—	—	—	92,6	7,4
	Warnung	3,7	3,7	3,7	—	11,1	3,7
	Normal	—	74,1	11,1	7,4	3,7	—
	Aktion	44,4	—	22,2	25,9	3,7	—
	Allg. Info	11,1	14,8	44,4	18,5	3,7	—
<i>Operation</i>	On	7,4	74,1	3,7	—	—	7,4
	Off	7,4	3,7	—	—	—	85,2
	Hilfe	—	3,7	14,8	3,7	11,1	—
	OK	7,4	7,4	40,7	18,5	14,8	3,7
	Exit	44,4	7,4	18,5	3,7	18,5	—

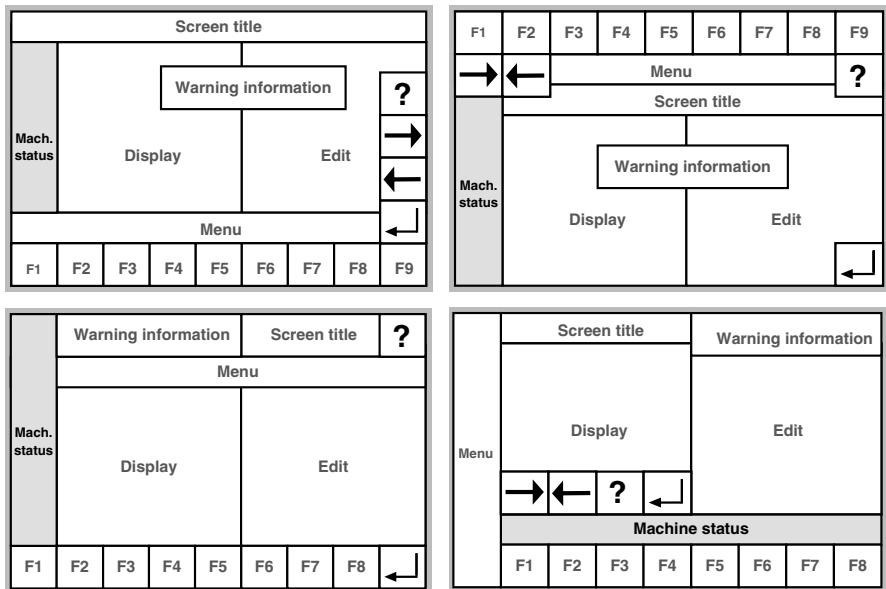


Abb. 4.6 Strukturierung des Interaktionsbereiches. (Röse et al. 2002)

einen Einfluss auf die Strukturierung wichtiger Informationsbereiche haben kann, z. B. bei der Überwachung des Maschinenstatus im dargestellten Beispiel. Die Ergebnisse bestätigen zwar eine Untersuchung von Dong und Salvendy aus dem Jahr 1999 (Dong und Salvendy 1999), sind aber in ihrer Aussage nicht statistisch signifikant. Sie werden jedoch als weiteres Indiz für die Anforderung konfigurierbarer Bediensysteme gesehen sowie des Ansatzes landesspezifischer Module für die Benutzungsschnittstelle.

Zusammenfassend bleibt festzustellen, dass die Zielsetzung bei der Entwicklung von Maschinenbenutzungssystemen für China auf die flexible Anpassbarkeit – der vom Hersteller vorgegebenen Menüstruktur – an die jeweilige Arbeitsorganisation sein sollte bzw. langfristig zu unterstützen ist.

4.4 Probleme kulturspezifischen Designs

Die Gestaltung interkultureller Mensch-Maschine-Systeme stellt eine bleibende Herausforderung dar. Untersuchungen (Hofvenschiold 2003; Ikonen 2003) haben gezeigt, dass selbst innerhalb der europäischen Union zwar einheitliche Währungen, etc. vereinbar sind, die Nutzeranforderungen jedoch kulturell variieren und nicht in einem europäischen Einheitsgeschmack vereinbar sind. Das interkulturelle Useware-Engineering ist untrennbarer Bestandteil einer konsequenten Kundenorientiertheit und eines user-centered Entwicklungsansatz, da im heutigen Globalis-

sierungszeitalter die Anzahl und Bedeutung der Produktexporte stetig steigt. Was sind die Hemmschwellen beim kulturspezifischen Design, bei der Produktlokalisierung, die es in der Praxis bzw. beider Realisierung zu überwinden gilt?

Ähnlich wie beim „klassischen Useware-Engineering“ gehören dazu:

1. Aufwand

Für die Ermittlung der einzelnen kultur- bzw. landesspezifischen Nutzeranforderungen sind zwar auch Useware-Experten notwendig, diese müssen jedoch von muttersprachlichen Kollegen aus der jeweiligen Kultur unterstützt werden. Derartige Dienstleistungen sind verbunden mit einem hohen Aufwand an Kosten und Zeit. Erst die langfristige Investition in einen kulturspezifischen Markt, in Kombination mit möglichst langen Produktlebenszyklen, ermöglicht eine entsprechende Amortisierung der Kosten

2. Komplexe Anforderungen an die Lokalisierung von MMS

Die alleinige Anpassung von Sprache oder die Evaluation von Produkten in verschiedenen Ländern können nur erste Schritte auf dem Weg zu einer kulturspezifischen Useware sein. Nur das Hand in Hand von systematischer Useware-Entwicklung, unter Einbindung und Berücksichtigung kultureller Spezifika, kann zu einer zufriedenstellenden Useware führen. Ein Ansatz dafür wurde im zurückliegenden Kapitel vorgestellt. Nur eine systematische Herangehensweise an alle Aspekte, die ein Mensch-Maschine-System darstellen, ist ein Erfolg versprechender Weg.

Oft ist auch eine verteilte Entwicklung von Produkten empfehlenswert bzw. die Einbindung lokaler Produktentwickler beim Useware-Engineering eines kulturspezifischen Produktes. Lokale Entwickler bringen neue Denkweisen und Strukturen in den Engineering-Prozess, die wiederum kulturell geprägt sind und somit der Zielrichtung eines kulturspezifischen Designs förderlich sind.

3. Fehlendes Know-how

Die Lokalisierung bzw. Internationalisierung von Useware stellt große Anforderungen an den Useware-Experten. Neben der Kenntnis interkultureller Besonderheiten kommt noch das nötige Produkt/Prozess-Know-how sowie die Grundlagen interdisziplinären Useware-Engineerings hinzu. Eine Kombination, über die in Deutschland nur sehr wenige Dienstleister verfügen. Gezielte Weiterbildungsangebote für Entwickler sind hier ebenfalls von Nöten.

4. Fehlende Verantwortlichkeiten und Einbindung in der Firma

Ein weiteres Problem stellt die oft unzureichende Einbindung des systematischen Useware-Engineerings in die einzelnen Geschäftsstrukturen dar. Dies macht es leicht, Verantwortlichkeiten für die Produktinternationalisierung zwischen den Abteilungen hin und her zu schieben. Nur in seltenen Fällen gibt es eine Abteilung bzw. einen Verantwortlichen, der sich ausschließlich um die kulturspezifische Anpassung des Produktes für den Kunden kümmert. Damit sind keine Übersetzungen gemeint; wie dargelegt geht es beim interkulturellen Useware-Engineering um wesentlich mehr, es muss eine ganzheitliche kulturspezifische Realisierung angestrebt werden.

Bei den Geschäftsleitungen wird die Herausforderung interkulturellen Useware-Engineerings oft unterschätzt. Dadurch werden meist unzureichende Budgets zur Umsetzung bereitgestellt, die negative Ergebnisse und damit meist die Abwendung von der Produktlokalisierung zur Folge haben. Aber wer kann schon einen Wolkenkratzer bauen, wenn er nur das Baumaterial für 4 Etagen hat?

5. Vielfalt des Marktes

Erfolgreiche Firmen sind meist auch global agierende Unternehmen. Dabei nimmt die Anzahl der Exportländer stetig zu. Oft fällt es schwer, sich für oder gegen den Eintritt in einen neuen Markt zu entscheiden. Bei dieser Entscheidung sollte auch der Aufwand für die Lokalisierung der Useware berücksichtigt werden. Der Erfolg in neuen Märkten hängt von der Kundenzufriedenheit ab und diese ist eng an eine gute Useware gekoppelt und muss mit entsprechendem Aufwand für Produktlokalisierung, Logistik, Service, Marketing, etc. angegangen werden. Daher kann es bei einer guten Analyse der Nutzerkultur manchmal durchaus ratsam sein, einen bestimmten Markt zu meiden, da die Zielgruppe des Produktes nicht identisch ist, unterschiedliche technische Voraussetzungen anzutreffen sind, religiöse Restriktionen gegen das Produkt sprechen oder andere kulturell und landesspezifisch begründete Einwände dagegen sprechen.

Abschließend bleibt zu erwähnen, dass schon der große Weltreisende Marco Polo wusste, dass seine Expeditionen nur erfolgreich sein können, wenn er offen für jede Kultur und Ihre Eigenheiten ist und sich bemüht, seinen Gegenüber zu verstehen und seine Wünsche zu respektieren. In diesem Sinne sollten auch Entwickler globaler Produkte die Anforderungen ihrer Nutzer respektieren und sie in die Produktentwicklung einbinden.

Aus der Praxis: Bei der Durchführung von internationalen Nutzerbefragungen muss bedacht werden, dass der Aufwand für die Organisation wesentlich größer ist als dies bei nationalen Analysen der Fall ist. Zeit kosten dabei nicht nur die umfangreicheren Reisevorbereitungen, sondern vor allem die Koordination der Termine. Teilweise müssen auch Dolmetscher hinzugezogen werden, was zusätzlichen Aufwand mit sich bringt.

Zu bedenken ist auch, dass auch innerhalb der Länder unterschiedliche Sprachen oder Dialekte gesprochen werden. So gibt es in den USA durchaus die Situation, dass die Mitarbeiter im Unternehmen besser Spanisch als Englisch sprechen. Ebenso kann es in ländlichen Umgebungen passieren, dass die Mitarbeiter sehr unverständliche Dialekte sprechen. Während die Analysen in den USA und Europa allerdings noch relativ einfach machbar sind, bedeuten Analysen in Fernost aufgrund der vielfältigen Kulturunterschiede meist einen sehr großen Aufwand.

Kapitel 5

Hilfsmittel bei der Useware-Entwicklung

Moderne Useware-Systeme mit allen Anforderungen, die bislang vorgestellt wurden, lassen sich nicht mehr ohne geeignete Hilfsmittel entwickeln. Auch das Software-Engineering lebt heutzutage sehr stark von der Auswahl der richtigen Entwicklungswerzeuge. In diesem Kapitel werden daher einige Hilfsmittel für die Useware-Entwicklung vorgestellt. Zum einen sind dies die Entwicklungswerzeuge und zum anderen die Beschreibungstechniken. Die Hilfsmittel können in verschiedenen Phasen der Useware-Entwicklung eingesetzt werden (s. Abb. 5.1).

Die Entwicklungswerzeuge lassen sich nach ihren Einsatzmöglichkeiten, Einsatzgebieten, aber auch Einsatztechniken gruppieren. Universelle Werkzeuge können in allen Anwendungsbereichen eingesetzt werden. Dagegen sind viele Visualisierungswerzeuge auf den Produktions- bzw. Prozessbereich zugeschnitten. Rapid-Prototyping-Werkzeuge ermöglichen eine schnelle und kostengünstige Erstellung von Prototypen für Useware-Bedienoberflächen. Modellbasierte Werkzeuge verfolgen den Ansatz, verschiedene Modelle zu Beschreibung von relevanten Useware-Aspekten zu verwenden. Dadurch ermöglicht diese Technik Skalierbarkeit und Wiederverwendbarkeit der Modelle.

Zur Beschreibung von Useware-Aspekten gibt es unterschiedlichen Notationen. Hier sind vor allem UML und XML zu erwähnen. Während UML eine universelle und mittlerweile Standard-Modellierungssprache ist, ist XML eine plattformübergreifende¹ bzw. plattformunabhängige Auszeichnungssprache. Auf Basis von XML kann man dann wieder problemspezifische Beschreibungssprachen entwickeln, wie es an dem Beispiel *useML* vorgestellt werden soll.

¹ Hier ist sowohl die Hard- als auch die Softwareplattform gemeint.

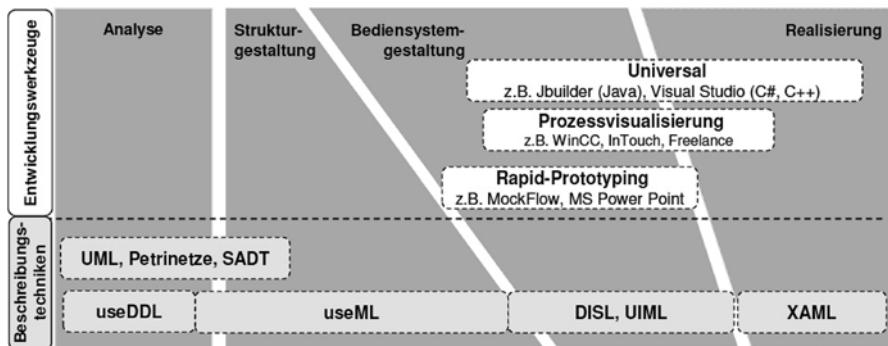


Abb. 5.1 Hilfsmittel bei der Useware-Entwicklung

5.1 Entwicklungswerkzeuge

5.1.1 Universelle Entwicklungswerkzeuge

Eine weit verbreitete Kategorie der Entwicklungswerkzeuge sind die universellen Tools der klassischen Programmiersprachen wie bspw. C++, C#, VISUAL BASIC (VB), JAVA und DELPHI. Diese sind den meisten Entwicklern geläufig und bieten ein großes Funktionsspektrum zur Realisierung von Algorithmen und zur Datenverarbeitung. Durch grafische Editoren ist es möglich, Benutzungsschnittstellen schnell zu erstellen, wobei den Entwicklern aber sehr wenig Unterstützung hinsichtlich ergonomischer Gestaltungsprinzipien gegeben wird. Meist resultieren daraus Bediensysteme, deren Grundstruktur dem funktionalen Aufbau der Software folgt und die sich nicht an den Bedürfnissen der Nutzer orientieren. Nachteil der grafischen Programmierumgebungen ist außerdem, dass bei Änderungen an Funktionalität oder Benutzungsschnittstelle ein komplexes Geflecht von Funktionen und Visualisierungsobjekten zu durchschauen ist.

Diese Werkzeuge sind dadurch gekennzeichnet, dass sie dem Entwickler grafische Elemente (*widgets*) zur Benutzungsschnittstellenentwicklung zur Verfügung stellen. Die Werkzeuge sind nicht auf ein bestimmtes Anwendungsgebiet begrenzt, sondern universell einsetzbar. Mit ihnen kann also eine Benutzungsschnittstelle für ein Krankenhausinformationssystem genauso wie für eine Maschinensteuerung entwickelt werden. Somit sind sie generisch und bieten ein breites Entwicklungsspektrum. In einem Editor für Benutzungsschnittstellen erfolgt eine visuelle Entwicklung und der damit verbundene Code kann im Codeeditor bearbeitet werden. Meistens ist auch ein „Fensterviewer“ zum Testen integriert. Trotz ihres breiten Einsatzspektrums fehlt aber eine Unterstützung durch ergonomische Gestaltungsregeln. Sie überlassen dem Entwickler alle Gestaltungsfreiheiten. Dieser muss sich allein auf sein Wissen und seine Fähigkeiten verlassen, um die Ergonomie in die Benutzungsschnittstelle zu bringen. Tabelle 5.1 gibt einen Überblick über einige universelle Entwicklungswerkzeuge.

Tab. 5.1 Überblick über einige aktuelle universelle Entwicklungswerkzeuge

	JBuilder 2008 R2 ^a (Ultimate) ^b	Visual Studio 2010 (Ultimate) ^b	Sun Studio 12 ^c	Beschreibung
<i>Editor</i>				
Projekt-Explorer	Ja	Ja	Ja	Übersichtsfenster
Wortvervollständigung	Ja	Ja	Ja	Vervollständigung bekannter Begriffe
Code-Abdeckung	Ja	Ja	Ja	Echtzeit-Darstellung bereits abgedeckter Klassen
Syntax-Highlighting	Ja	Ja	Ja	Vis. Unterscheidung von Konstrukten/Begriffen
<i>UML-Modellierung</i>				
Integrierte UML -Unterstützung	Ja	Ja	Ja	Integrierter UML Editor
Roundtrip	Ja	Teils	Nein	Synchronisation von UML-Modell und Code
<i>Entwicklungsumgebung</i>				
Programmiersprachen	Java	VB, VC#, VC++, VJ#, VF#	C, C++, Fortran	Verwendete (Unterstützte) Programmiersprache(n)
Betriebssystem	Windows, Solaris, Linux, Mac	Windows	Solaris, Linux	
Deutsche Version	Nein	Ja	Nein	
XML-Datenaustausch	Ja	Ja	Ja	
<i>Debugger</i>				
Code-Analyser	Ja	Ja	Ja	Analyse von Source-code
Anzeige von Ausdrücken/Variablen beim Debuggen	Ja	Ja	Ja	Ausdrücke/Varianten beim Debuggen anzeigen und überwachen
Popup-Hilfe bei Syntax-Fehlern	Ja	Ja	Ja	
Code-Änderung im Debugg-Modus	Ja	Ja	Ja	
Leistungsanalyse	Ja	Ja	Ja	
Code-Überprüfung anhand bereitge- stellter Metriken	Ja	Ja	Nein	Analyse des Codes hinsichtlich des verbrauchten Speichers und der benötigten CPU-Zeit Bewertung des Codes hinsichtlich z. B. Kohäsion um die Gütekriterien eines Systems zu bestimmen

Tab. 5.1 (Fortsetzung)

	JBuilder 2008 R2 ^a	Visual Studio 2010 (Ultimate) ^b	Sun Studio 12 ^c	Beschreibung
Thread Analyse	Ja	Nein	Ja	Identifikation eigentlicher und potentieller Thread-Errors
<i>Extras</i>				
Kollaboration	Ja	Ja	Nein	Zusammenarbeit mit anderen Teammitgliedern
Webservice-Designer	Ja	Ja	Nein	u. a. visuelle Darstellung der WSDL
Datenbank	Ja	Ja	Teils	Zur Unterstützung von Reuse-Konzepten
Reverse-Engineering	Ja	Nein	Nein	
Open Source	Nein	Nein	Ja	

^a Embarcadero Technologies, Inc. (<http://www.embarcadero.com/products/jbuilder>)^b Microsoft Corp. (<http://www.microsoft.com/visualstudio/en-us/products/2010-editions/ultimate>)^c Oracle Corp. (<http://developers.sun.com/sunstudio/>)

5.1.2 Prozessvisualisierungswerkzeuge

Besonders in einigen anwendungsspezifischen Domänen wie dem Maschinen- und Anlagenbau haben sich Entwicklungswerkzeuge wie z. B. Wonderware INTOUCH, SIEMENS SIMATIC WINCC oder INOSOFT VISIWINNET etabliert, die genau auf die entsprechenden Anforderungen zugeschnitten sind (s. Tab. 5.2). Prozessvisualisierungswerkzeuge stellen eine Kombination aus Visualisierungsbaukästen und gerätespezifischen Prozesskopplungstreibern dar. Sie bieten Programmvorlagen, die die einfache Realisierung der Maschinen- oder Anlagenfunktionalität ermöglichen. Sensoren und Aktoren können problemlos angebunden und programmiert werden. Neben den Komponenten zur Realisierung der Prozesssteuerung verfügen diese Werkzeuge über Editoren für die grafische Ausgestaltung der Benutzungsschnittstelle. Für bestimmte, in der Prozessvisualisierung häufig verwendete Visualisierungsobjekte wie z. B. Ventile und Stellmotoren bringen sie entsprechende Vorlagen mit, die nur noch parametrisiert und an die Anlage angepasst werden müssen.

Anwendungsschwerpunkt der Prozessvisualisierungswerkzeuge ist aber noch immer die Herstellung der Prozessfunktionalität unter Einbindung vielfältiger Sensor- und Aktor-Komponenten des Maschinen- oder Anlagenumfeldes. Automatische Sensorankopplung, große Treiberbibliotheken für Automatisierungskomponenten und die schnelle Logikprogrammierung werden wichtiger als die Unterstützung der nutzergerechten Gestaltung erachtet.

Aus der Praxis: Eine Besonderheit bei den Prozessvisualisierungswerkzeugen stellen die Vertriebsmodelle dar. Meistens werden diese Entwicklungswerkzeuge zu einem vergleichsweise günstigen Basislizenzpreis angeboten. Für jede damit erzeugte Applikation werden dann Runtime-Lizenzkosten fällig, die sich häufig an der Anzahl der benutzten Prozessvariablen orientieren. Diese Lizenzpolitik ist auf das typische Anwendungsfeld der großen und komplexen Prozesssteuerungen in der Verfahrens- und Energietechnik abgestimmt, die durch wenige aber sehr umfangreiche Installationen gekennzeichnet sind. Für die typischen Anwendungen bei Maschinensteuerungen, die eher durch viele Lizenzen mit jeweils geringer Komplexität gekennzeichnet sind, führt diese Lizenzpolitik häufig zu nicht akzeptablen Kosten.

5.1.3 Domänenspezifische Entwicklungswerkzeuge am Beispiel Automotive

Eines der populärsten, kommerziell verfügbaren Werkzeuge für die modellbasierte Entwicklung von automotive-HMI-Systemen ist das durch die Firma Elektrobit

Tab. 5.2 Überblick über einige Prozessvisualisierungswerkzeuge

	Wonderware INTOUCH 10.0 HMI ^a	Simatic WinCC Flexible ^b	VisiWinNET ^c	Beschreibung
<i>Systemvoraussetzungen</i>				
Prozessor (min.)	Pentium PC	Pentium PC	Pentium PC	
Betriebssystem	Windows	Windows	Windows	
Deutsche Version	Ja	Ja	Ja	
<i>Dialoggestaltung/Visualisierung</i>				
Erzeugen von Anzeigobjekten	Drag & Drop	Drag & Drop	Drag & Drop	Wie kommen Objekte auf die Arbeitsfläche
Einstellungen dynamischer Eigenschaften	Parametrierung	Parametrierung	Parametrierung	Wie werden dynamische Eigenschaften definiert?
Programmierung	.NET	Visual Basic	Visual Basic	Welche Programmiersprache?
Format für Grafiken	Windows-kompatibel	Standard (z. B. JPG)	Standard (z. B. JPG)	Unterstütztes Grafikformat
<i>Anbindung an den Prozess</i>				
Verknüpfung von Variablen	E/A, D/A und OPC-Server	OPC Server	OPC Server	Wie werden Variablen mit Anzeigen verknüpft?
SPS-Treiber	Viele	Viele	Alle	unterstützte SPS-Treiber
<i>Weiteres</i>				
Kopierschutz für Applikationen	Access-Level Password Security, MS-Authentification, Data-Level-Security, FDA 21 CFR 11 Funktionalität, Read-Only Software	Dongle/Software	Dongle/Software	Schutz der Applikation gegen unerlaubte Verwendung
Online Konfiguration	Ja	Ja	Ja	

^a Wonderware GmbH (<http://www.wonderware.de>)^b Siemens AG (<http://www.automation.siemens.com>)^c Inosoft GmbH (<http://www.inosoft.com>)

Automotive (ehemals 3SOFT) entwickelte und vertriebene EB GUIDE Studio. EB GUIDE Studio bietet als integrierte Entwicklungsumgebung (IDE) die Möglichkeit Benutzungsschnittstellen mittels grafischer Editoren, z. B. View-Editoren für die Bildschirmgestaltung und Zustands-Editoren für die Ablaufspezifikation, zu entwerfen. Durch die Integration der Open Graphics Library (OpenGL) erlaubt die Entwicklungsumgebung weiterhin die Gestaltung von 3D-Benutzungsschnittstellen. Ab der Version 3.0 bietet die Entwicklungsumgebung zusätzlich die Möglichkeit multimodale Konzepte in Form sprachbasierter Dialoge zu modellieren.

Im Anschluss an den Gestaltungsprozess kann für die modellierte Benutzungsschnittstelle C++ Code erzeugt werden, der auf der Fahrzeug-Zielplattform lauffähig ist. Integrierte Test-Werkzeuge unterstützen die Simulation der modellierten Benutzungsschnittstelle und erlauben somit eine frühzeitige Evaluation der Entwicklungsergebnisse.

Für die Verwaltung des Datenmodells der Benutzungsschnittstelle wird ein proprietäres, zentrales XML-Datenformat verwendet, was die Benutzungsschnittstelle mit ihren unterschiedlichen Sichten abspeichert.

Mittels der PlugIn-Unterstützung der Entwicklungsumgebung können weitere Daten, wie z. B. Designvorlagen, in den Entwicklungsprozess integriert werden, welche die Entwickler beim Rapid-Prototyping der Benutzungsschnittstellen unterstützen.

5.1.4 *Rapid-Prototyping-Werkzeuge*

Neben den bislang vorgestellten Entwicklungswerkzeugen, deren Schwerpunkt die funktionale Realisierung ist, existiert noch die Gruppe der Rapid-Prototyping-Werkzeuge, die ein schnelles Gestalten und Evaluieren von Benutzungsschnittstellen ermöglichen. Mittels prototypischer Benutzungsschnittstellen (Mock-Ups) kann eine gemeinsame Verständnisbildung über die visualisierten Konzepte erreicht werden. Jedoch können die erstellten Prototypen in der Regel nicht als Software auf das Bediensystem übertragen werden. Wie bereits aus der Bezeichnung hervorgeht, steht das schnelle Erstellen von Prototypen im Vordergrund.

Mit steigender Leistungsfähigkeit der visuellen Programmierumgebungen haben die Rapid-Prototyping-Werkzeuge an Bedeutung verloren und wurden teilweise in bestehende Programmierumgebungen, mit dem Vorteil, dass die Prototypen weiterverwendet werden können, integriert. Allerdings existieren eine Reihe von Werkzeugen, die sehr effizient in den frühen Phasen der Gestaltung einer Benutzungsschnittstelle zum Zwecke der Evaluation mit Endnutzern getestet werden können. Mit einfachen Werkzeugen wie bspw. MockFlow oder Balsamiq Mockups ist es möglich grafische Mock-Ups zu erstellen und mit Endnutzern zu diskutieren. Eine interaktive Evaluierung oder Simulation ist mit diesen Werkzeugen jedoch nicht möglich.

An Rapid-Prototyping-Tools werden i. d. R. folgende Anforderungen gestellt:

- Einfache und schnelle Erzeugung von Mock-Ups
- Darbietung möglichst vieler grafischer Elemente
- Verbesserte Wiederverwendbarkeit

- Unterstützung unterschiedlicher (herkömmlicher) Ein- und Ausgabegeräte (Tastatur, Maus, Bildschirm)
- Möglichkeit der Interaktion und Simulation

Aus der Praxis: Als ein universell einsetzbares Werkzeug bei der Useware-Entwicklung hat sich das Office-Tool MS POWERPOINT erwiesen. Grundsätzlich besteht durch die Interaktivitätsfunktionen dieses Programms ein vergleichbares Funktionsspektrum wie das der etablierten Rapid-Prototyping-Werkzeuge. POWERPOINT bietet eine umfangreiche Bibliothek vorgefertigter Interaktionselemente und grafischer Zeichenwerkzeuge sowie Funktionen zur Interaktionssteuerung. Außerdem können über die Programmierschnittstelle VBA viele weitere Effekte selbst programmiert werden. Auf einfacherem Wege ist es also möglich, grafisch ausgestaltete und am Bildschirm bedienbare Interaktionssysteme zu entwerfen und prototypisch umzusetzen (s. Abb. 5.2).

Der immense Vorteil dieser Software liegt aber in ihrem großen Verbreitungsgrad. Auf fast jedem Office-PC können so Entwürfe wiedergegeben und direkt diskutiert werden. Lizenzprobleme oder aufwändige Installationen von Runtime-Umgebungen gibt es hierbei nicht. Vor allem Entscheidungsträger können so ständig über den aktuellen Stand der Entwicklungen informiert werden. Nicht zu vergessen ist auch die Bedeutung dieser Daten zur Formulierung eines Lastenhefts bzw. der Produktdokumentation. Die Ansichten der Powerpoint-Prototypen können auf einfacherem Weg direkt in Textdokumente z. B. im Word-Format eingebunden werden und so die Gestaltungentscheidungen dokumentieren.

Der Nachteil der Verwendung von POWERPOINT ist, dass vor allem bei komplexeren Bediensystemen Änderungen mit zunehmendem Umfang des Prototyps immer mehr Aufwand bedeuten. Damit entsteht sehr schnell ein Konsistenzproblem, wenn z. B. Größen oder Gestalt von Interaktionselementen nicht in allen Ansichten geändert wurden. Außerdem sind Programmierkenntnisse in VBA erforderlich, um die erwartete Dynamik des Prototyps zu erreichen. Ferner fehlen Tools zum Transformieren von Dateien in bekannte Sprachen wie XML, C++ oder Java. Für die Realisierung müssen die POWERPOINT-Prototypen in der Entwicklungsumgebung erneut erstellt werden, was wieder Aufwand bedeutet und zu Inkonsistenzen führen kann.



Abb. 5.2 Beispiel eines Automatenprototypen mit MS POWERPOINT

5.2 UML – Unified Modeling Language

Der Grundgedanke von UML ist, Entwicklern von Softwaresystemen ein adäquates Werkzeug zur Verfügung zu stellen, um die Geschäftsprozesse, Anforderungen und die daraus resultierenden (statischen und dynamischen) Entwürfe ihres Produktes visuell beschreiben zu können und damit eine leicht verständliche Kommunikationsbasis zwischen Entwicklern zu garantieren.

UML ist eine objektorientierte Modellierungssprache, die die Spezifikation von Softwaresystemen unabhängig von Fach- oder Spezialgebieten ermöglicht. Dazu definiert sie sowohl Notationselemente und Beschreibungen zu deren korrekter Anwendung als auch Erweiterungsmechanismen, um eigene Diagrammarten bzw. Modifikationen von Diagrammen auf Basis dieser Standardkonstrukte erstellen zu können.

Das Spektrum der Anwendungsgebiete von UML reicht von der Beschreibung von „klassischen“ Datenbankanwendungen über Workflow-Anwendungen bis hin zu Echtzeitsystemen. Dies zeigt klar das Ziel, welches von den Entwicklern der UML verfolgt wird: eine einheitliche Darstellung einer Vielzahl von funktionalen Elementen von Softwaresystemen mittels einheitlicher Notation.

Mit der Entwicklung von UML wurde Anfang der 90er Jahre begonnen. Im Januar 1997 wurde UML in der Version 1.0 veröffentlicht und durch die Object Management Group (OMG) im November 1997 standardisiert. Im Februar 2009 hat die OMG die aktuelle Version 2.2 veröffentlicht.

UML ist eine Modellierungssprache und somit weder eine Methode noch eine Vorgehensweise. Es werden keine Anleitungen zur Verfügung gestellt, welche Diagrammart in welchem Schritt eines Entwicklungsprozesses benutzt werden sollten. Gerade bei „neuen“ Entwicklern, die die verschiedenen Diagrammarten noch nicht kennen, kann dies zu Problemen führen. Die UML 2.2 definiert 14 verschiedene Diagrammtypen, die sich in die zwei Hauptkategorien (statische) Strukturdiagramme und (dynamische) Verhaltensdiagramme unterteilen lassen. Abbildung 5.3 zeigt die Gliederung der Diagrammtypen im UML 2.2 Sprachraum.

Zu den Diagrammtypen zählen bspw. Anwendungsfalldiagramme (engl. *use case diagram*), Klassendiagramme (engl. *class diagram*), Sequenzdiagramme (engl. *sequence diagram*) oder Zustandsdiagramme (engl. *state diagram*). Ein UML-Modell besteht aus mehreren dieser Diagramme, die in ihrer Gesamtheit ein Softwareprodukt möglichst vollständig modellieren sollen (Neumann 2002).

Die Erstellung solcher Diagramme geschieht meist mit entsprechenden Softwareentwicklungswerkzeugen. Bekannte Vertreter sind hier IBM RATIONAL ROSE, RATIONAL RHAPSODY, OBJECTIF, ARGOUML, MICROSOFT VISIO, BOUML, NETBEANS, oder STARUML. Diese Werkzeuge dienen meist nicht nur zur Erstellung der Diagramme, sondern helfen dem Softwareentwickler auch bei der Erstellung des Quellcodes und der Dokumentation. Inzwischen existieren eine Vielzahl von Werkzeugen, die nahezu alle Anwendungsbereiche abdecken: Angefangen bei einfachen Werkzeugen zum Zeichnen von UML-Diagrammen über integrierte Entwicklungsumgebungen mit automatischer Quellcodegenerierung, Prototyping

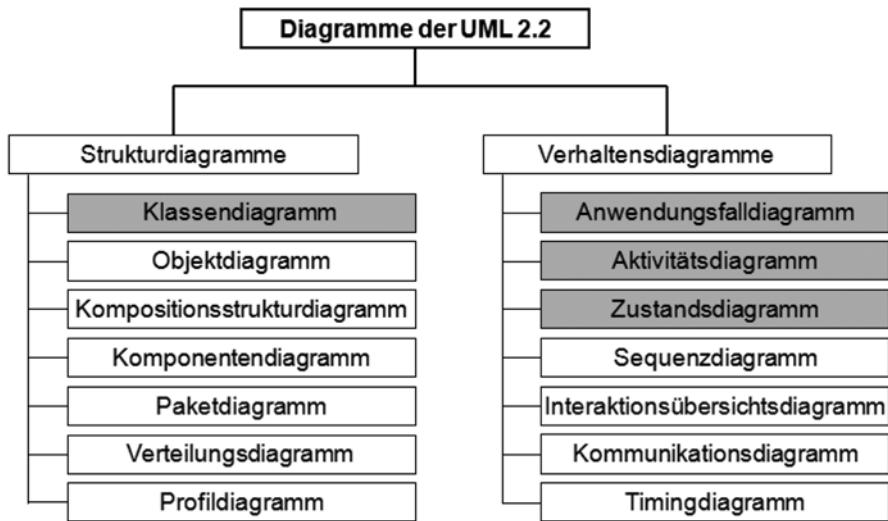


Abb. 5.3 Gliederung der Diagrammtypen der UML 2.2

der Benutzungsschnittstelle und Reverse-Engineering zu kompletten Paketen mit Testwerkzeugen, Teamunterstützung und Konfigurationsmanagement. Speziallösungen decken Bereiche wie Real-Time-Modelling und Prozesssysteme ab.

Trotzdem ist Schwerpunkt vieler UML-Werkzeuge die Modellierung der Systemarchitektur (*back-end*) und nicht seiner Benutzungsschnittstelle (*front-end*). D. h. nach der Modellierung muss die Benutzungsschnittstelle mit getrennten Werkzeugen gestaltet werden. Da die meisten UML-Werkzeuge auch die Möglichkeit anbieten, das Codegerüst des Modells für Programmiersprachen wie C++ oder JAVA zu generieren, findet die eigentliche Gestaltung dann in diesen generischen Entwicklungsumgebungen statt.

Es gibt sicherlich nicht *das* Werkzeug zur Modellierung von Software, jedes Werkzeug hat seine Vor- und Nachteile.

5.2.1 UML-Anwendungsbeispiel

Eine Firma möchte für ihre Maschine ein Bediensystem entwickeln, das die Benutzergruppen aus Tab. 5.3 unterstützen soll.

Da die Benutzer unterschiedlich qualifiziert sind, soll der Zugriff über das Bediensystem über ein Login erfolgen. So kann sichergestellt werden, dass keine unbefugten Zugriffe stattfinden. Nach dem Login soll der Benutzer nur die für ihn relevanten Informationen und Funktionen angezeigt bekommen. Tabelle 5.4 zeigt einige Benutzeroberflächen.

Tab. 5.3 Benutzergruppen

Nutzergruppe	Ausbildungsgrad
Operator (O)	Keine oder niedrige Ausbildung
Schichtleiter (S)	Facharbeiter
Experte (E)	Hoch ausgebildeter Ingenieur

Tab. 5.4 Benutzerobergaben

Produktion starten/stoppen (Alle)	Wartung durchführen (S, E)
Maschine konfigurieren (E) Störung beheben (S, E)	Aufträge laden/ löschen (S) Archivierung durchführen (S)

Die Aufgabe ist es, dieses Bediensystem mit der UML-Notation modellhaft darzustellen.

5.2.2 *Anwendungsfalldiagramm*

Das Anwendungsfalldiagramm ist ein Verhaltensdiagramm. Es zeigt eine Sicht auf das erwartete Verhalten eines Systems und wird deshalb für die Spezifikation der Anforderungen an ein System eingesetzt, wobei ein Anwendungsfall typischerweise eine Interaktion eines Nutzers mit einem System innerhalb eines Geschäftsprozesses beschreibt. Es wird damit das gewünschte externe Systemverhalten aus der Sicht des Benutzers und somit auch Anforderungen, die das System aus Nutzersicht zu erfüllen hat, beschrieben.

In einem Anwendungsfalldiagramm werden Anwendungsfälle und Akteure mit ihren Abhängigkeiten und Beziehungen dargestellt. Ein Akteur ist ein bestimmter Nutzer des Systems. Ein Anwendungsfalldiagramm stellt somit eine logisch zusammengehörige, wiederkehrende Anwendung innerhalb eines Systems dar. Abbildung 5.4 zeigt das Anwendungsfalldiagramm für das Bediensystem aus dem Beispiel.

5.2.3 *Klassendiagramm*

Nachdem die Anwendungsfälle des Systems modelliert worden sind, erfolgt eine Beschreibung der internen Systemstruktur. Die Struktur besteht aus unterschiedlichen Objekten mit unterschiedlichen Eigenschaften. Ferner existieren Beziehungen zwischen den Objekten. Diese Objekte und die Beziehung zwischen ihnen werden in einem Klassendiagramm beschrieben. Wichtige Elemente eines Klassendiagramms sind:

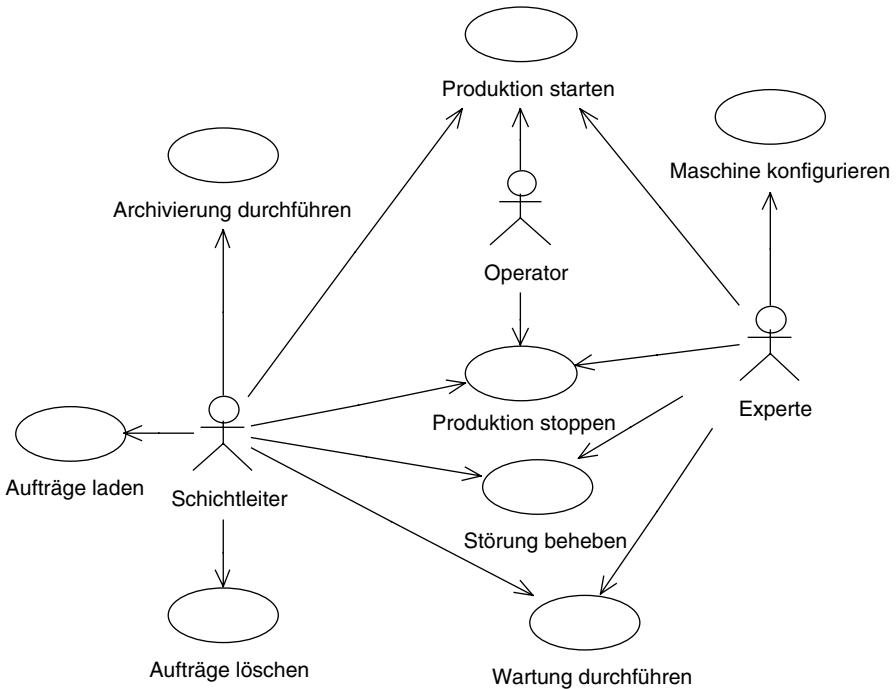


Abb. 5.4 Anwendungsfalldiagramm für das Bediensystem

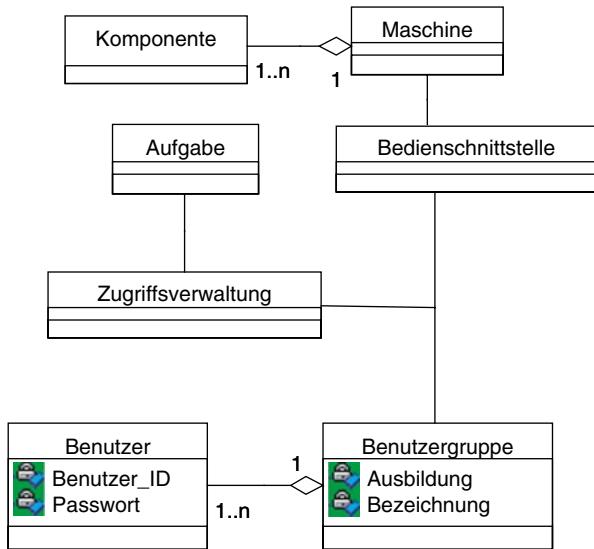
- **Klassen**

Als Klasse bezeichnet man eine Menge von Objekten vom gleichen Typ. Objekte sind dann vom gleichen Typ, wenn sie gleiche Eigenschaften (Attribute) und gleiches Verhalten (Operationen) aufweisen. Beispielsweise haben alle Objekte vom Typ „Auto“ die Attributen „Farbe“, „Baujahr“ und „Leistung“ so wie das Verhalten „bremsen“ und „hupen“. Es ist darauf zu achten, dass „Attribute“ und „Operationen“ für die Klassenzugehörigkeit entscheidend sind, nicht aber ihre Werte. Es spielt also keine Rolle, ob ein Auto Rot, Blau, Schwarz, Gelb oder Lila ist und ob es laut oder leise hupt. Es ist ein Auto. „Dasselbe“ ist ja nicht „das gleiche“.

- **Assoziationen**

In einem System können Objekte eine oder mehrere Beziehungen zueinander aufweisen. Objekte, die zueinander in Beziehung stehen, können Informationen austauschen. Dabei wird unterschieden, ob nur ein Objekt Zugriff auf Informationen des jeweils anderen Objekts (unidirektional) hat, oder ob beide Objekte auf Informationen des anderen zugreifen dürfen (bidirektional). Die Beziehung wird Assoziation genannt. Zu einer Beziehung gehört auch die Angabe der Zahl der gleichzeitig assoziierten Objekte. Bei einer 1:1 Assoziation darf nur ein Objekt von einer Klasse mit genau einem Objekt aus der anderen Klasse verbunden sein und umgekehrt, z. B. Mann und Frau in der Ehe. Bei einer 1:n Assoziation

Abb. 5.5 Vereinfachtes Klassendiagramm für das Bediensystem



darf ein Objekt auf der 1 Seite mit mehreren Objekten auf der n Seite verbunden sein aber nicht umgekehrt, z. B. Person und Auto. Hingegen dürfen Objekte bei einer n:m Assoziation mit beliebig vielen verbunden sein. Die Zahlen 1, n und m werden Kardinalität genannt. Die UML-Notation verwendet hierfür das Zeichen „*“ für „beliebig viele“.

- **Rollen**

Optional kann die Rolle eines Objektes in einer Assoziation angegeben werden. Beispielsweise spielt die Person die Rolle eines Besitzers in der Assoziation zwischen Person und Auto.

Im Abb. 5.5 wird ein vereinfachtes Klassendiagramm für das System „Benutzungs-oberfläche“ dargestellt.

Beispielsweise können folgende Aussagen dem Diagramm entnommen werden:

- Eine Benutzergruppe besteht mindestens aus einem Benutzer. Jeder Benutzer gehört zu genau einer Benutzergruppe.
- Der Zugriff auf die Bedienschnittstelle wird von den Objekten der Klasse „Zugriffsverwaltung“ verwaltet, die dann die Objekte der Klasse Aufgabe mit Benutzergruppen und Bedienschnittstellen zusammenbringt.
- Eine Maschine besteht aus mindestens einer Komponente, und jede Komponente gehört zu genau einer Maschine.

5.2.4 Aktivitätsdiagramm

Das Aktivitätsdiagramm ist ein Verhaltensdiagramm und zeigt eine bestimmte Sicht auf die dynamischen Aspekte des modellierten Systems. Ein Aktivitätsdia-

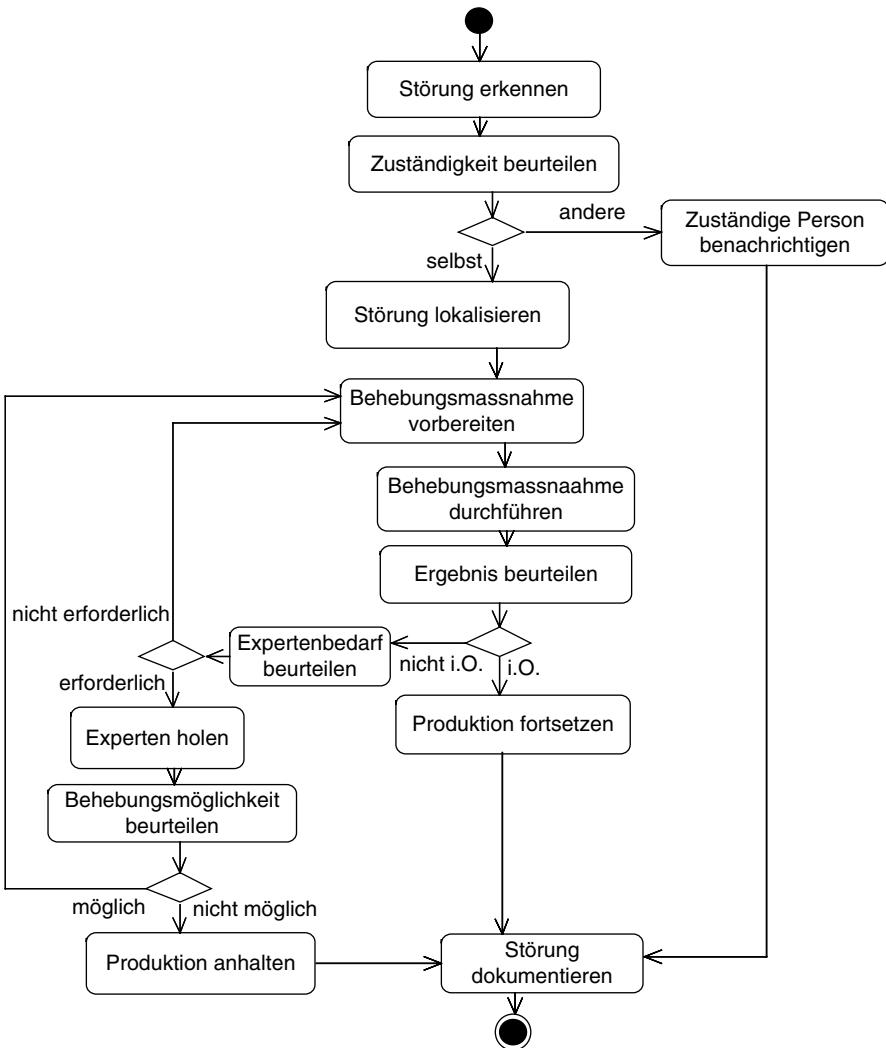


Abb. 5.6 Aktivitätsdiagramm für die Behebung einer Störung

gramm stellt die Vernetzung von elementaren Aktionen und deren Verbindungen mit Kontroll- und Datenflüssen grafisch dar. Mit einem Aktivitätsdiagramm wird meist der Ablauf eines Anwendungsfalls beschrieben. Ein Aktivitätsdiagramm stellt eine mögliche schrittweise Abwicklung der Aufgaben in Form von Aktivitäten dar. Es entspricht einem bestimmten Arbeitsszenario. Die Funktionsweise eines Aktivitätsdiagramms ist in Abb. 5.6 für die Aufgabe „Störung beheben“ erklärt. Je nach Organisation kann der Ablauf der Aufgaben abweichen.

Seit Version 2.0 der UML wurde das Aktivitätsdiagramm um eine Vielzahl von Konstrukten erweitert. Eine der größten Erweiterungen ist die Annäherung an die

Semantik von Petri-Netzen. Dadurch ist es möglich, vollständige Prozessmodellierungen mit parallelen Prozessen durchzuführen.

Die Aufgabe beginnt, in dem der Benutzer die Störung erkennt. Durch diese Erkennung kann der Benutzer dann beurteilen, ob er die Störung beheben darf oder nicht. Ist die Behebung nicht in seiner Zuständigkeit, so kann er die zuständige Person benachrichtigen und die Störung dokumentieren. Ist er aber dafür zuständig, so fängt er mit der Lokalisierung der Störung an. Dann bereitet er die Erhebungsmaßnahmen und führt sie durch. Danach erfolgt eine Überprüfung des Ergebnisses. Falls die Maßnahme erfolgreich war, kann die Produktion fortgesetzt werden. Ist es aber nicht der Fall, besteht die Möglichkeit, einen Experten zu Hilfe zu holen oder eine andere Maßnahme ohne Hilfe von Experten zu probieren. Wird ein Experte gerufen, kann dieser entscheiden, ob weitere Maßnahmen durchgeführt werden sollen oder die Produktion angehalten werden soll. Nach Bearbeitung der Störung wird diese dokumentiert.

5.2.5 Zustandsdiagramme

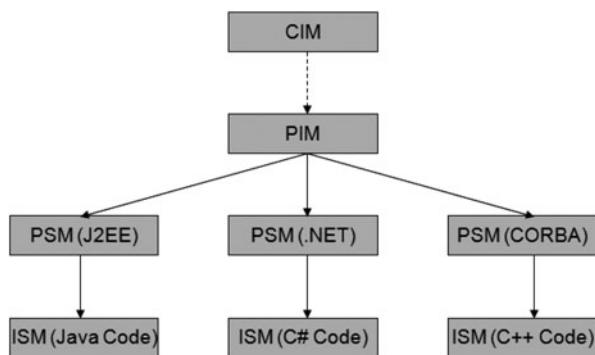
Zustandsdiagramme stellen eine andere Form eines Verhaltensdiagrammes dar uns sollen an dieser Stelle nur kurz erwähnt werden. Zustandsdiagramme werden i. d. R. einzelnen Klassen zugeordnet und beschreiben das dynamische Verhalten der jeweiligen Objekte. Ein Zustandsdiagramm zeigt eine Folge von Zuständen, die ein Objekt während seines Lebenszyklus annehmen kann und darüber hinaus aufgrund welcher Ereignisse Zustandsänderungen stattfinden können. Es wird nicht jede Änderung eines Attributwertes als Zustandsänderung angesehen. Die Abstraktion besteht vielmehr darin, nur solche Ereignisse zu berücksichtigen, die das Verhalten des Objekts maßgeblich beeinflussen. Der Zustand muss also nach außen sichtbar oder erkennbar sein.

Seit 2005 wird die Sprache State Chart XML (SCXML) von der W3C entwickelt. SCXML ist eine XML basierte Sprache zur Beschreibung von Zustandsdiagrammen. Insbesondere ist geplant, dass SCXML als Kontrollmodell im Rahmen von sprachbasierten Benutzungsschnittstellen in Verbindung mit VoiceXML eingesetzt wird. In anderen Forschungsarbeiten wird SCXML aber u. a. auch für die Steuerung in digitalen Spielen oder für multimodale Benutzungsschnittstellen eingesetzt.

5.3 Model-Driven Architecture (MDA)

Mit der Model Driven Architecture (MDA) liegt nun ein Standard der Object Management Group (OMG) vor, der die Repräsentation von Software von der Programmcodeebene auf die Modellebene heben möchte. Um die Komplexität auf Modellebene zu reduzieren, werden Modelle und Plattformen verschiedener Abstraktionsebenen unterschieden, die aufeinander aufbauen. Die Grundidee des Ansatzes

Abb. 5.7 Transformationen der MDA



ist nicht neu, sondern folgt dem Grundsatz, dass die Spezifikation einer Komponente unabhängig von der technischen Umsetzung zu beschreiben ist. Neu ist, dass mit der MDA aufeinander abgestimmte Spezifikationen und ein Konzept für die modellgetriebene Softwareentwicklung existieren. Dabei kann die Realisierung eines plattformunabhängigen Modells durch die Wahl einer Plattform, also der konkreten technischen Umsetzung, teilweise oder vollständig automatisiert erfolgen. Mit der MDA wird das Modell zum zentralen Element des Softwareentwicklungsprozesses. Die verschiedenen Modelle der MDA werden im Folgenden kurz erläutert.

Das **Computation Independent Model (CIM)** liefert eine Sicht auf das Gesamtsystem, die völlig unabhängig davon ist, wie es implementiert wird. Das Modell wird im Vokabular seiner Domäne beschrieben und betont die Anforderungen an das System und seine Umwelt. Da ein CIM oftmals einen nur geringen Formalisierungsgrad aufweist, ist es für die nachfolgenden automatisierten Transformationen und die Codegenerierung kaum von Bedeutung.

Mittels des **Platform Independent Model (PIM)** werden Funktionalitäten unabhängig von der verwendeten Plattform modelliert. Mit Hilfe des PIM lassen sich somit Teile eines Systems modellieren, ohne die endgültige Zielplattform zu kennen. Aufgrund der plattformunabhängigen Beschreibung enthält es nicht alle für die Implementierung auf einer bestimmten Plattform notwendigen Details.

Das **Platform Specific Model (PSM)** ist ein plattformspezifisches (konkreteres) Modell, welches nicht nur die fachlichen Anforderungen, sondern auch plattformspezifische Informationen enthält.

Das **Implementation Specific Model (ISM)** berücksichtigt die spezifischen Anforderungen für die Implementierung des Systems auf einer bestimmten Plattform und wird durch Quellcode repräsentiert.

Wie Abb. 5.7 zeigt, sieht die MDA z. B. vor, PIMs in PSMs und PSMs in ISMs zu transformieren. In Klammern werden jeweils Beispiele der entsprechenden Modelle dargestellt.

Eine solche Transformation kann manuell, semiautomatisch oder vollautomatisch erfolgen. Jede Transformation besteht aus der Anwendung von Transformationsregeln, sogenannten Mappings, die für die (semi-) automatische Ausführung formal definiert sind. Die Transformation reichert die Information z. B. des PIMs durch zusätzliche Information der ausgewählten Plattform und der verwendeten

Transformationsregeln an. Neben statischen Erweiterungen kann hier auch technisches, durch die Wahl der Plattform motiviertes, dynamisches Verhalten hinzukommen.

Bisher dienten Modelle in klassischen Entwicklungsprozessen lediglich zur Dokumentation in Handbüchern und zur Kommunikation zwischen den beteiligten Personen. Das Ziel der MDA ist neben der Verbesserung der Portierbarkeit und Wiederverwendung von Modellen auch die Reduktion der Kosten bei der Entwicklung von Software.

Der Kerngedanke der MDA, abstrakte Modelle in konkretere Modelle und schließlich in Quellcode (teil-)automatisiert transformieren zu können, ist somit von großer Bedeutung. Eines der wichtigsten Kriterien bei der erfolgreichen Anwendung der MDA ist die Unterstützung der Prozesse durch Werkzeuge. Aktuell können aber lediglich unvollständige Modelle und Quelltextfragmente mittels Transformationen der MDA erstellt werden. Einer der Hauptgründe liegt in der fehlenden Werkzeugunterstützung bei der Transformation der Modelle.

5.4 Modellbasierte Useware-Entwicklung

Die stetig wachsende Zahl verschiedenster (mobiler) Geräte stellt eine Herausforderung für Entwickler dar. Es muss nicht nur eine Applikation bzw. eine Benutzungsschnittstelle für eine Plattform, sondern eine Applikation mit einer Vielzahl von Benutzungsschnittstellen (mit verschiedenen Bildschirmgrößen, Grafikbibliotheken, Ressourcen, etc.) für verschiedene Geräte entwickelt werden. Zum einen ist es äußerst schwierig, definierte Anforderungen zu erfüllen und gleichzeitig, über mehrere Plattformen hinweg, konsistent in Bezug auf Funktionalität bzw. Visualisierung zu sein und zum anderen impliziert die Entwicklung von Anwendungen für verschiedene Plattformen auch, dass Entwickler eine Vielzahl von Programmiersprachen und Methoden beherrschen müssen.

Abbildung 5.8 zeigt, dass aktuelle Entwicklungsmethoden für Benutzungsschnittstellen für jede einzelne Plattform (z. B. Handy, PDA, Desktop PC) eine separate Benutzungsschnittstelle (z. B. in Java, HTML, WAP) entwickeln.

Bei einer Änderung müssen somit Anpassungen auf mehreren Plattformen parallel durchgeführt werden, was einen großen zeitlichen, personellen und somit auch wirtschaftlichen Aufwand bedeutet. Auch ist es nicht möglich von einer Plattform, z. B. HTML, auf eine andere, z. B. Java, zu migrieren. Aufgrund der beschriebenen Komplexität aktueller Entwicklungen von Benutzungsschnittstellen wird im restlichen Abschnitt dieses Kapitels das Konzept der modellbasierten Entwicklung von Benutzungsschnittstellen (engl. „Model-Based User Interface Development“, MBUID) vorgestellt.

MBUID unterstützt Entwickler bei der Entwicklung von Benutzungsschnittstellen durch eine systematische Vorgehensweise, die bspw. an den Useware-Engineering Prozess angelehnt sein kann (Hußmann, Meixner, Zühlke 2011). Um dies zu ermöglichen, werden Benutzungsschnittstellen mit deklarativen Sprachen, meist als XML-Anwendung, spezifiziert.

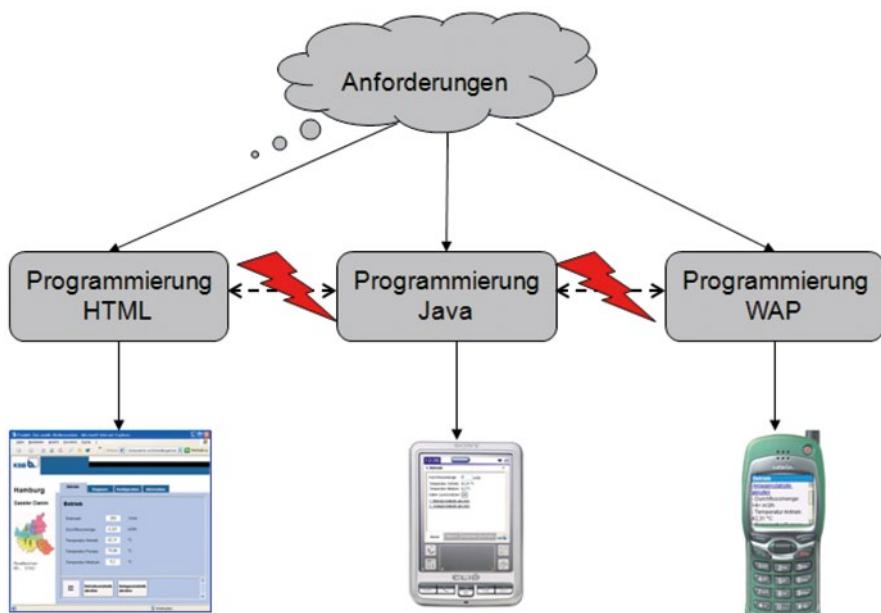


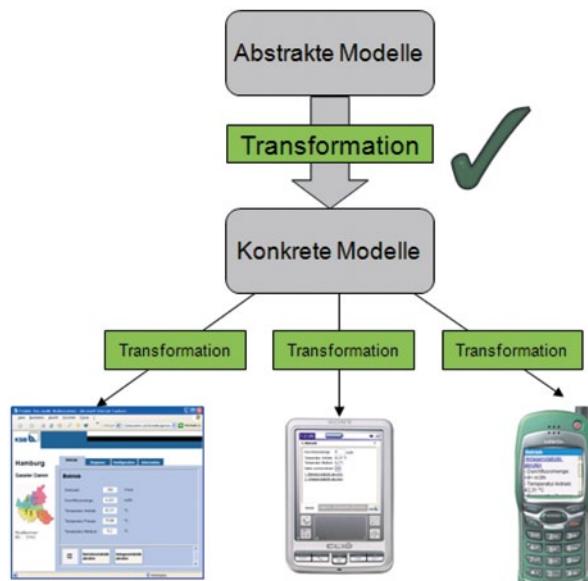
Abb. 5.8 Bisherige Entwicklung einer multiplattform Benutzungsschnittstelle

Vorteile des Einsatzes von XML zur Modellierung und Entwicklung der Benutzungsschnittstelle sind bspw.

- Möglichkeit zur generischen Beschreibung
- Unterstützung der systematischen Entwicklung
- Klare Trennung zwischen Applikations- und Benutzungsschnittstellenentwicklung
- Unterstützung der Entwicklung für multiplattformartige oder multimodale Benutzungsschnittstellen
- Möglichkeit zur (semi-) automatisierten Generierung (von Teilen) der Benutzungsschnittstelle.

Die Vorgehensweise des MBUID entspricht weitgehend den modellbasierten Ansätzen, die bei der Entwicklung der Applikationslogik von Software verwendet werden. Ansätze, wie z. B. durchgängige Transformation von Modellen, die im Rahmen der modellgetriebenen Entwicklung von Software (MDA) verfolgt werden, sind auch bei MBUID ein wichtiges und zentrales Element, um den Entwicklungsprozess effizienter zu gestalten. Das Paradigma empfiehlt einen Top-Down-Ansatz, beginnend mit abstrakten Beschreibungen relevanter Informationen über die Benutzungsschnittstelle. Diese abstrakten Beschreibungen werden in konkretere Modelle transformiert, um eine partielle oder vollständige Generierung von Quellcode zu erreichen (s. Abb. 5.9). Somit ist es nicht mehr erforderlich, für jede Zielplattform (s. Abb. 5.8) separaten Quellcode zu entwickeln.

Abb. 5.9 Entwicklung einer multiplattform Benutzungsschnittstelle beim MBUID



Die folgenden Unterkapitel zeigen anhand einer modellbasierten Architektur für Benutzungsschnittstellen das Potenzial dieser noch jungen Methodik auf und führen in entsprechende Hilfsmittel wie bspw. Sprachen und anwendungsnahe Werkzeuge ein. Dabei lehnt sich die hier vorgestellte Architektur an den bereits vorgestellten Useware-Entwicklungsprozess an und gewährleistet somit eine durchgängige nutzerzentrierte Entwicklung der Benutzungsschnittstelle.

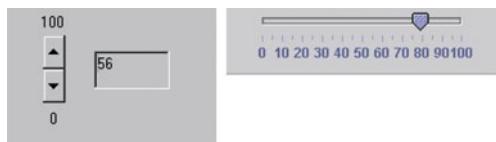
5.4.1 Kernmodelle

Im Rahmen eines modellbasierten Ansatzes können eine Reihe verschiedener Modelle verwendet werden. Die wichtigsten Kernmodelle sind:

Aufgabenmodell: Im Rahmen eines modellbasierten nutzer- und aufgabenzentrierten Entwicklungsprozesses für Benutzungsschnittstellen haben Aufgabenmodelle eine besonders wichtige Bedeutung. Aufgabenmodelle sind eine explizite Repräsentation von Nutzeraufgaben, die während der Aufgabenanalyse und -auswertung erstellt und eingesetzt werden. Das Aufgabenmodell ist ein (abstraktes) Modell, um Aufgaben hierarchisch zu strukturieren und die Abfolge der Aufgaben zeitlich mittels Temporaloperatoren in Relation zu setzen.

Es gibt eine generelle Übereinstimmung, dass jede Person eine eigene hierarchische Abbildung der auszuführenden Aufgabe mental modelliert. Der hierarchische Aspekt einer Aufgabe hat seine Ursache somit im Menschen selbst. Aufgrund end-

Abb. 5.10 Beispiel eines Zählers (*links*) und eines Schiebereglers (*rechts*)



licher Leistungsfähigkeit des Gedächtnisses ist eine Strukturierung von Aufgaben zwingend erforderlich. Konsequenterweise werden komplexe Aufgaben in weniger komplexe Unteraufgaben aufgeteilt.

Dialogmodell: Das Dialogmodell ist das Bindeglied zwischen Aufgaben- und Präsentationsmodell. Es beschreibt den Ablauf, den Startpunkt, den Zielpunkt und die Ereignisse (z. B. Drücken eines Knopfes), die den Ablauf an der Benutzungsschnittstelle steuern und stellt das dynamische Navigationsmodell einer Benutzungsschnittstelle dar. Das Dialogmodell beschreibt den Ablauf einer Benutzungsschnittstelle ohne eine konkrete Technik bzw. „Look&Feel“ zu berücksichtigen. Es fungiert somit als Zwischenmodell und bildet eine andere Sicht auf das Aufgabenmodell ab. Es ist aber nicht autark, sondern von anderen Modellen, insbesondere dem Präsentationsmodell, abhängig.

Präsentationsmodell: Die Darstellung der Benutzungsschnittstelle wird durch das Präsentationsmodell definiert. Es spezifiziert, wie visuelle, haptische oder auditive Interaktionsobjekte (engl.: widgets) an der Benutzungsschnittstelle dargestellt werden. Dabei kann die Interaktion z. B. über eine grafische Oberfläche eines PCs oder durch Sprachsteuerung mit einem Smartphone stattfinden. Die Art und Weise, wie z. B. eine Benutzungsschnittstelle repräsentiert wird, hängt von den Bedingungen des Ausgabemediums ab (u. a. Displaygröße, Farbdarstellung, etc.). Das Präsentationsmodell ist die statische und das Dialogmodell die dynamische Komponente einer Benutzungsschnittstelle. Dabei setzt sich das Präsentationsmodell aus dem abstrakten und konkreten Präsentationsmodell zusammen. Das abstrakte Präsentationsmodell besteht typischerweise aus einer Menge abstrakter Interaktionsobjekte (engl.: AIO), die in einem weiteren Transformationsschritt auf korrespondierende konkrete Interaktionsobjekte abgebildet (engl.: CIO) werden. Die Gesamtheit der AIO's ergeben das abstrakte und die Gesamtheit der CIO's das konkrete Präsentationsmodell. Ein CIO ist demnach die konkrete Darstellung eines Objektes. Abbildung 5.10 zeigt zwei verschiedene Formen eines AIO „Helligkeitsregler für Bildschirme“. Der Helligkeitsregler kann zum einen als CIO „Zähler“ (linker Teil der Abbildung), zum anderen als CIO „Schieberegler“ (rechter Teil der Abbildung) dargestellt werden. Des Weiteren könnte ein AIO „Helligkeitsregler für Bildschirme“ mit einer Spracheingabe bzw. Sprachausgabe als weiteres CIO realisiert werden.

Das Präsentationsmodell ist im Gegensatz zum Dialogmodell ein autarkes Modell und kann unabhängig von anderen Modellen in einem Entwicklungsprozess genutzt werden. Sprachen zur Beschreibung von Präsentationsmodellen sind z. B. die User Interface Markup Language (UIML), die eXtensible Interface Markup Language (XIML) oder die Extensible User Interface Language (XUL). Diese Spra-

chen bilden zur Beschreibung der Benutzungsschnittstelle eine gute Grundlage, da mit Hilfe dieser Sprachen z. B. eine Trennung von Navigationsstruktur, Design, Dialogelementen, dynamischem Verhalten und Lokalisierung erreicht wird. Mit den meisten der o. g. Sprachen ist es möglich neben dem Präsentations- auch das Dialogmodell zu beschreiben.

5.4.2 Modellbasierte Architektur

Damit, basierend auf verschiedenen Modellen, die Entwicklung von modernen und flexiblen Benutzungsschnittstellen durchgeführt werden kann, müssen strukturierte Entwicklungsprozesse und Architekturen definiert werden. Ziel strukturierter Entwicklungsprozesse und Architekturen ist es, den Entwicklern einen Rahmen zur Unterstützung der modellbasierten Entwicklung von Benutzungsschnittstellen zur Verfügung zu stellen.

Durch die Verwendung von Aufgabenmodellen wird der Nutzer der zukünftigen Benutzungsschnittstelle in den Mittelpunkt der Entwicklung der Benutzungsschnittstelle gerückt. Die weiteren verwendeten Modelle stellen sicher, dass alle Anforderungen an die Benutzungsschnittstelle erhoben und bei der Entwicklung bedacht werden. Werden Anforderungen geändert, impliziert dies lediglich eine Änderung eines oder mehrerer Modelle, aber nicht des gesamten Entwicklungsprozesses.

Ein weiterer Vorteil ist die Möglichkeit der frühzeitigen Erstellung von Prototypen, da die Benutzungsschnittstelle aus den verschiedenen Modellen der Entwicklungsumgebung des MBUID heraus generiert werden kann. Die Trennung des Entwicklers der Applikationslogik und des Designers der Benutzungsschnittstelle wird bei aktuellen Architekturen hervorgehoben. Somit ist es dem Designer möglich, unabhängig vom Applikationsentwickler zu arbeiten. Abbildung 5.11 visualisiert eine mögliche Architektur für die modellbasierte Entwicklung von Benutzungsschnittstellen. Auf Teile der in Abb. 5.11 genannten Notationen und Tools wird in den folgenden Kapiteln näher eingegangen.

5.4.3 Notationen und Tools für die Analyse

Um einen Überblick über die Vielzahl an anfallenden Daten im Useware-Entwicklungsprozess zu wahren, wird mit dem Analysewerkzeug eine strukturierte und EDV-basierte Datenhaltung und Auswertung ermöglicht (Bödcher 2007). Das Analysewerkzeug setzt sich aus einer MySQL-Datenbank und einer Benutzungsschnittstelle zusammen. Es wird entlang eines strukturierten Vorgehens innerhalb der Analysephase des Useware-Engineering Prozesses verwendet, das aus den Phasen Vorbereitung, Durchführung und Auswertung besteht.

Zentrales Planungselement des Analysewerkzeugs in der Vorbereitungsphase ist ein generischer Anforderungskatalog mit 36 vordefinierten Datenkategorien,

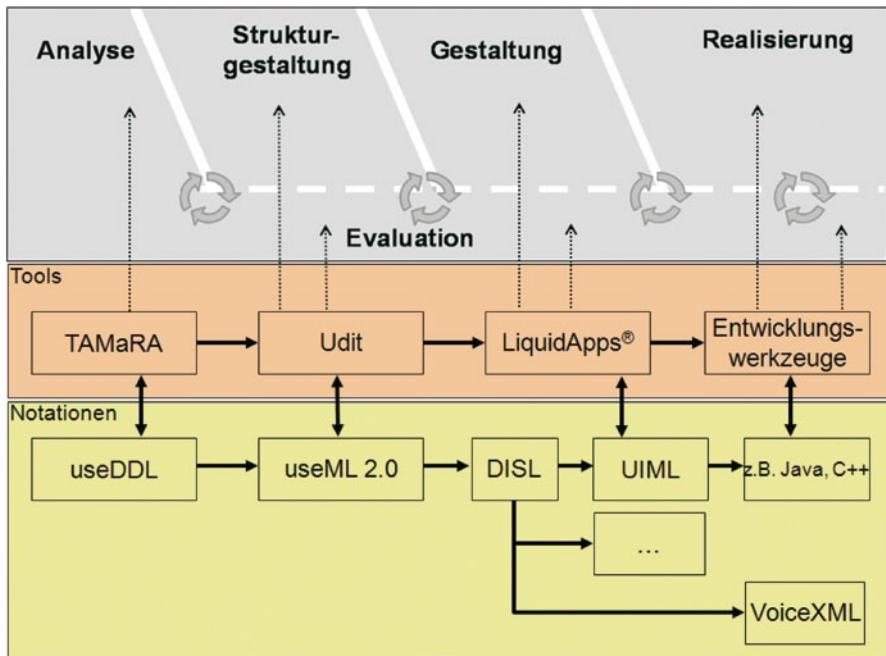


Abb. 5.11 Architektur des modellbasierten Useware-Engineering Prozesses nach. (Meixner 2010)

der in seiner Grundform zu Beginn der Planungsphase durch das Analysewerkzeug zur Verfügung gestellt wird. Er ist in seiner Gesamtheit jederzeit zugreifbar und bildet die Basis jedes Datensatzes (d. h. jeder Einzelerhebung). Außerdem erhöht die Strukturierung die Übersichtlichkeit und erlaubt einen schnellen Zugriff auf einzelne Datenkategorien bzw. Gruppen von Datensätzen. Die Spezifikation von Datenkategorien ermöglicht die Entscheidung, in welcher Form Daten letztlich zur Verfügung stehen. Neben der Erfassung quantitativer Daten (Statistiken, etc.) sowie qualitativer Daten (z. B. individuelle Äußerungen in Nutzerinterviews) können auch modellbasierter bzw. allgemein strukturierter Daten (Aufgaben- & Benutzungsmodelle) gespeichert werden. Die Dokumentation der Daten erfolgt direkt durch die Eintragung in das Analysewerkzeug und damit parallel zur Erhebung. Ausgewertete und ggf. angereicherte qualitative und quantitative Daten können in gängige Microsoft-Office-Komponenten (z. B. PowerPoint, Word) exportiert werden. Modellbasierte Daten können in Form eines minimalisierten Benutzungsmodells in die Sprache useML exportiert werden.

Das Analysewerkzeug (s. Abb. 5.12 und Abb. 5.13) bewährte sich im realen Projekteinsatz; erwartungsgemäß wurden aber zusätzliche Anforderungen ermittelt und seine Erweiterung etwa um autarke Datenhaltungsmechanismen (ohne Zugriff auf vorinstallierte Datenbanksysteme) sowie um Multilingualität (speziell eine englische Version der Software) erforderlich.

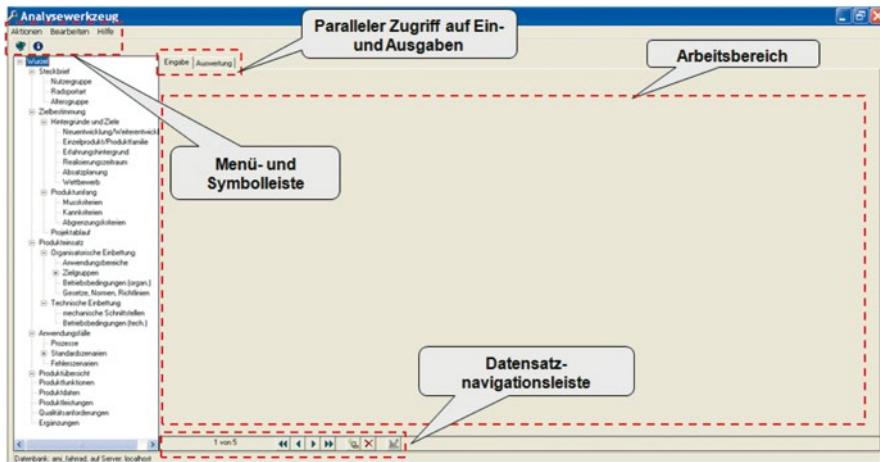


Abb. 5.12 Screenshot mit Überblick über das Analysewerkzeug

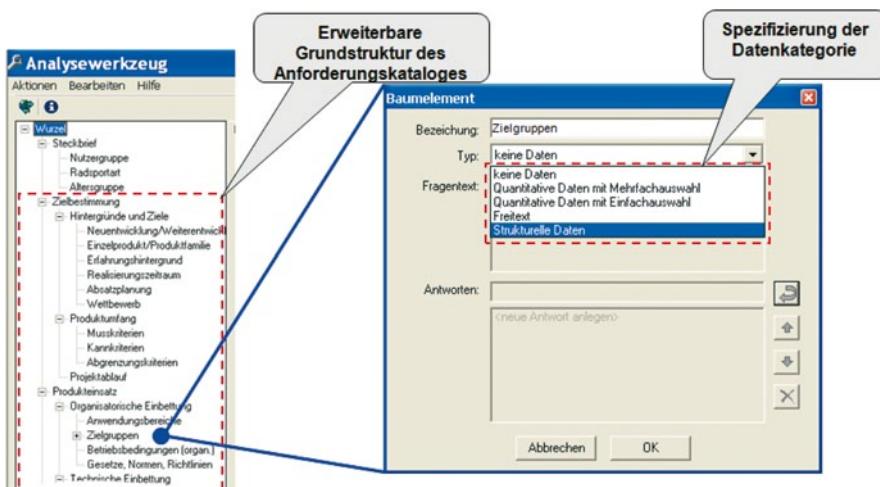


Abb. 5.13 Screenshot mit Überblick den Anforderungskatalog

Der „Tasks, Activities, Models and Requirements Analyzer“ (TAMaRA) stellt somit eine logische Weiterentwicklung des ursprünglichen Analysewerkzeugs dar, wartet aber mit einer Vielzahl neuer Funktionen auf. Um TAMaRA (s. Abb. 5.14) einfach bedienbar zu halten, wurden zunächst die Funktionen zur Bearbeitung des Anforderungskataloges in ein separates Programm, den „Catalog Editor“ (CatE), ausgelagert (s. Abb. 5.15). Er gestattet das Anlegen und Bearbeiten eines Kataloges vor Durchführung der Analyse, garantiert aber auch bei nachträglicher Änderung des Kataloges während und nach der Analysephase die Integrität aller Datensätze.

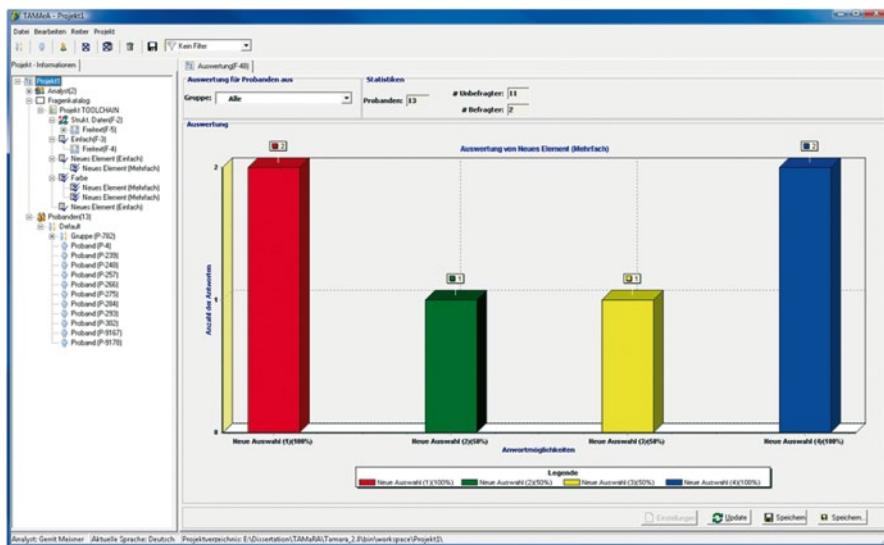


Abb. 5.14 Screenshot von TAMaRA

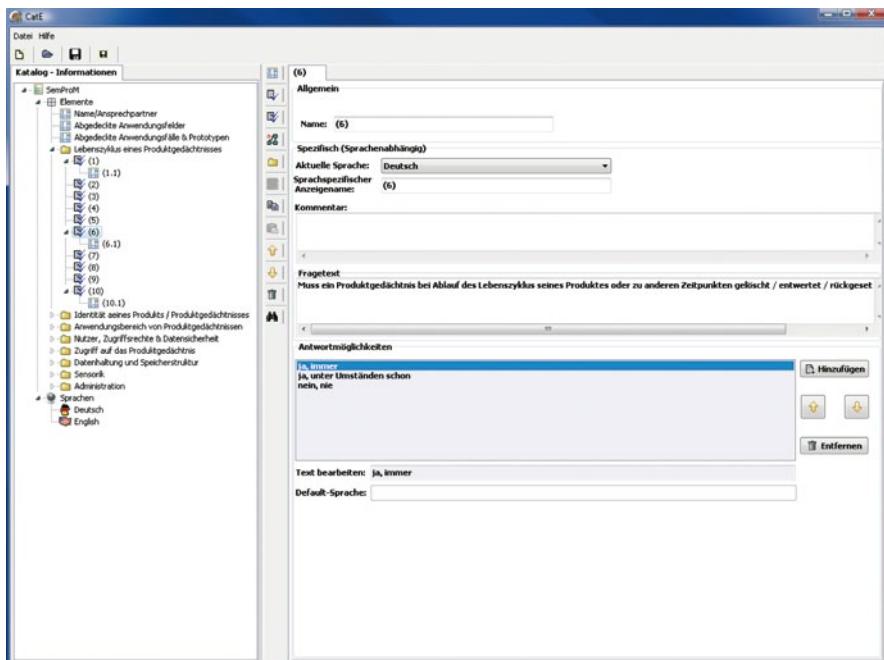


Abb. 5.15 Screenshot von CatE

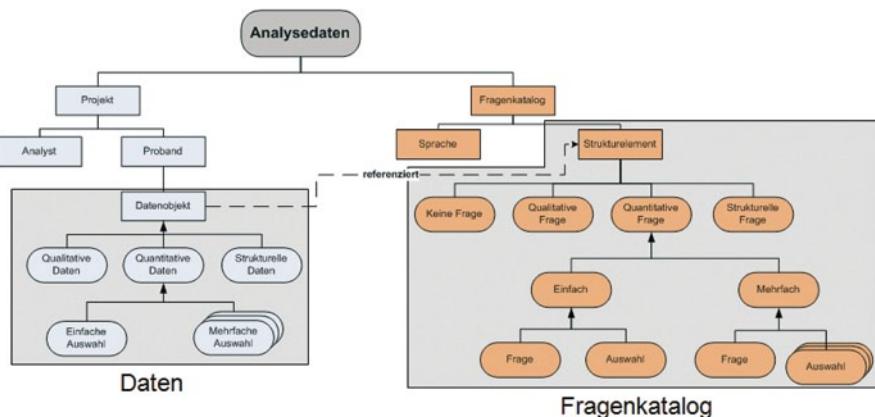


Abb. 5.16 Datenstruktur der Useware Data Description Language (useDDL)

Während der Durchführung der Analyse erlaubt TAMaRA die rollenbasierte Unterscheidung der beteiligten Analysten, so dass jederzeit nachvollzogen werden kann, welcher Analyst welche Daten erhoben und eingegeben hat. Dabei werden neben den eigentlichen Analysedaten auch Daten über das Projekt selbst evaluiert, beispielsweise die Anzahl der interviewten Probanden je Analyst. Auf die eigentlich während der Analyse erhobenen Daten kann TAMaRA während der Eingabe wie auch der Auswertung verschiedene Filter anwenden, etwa um ausschließlich als Freitext eingegebene Daten (aus Interviews) oder strukturierte Daten (z. B. individuelle Aufgabenmodelle) anzuzeigen. Diese Unterteilungen und Filterfunktionen ermöglichen eine wesentlich einfachere Bedienung der Software sowie das schnelle Einsortieren und Auffinden von Daten. Gespeichert werden alle Daten in einem speziell entwickelten XML-Dialekt, der „Useware Data Description Language“ (useDDL, s. Abb. 5.16). Die daraus entstehende XML-Datei ist plattformunabhängig und kann problemlos weitergegeben (kopiert) werden; TAMaRA selbst ist unter Windows mit dem .NET-Framework ohne vorangehende Installation lauffähig.

5.4.4 Notationen und Tools für die Strukturgestaltung

Die UML-Diagrammtypen können zwar für vielfältige Belange des Entwurfs eingesetzt werden, allerdings gibt es gerade beim Useware-Engineering Defizite bei der Modellierung von Nutzeraufgaben (Meixner 2010). So fehlt die Möglichkeit, die Nutzer selbst bzw. ihr Verhalten bei der Aufgabenbearbeitung zu beschreiben. Ebenso fehlt ein Konzept zur Modellierung des Nutzungskontextes, in dem eine Interaktion stattfindet. UML ist also nur bedingt für Einsatzfälle geeignet, die über eine technikzentrierte Darstellung hinausgehen.

Das im Folgenden dargestellte Konzept der Useware Markup Language (useML) zeigt auf, wie Nutzeraufgaben als Grundlage für das Useware-Engineering genutzt

werden und wie der XML-Ansatz die prototypische Aufbereitung unterschiedlicher Ziel-Plattformen und so die konsequente Nutzerintegration ermöglicht.

Die **Useware Markup Language (useML)** wurde in Version 1.0 zur aufgaben- und nutzerzentrierten Entwicklung des Useware-Entwicklungsprozesses entwickelt (Reuther 2003). Die Umsetzung des Benutzungsmodells wurde mittels der Sprache useML durchgeführt, was als XML-Anwendung entwickelt wurde. Mittels useML ist es somit möglich Benutzungsmodelle zu definieren und auf einem Datenträger zu speichern.

Aufgrund der durchgeführten Erweiterung des Benutzungsmodells (s. Kap. 3.3.2) wurde die Sprache useML entsprechend angepasst und erweitert sowie mit der neuen Versionsnummer 2.0 versehen (Meixner 2010).

Zur einfachen Bearbeitung von Benutzungsmodellen der Sprache useML wurde ein grafischer useML-Editor (Udit) in VB.NET entwickelt. Mittels Udit ist der Entwickler in der Lage, Benutzungsmodelle grafisch zu erstellen und zu bearbeiten. Udit verfügt über einen integrierten Validierungsmechanismus, d. h. wenn ein useML-Dokument geöffnet wird, so wird das Dokument gegen das XML-Schema von useML 2.0 validiert. Ist das useML-Dokument nicht valide, werden dem Entwickler entsprechende Fehlermeldungen bzw. Hinweise angezeigt. UseML-Dokumente, die mit Udit erstellt und gespeichert werden, sind immer zur useML 2.0-Spezifikation valide. Aufgrund der vorhandenen Möglichkeit, useML an verschiedene projektspezifische Randbedingungen, wie z. B. unterschiedliche Benutzergruppen, vordefinierte Personas, verschiedene Zugriffsorte, mehrere Gerätetypen oder bestimmte Funktionsmodelle anzupassen, wurde ebenfalls ein Schema-Editor integriert, der diese Randbedingungen bearbeiten kann.

Darüber hinaus wurde Udit um weitere nützliche Funktionen erweitert, z. B. Ausdruck des Benutzungsmodells (zur Strukturevaluation) oder Speichern des Benutzungsmodells als Grafik (zur Dokumentation). Damit Udit auch international eingesetzt werden kann, wurde Udit multilingual konzipiert und entwickelt. Aktuell sind die Sprachen Deutsch und Englisch implementiert. Weitere Sprachen können bei Bedarf durch einfache interne Mechanismen integriert werden. Wie in Abb. 5.17 zu erkennen ist, werden die Elemente der useML 2.0-Spezifikation (Benutzungsmodell, Benutzungsobjekt, elementare Benutzungsobjekte) verschiedenfarbig dargestellt. Dies erleichtert – gerade bei größeren Benutzungsmodellen – die Orientierung innerhalb des Modells. Auch ist in Abb. 5.17 zu erkennen, dass Teilbäume des Benutzungsmodells eingeklappt werden können. Die Temporaloperatoren werden innerhalb der Verbindungslien der jeweiligen Objekte angezeigt. Udit wurde speziell entwickelt, um alle Ausdrucksmöglichkeiten der überarbeiteten useML 2.0-Spezifikation optimal zu unterstützen. Des Weiteren verfügt Udit über eine vollwertige Drag&Drop-Funktionalität.

Ein weiteres wichtiges Hilfsmittel zur Evaluierung von Benutzungsmodellen ist deren Simulation. In einer Simulation kann geprüft werden, ob das Benutzungsmodell sich so verhält wie es der Entwickler geplant hatte, bevor das System weiter entwickelt wird. Auf Basis der Struktur des Benutzungsmodells und dessen Temporaloperatoren ist es mit Udit möglich, Benutzungsmodelle zu simulieren (Abb. 5.18). Entwickler erhalten somit ein effizientes Werkzeug zur Simulation und Evaluierung

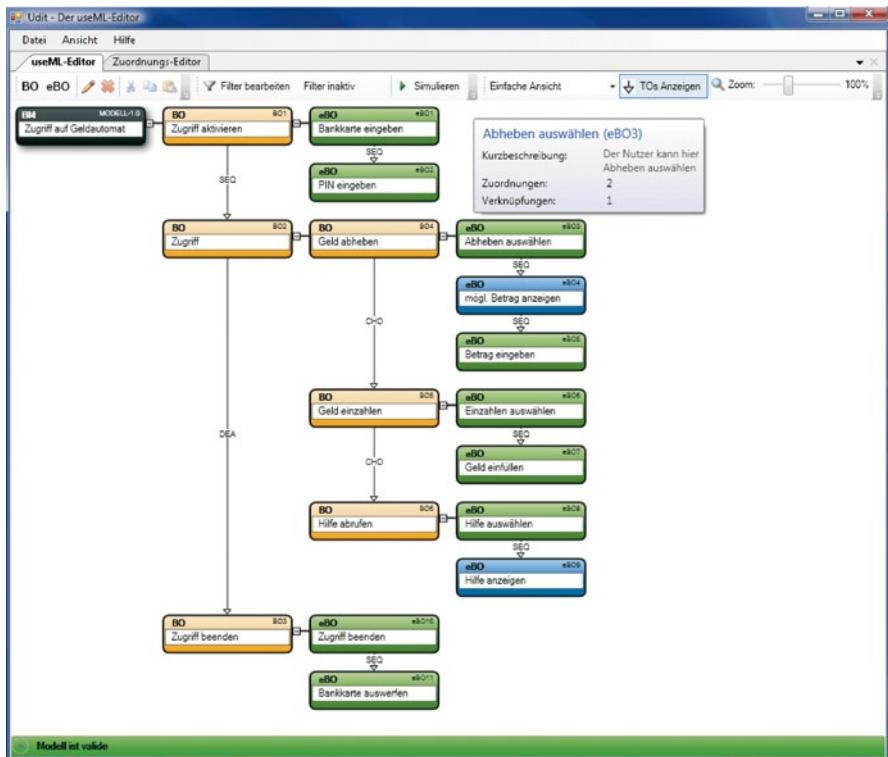


Abb. 5.17 Screenshot von Udit

der Benutzungsmodelle. Benutzungsmodelle können mit Udit in ein Präsentations- und Dialogmodell exportiert und von weiteren Werkzeugen bearbeitet werden.

5.4.5 Notationen und Tools für die Gestaltung

Die Phase der Gestaltung teilt sich in eine abstrakte und eine konkrete Modellierungsphase der Benutzungsschnittstelle auf.

Die abstrakte Benutzungsschnittstelle wird durch das **abstrakte Präsentations- und Dialogmodell** gebildet. Zur Beschreibung des abstrakten Präsentations- und Dialogmodells wird die etablierte **Dialog and Interaction Specification Language (DISL)** verwendet, die an der Universität Paderborn entwickelt wurde (Müller 2004). Mit DISL ist es möglich, plattform- und modalitätenunabhängige Benutzungsschnittstellen zu modellieren. Treibende Faktoren bei der Entwicklung der Sprache waren Skalierbarkeit, Reaktivität und niedrige Anforderungen an Prozessorgeschwindigkeit und Speichergröße. Um den Anforderungen im Bezug auf niedrige Rechengeschwindigkeit und Speicherbedarf gerecht zu werden, wurde für

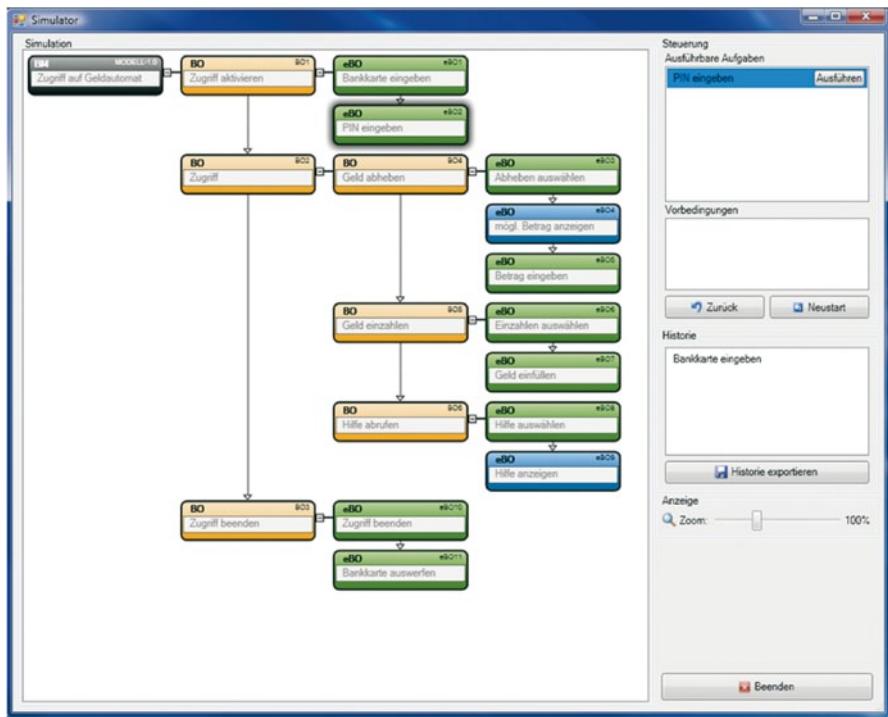


Abb. 5.18 Screenshot des in Udit integrierten Simulators

DISL zusätzlich ein serialisiertes und komprimiertes Format entwickelt, welches durch eine einfache Transformationskette erzeugt wird und sogar erfolgreich mit Smart-Cards als Träger einer gesicherten Benutzungsschnittstelle für sicherheitskritische Anwendungen verwendet wurde (Schäfer et al. 2007). Dadurch ist es mit geeigneten Laufzeitumgebungen (Interpreter) sogar direkt möglich, DISL bspw. auf kleinsten mobilen Endgeräten direkt auszuführen. Durch die Umsetzung dieser Anforderungen ist die Sprache DISL für zukünftige multimodale Benutzungsschnittstellen besonders geeignet (Meixner et al. 2008). Eine wichtige softwaretechnische Voraussetzung für die systematische Entwicklung von Benutzungsschnittstellen ist die Trennung von Struktur, Präsentation und Verhalten einer Benutzungsschnittstelle. DISL unterteilt sich daher in Struktur, Stil und Verhalten (s. Abb. 5.19). Mit einem optionalen Head-Element können zusätzliche modellspezifische Informationen in einem DISL-Dokument beschrieben werden.

Die Strukturbeschreibung dient der Hierarchiebildung zwischen den generischen Interaktionsobjekten. Aufbauend auf dieser Hierarchie kann anschließend die Struktur der Benutzungsschnittstelle einer weiterführenden konkreten Benutzungsschnittstelle erzeugt werden. Die Struktur wird dabei im <structure>-Element eines DISL-Dokumentes definiert, wobei jedes Interaktionsobjekt durch <widget>-Elemente repräsentiert wird. Ein <widget>-Element besitzt eine eindeutige ID sowie den Typ des Interaktionsobjekts, den es repräsentiert. Durch die Schachtelung von <widget>-Elementen lassen sich hierarchische Baum-Strukturen bilden. Dabei wir-

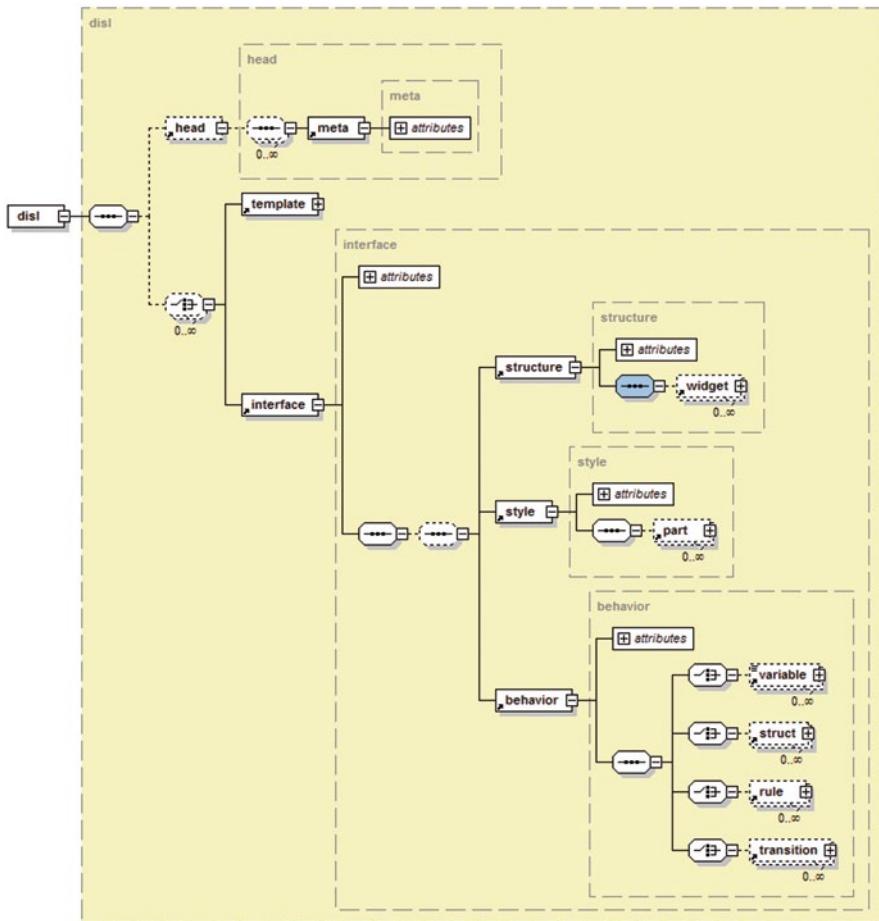


Abb. 5.19 Aufbau eines DISL Dokuments

ken sich die Hierarchien nicht auf Positionierung und Darstellung der widgets aus, sondern dienen der Festlegung der Interpretationsreihenfolge, was u. a. eine serielle Verarbeitung eines Dokuments ermöglicht (Optimierung der Speichergröße von DISL-Dokumenten). In der Stilbeschreibung werden die Eigenschaften der generischen Interaktionselemente definiert. Die Stilbeschreibung wird im `<style>`-Element definiert, wobei die Eigenschaften eines jeden generischen Interaktionselements innerhalb eines `<part>`-Elements spezifiziert werden. Jedes `<part>`-Element besitzt ein eindeutiges Attribut `generic-widget`, welches die Zuordnung der Eigenschaften zu einem `<widget>`-Element in der Strukturbeschreibung ermöglicht. Die Zuordnung wird durch das `id`-Attribut des `<widget>`-Elements und das `generic-widget-attribut` des `<part>`-Elements eindeutig festgelegt.

Zur Beschreibung des abstrakten Dialogmodells (Verhalten) übernimmt DISL das Konzept der Dialogue Specification Notation (DSN). Die Beschreibung des

Tab. 5.5 Die abstrakten Interaktionsobjekte in DISL

Ausgabeelemente	Interaktionselemente	Kollektionselemente
Variablefield textfield	Command confirmation variablebox textbox	Widgetlist choicegroup

logischen Dialogverhaltens befindet sich im <behaviour>-Element des DISL-Dokuments. Dabei setzt sich die Verhaltensbeschreibung aus drei Grundelementen: Regeln (<rule>), Transitionen (<transition>) und Ereignisse (<event>) zusammen. Zusätzlich bietet das Modell die Verwendung von Zustandsvariablen (<variable>), die zur Speicherung von Werten verwendet werden können. Dabei ist es bspw. möglich, einzelne Zustände der Benutzungsschnittstelle, wie „Der Menüpunkt A ist ausgewählt und der Auswahlschalter steht auf C“, abzuspeichern, um sie später bei einem bestimmten Ereignis auswerten zu können.

Für die Beschreibung des abstrakten Präsentationsmodells existieren in DISL abstrakte Interaktionsobjekte zur modalitätenunabhängigen Definition. Mittels geeigneter Transformationen können aus diesen abstrakten Interaktionsobjekten konkrete Interaktionsobjekte des konkreten Präsentationsmodells generiert werden. Jedes der Interaktionsobjekte lässt sich einer der drei in Tab. 5.5 vorgestellten Klassen zuordnen.

Diese Klassen spezifizieren, welche Funktion das Interaktionsobjekt innerhalb der Benutzungsschnittstelle übernimmt. In DISL gibt es zwei Elemente, die für die Modellierung einer reinen Informationsausgabe eingesetzt werden können. Interaktionselemente dienen der Eingabe und repräsentieren Elemente, mit denen der Benutzer aktiv mit der Benutzungsschnittstelle in Interaktion treten kann. Kollektionsobjekte werden zur Modellierung einer Auswahl benutzt, d. h. mit ihnen ist es dem Benutzer möglich aus einer Menge von Ausgabeelementen auszuwählen. Die Auswahl wird dabei über die Struktur der Kollektionsobjekte beschrieben. Elemente, die Teil der Auswahl sein sollen, müssen in der Hierarchie der Kollektionsobjekte als Kinder untergeordnet sein. Dabei kann zwischen einer inklusiven und einer exklusiven Auswahl unterschieden werden. Bei einer inklusiven Auswahl können mehrere Elemente gleichzeitig ausgewählt werden. Bei der exklusiven Auswahl hingegen kann immer nur eines der Elemente exklusiv ausgewählt werden.

Die konkrete Benutzungsschnittstelle wird durch das **konkrete Präsentations- und Dialogmodell** gebildet. Sie kann bspw. grafisch oder sprachbasiert modelliert werden. Beispielsweise wird im weiteren Verlauf dieses Kapitels auf grafische Benutzungsschnittstellen, als eine mögliche Art von konkreten Benutzungsschnittstellen, fokussiert. Dabei wird die in Forschung und Industrie etablierte **User Interface Markup Language (UIML)** verwendet (Meixner 2010).

Mit der User Interface Markup Language (UIML) wurde eine XML-basierte Metasprache entwickelt, die der universellen und geräteunabhängigen Beschreibung von Benutzungsschnittstellen dient. UIML basiert auf dem Meta-Interface Modell (MIM). Die grundlegende Idee dieser Modelle bzw. Paradigmen liegt in der Separierung der Benutzungsschnittstelle von der Applikationslogik. Das MIM, welches in Abb. 5.20 visualisiert wird, separiert Präsentation, Schnittstelle und Logik von einander. Dies gewährleistet insbesondere Aspekte der Plattformunabhängigkeit, Wiederverwendbarkeit und Wartbarkeit einzelner Komponenten.

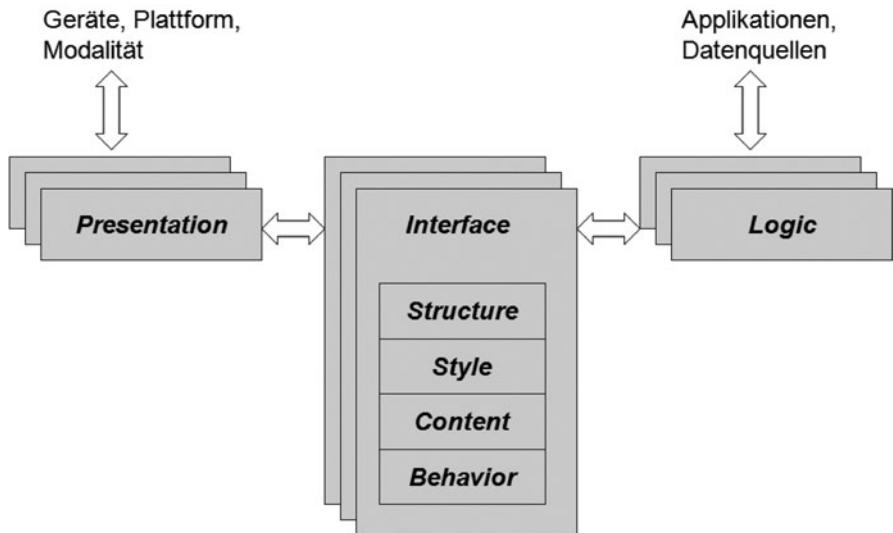


Abb. 5.20 Das Meta-Interface Model von UIML

Eine in UIML spezifizierte Benutzungsschnittstelle wird durch das **Interface**-Element (<interface>) beschrieben. Dabei setzt sich dieses Element aus den vier untergeordneten Elementen Structure, Style, Content und Behavior zusammen. Das **Structure**-Element (<structure>) gibt eine hierarchische Gliederung der Benutzungsschnittstelle wieder. Dabei lässt sich die Benutzungsschnittstelle in Teilelemente (<part>) weiter untergliedern. Ein Teilelement ist ein einzelner Baustein, der unabhängig von jeglichem Typ und konkreter Realisierung existiert. Ein Teilelement kann wiederum beliebig viele weitere Teilelemente enthalten (Containment-Prinzip), wodurch sich strukturierte und komplexe verschachtelte Benutzungsschnittstellen definieren lassen. Ebenso ist es möglich, während der Ausführung der Benutzungsschnittstelle die Struktur anzupassen. Dazu können Teilbäume der Struktur entfernt, verschoben, oder neu eingegliedert werden.

Die Darstellungsbeschreibung der Benutzungsschnittstellen-Elemente geschieht mit Hilfe von **Styles** (<style>). Diese definieren, wie ein Interaktionsobjekt in der Zielsprache dargestellt wird, d. h. das <style>-Element enthält eine Liste aller Präsentationseigenschaften und Werte (z. B. Schriftart, Farbe), die für die jeweilige Modalität relevant sind.

In UIML wird das Verhalten der Benutzungsschnittstelle mit dem **Behavior**-Element <behavior> spezifiziert. Das Behavior-Element beschreibt, wie die Benutzungsschnittstelle reagieren soll, wenn ein Benutzer mit ihr interagiert. Hierfür enthält jedes Behavior-Element eine Reihe von Regeln (Rule-Elemente (<rule>)), die wiederum jeweils eine Bedingung, das Condition-Element (<condition>) und mehrere Action-Elemente (<action>) enthalten. Das Rule-Element verbindet somit eine Bedingung mit Aktionen. Ist die Bedingung erfüllt, dann wird die entsprechende Aktion ausgeführt. Häufig hat ein Ereignis der Benutzungsschnittstelle Auswirkungen sowohl auf die Anzeige als auch auf die dahinter stehende Applikationslogik.

(MVC-Paradigma). Diese Verhaltensbeschreibung entspricht im Wesentlichen der Modellierung des (konkreten) Dialogmodells.

Das **Peers**-Element (<peers>) erlaubt UIML die Übersetzung der Part-Elemente und deren Stilbeschreibungen in Interaktionsobjekte einer Zielplattform (Finale Benutzungsschnittstelle). Neben der Anbindung von Interaktionsobjekten kann mit dem Peers-Element auch die Anbindung der in UIML spezifizierten Applikationslogik (<logic>) an die Zielsprache beschrieben werden. Bisher musste für jede Transformation in eine finale Benutzungsschnittstelle jeweils ein eigenes Vokabular entwickelt werden (Souchon, Vanderdonckt 2003). Dieses Defizit wurde jedoch durch aktuelle Entwicklungen behoben.

UIML unterstützt ebenfalls die Wiederverwendung von häufig benötigten Benutzungsschnittstellenkomponenten. Solch eine Komponente kann z. B. ein „Datei öffnen“-Dialog sein. Dieser lässt innerhalb eines **Templates** in UIML definieren und kann von beliebigen Dokumenten referenziert, und in die aktuelle Benutzungsschnittstelle nachgeladen werden.

Als eine Sprache zur Beschreibung von grafischen Benutzungsschnittstellen hat sich die User Interface Markup Language (UIML) als geeignet erwiesen. Mittels UIML ist es möglich, grafische Benutzungsschnittstellen in generischer Weise, durch Verwendung von generischen Vokabularen, zu erstellen und auf eine Vielzahl von Zielplattformen (Finale Benutzungsschnittstelle) zu transformieren (Meixner und Schäfer 2009). Ein generisches Vokabular für grafische Benutzungsschnittstellen definiert eine Menge an Interaktionsobjekten, die in eine Vielzahl von Grafikbibliotheken abgebildet werden können. Bspw. können Textfelder, Comboboxen oder Radiobuttons in eine Reihe verschiedener Grafikbibliotheken wie bspw. Swing, GTK# abgebildet werden.

Ansätze zur Integration von UIML in durchgängige modellbasierte Entwicklungsprozesse werden bspw. in (Meixner, Schäfer 2009) beschrieben. Ein weiterer Vorteil bei der Verwendung von UIML als XML-basierte Sprache zur Beschreibung der konkreten Benutzungsschnittstelle ist die Verfügbarkeit von ausgereiften Entwicklungswerkzeugen wie bspw. LiquidUI (als Bestandteil der LiquidApps-Umgebung²) zur Bearbeitung von UIML-Dokumenten. Abbildung 5.21 zeigt die Entwicklungsumgebung von LiquidUI zum intuitiven Erstellen und Bearbeiten von UIML-Dokumenten. Des Weiteren kann UIML, neben der Transformation in Programmiersprachen, für die direkte Interpretation zur Visualisierung und Simulation genutzt werden (Meixner, Schäfer 2009).

5.4.6 Zusammenfassung und Ausblick

Zusammenfassend lässt sich feststellen, dass das Gebiet der modellbasierten Entwicklung von Benutzungsschnittstellen die Entwicklung von (multimodalen) Benutzungsschnittstellen auf verschiedenen Plattformen deutlich effizienter gestalten

² <http://www.liquidappsworld.com/>

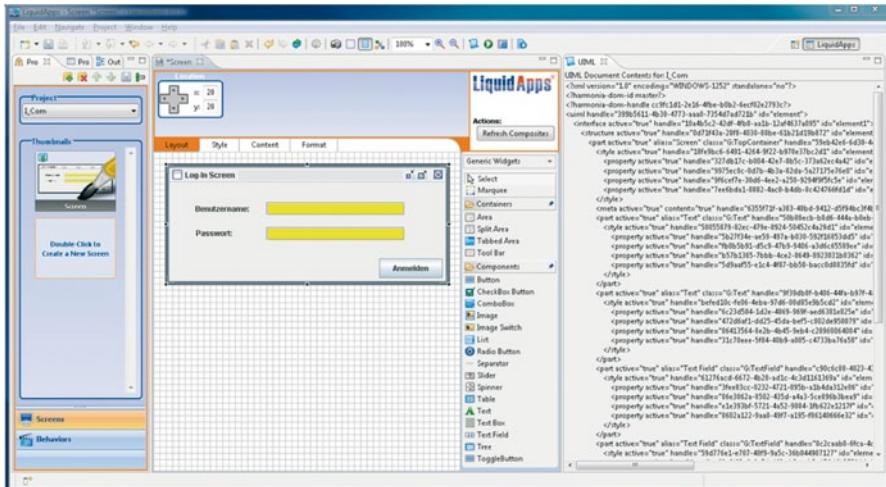


Abb. 5.21 Screenshot eines Beispielprojektes in LiquidUI 5.0

kann. Dabei ist u. a. auch zu bemerken, dass sich durch eine inhaltliche und strukturelle Trennung von Modellen, Designer und Programmierer auf ihre jeweiligen Kernkompetenzen konzentrieren können. Erweiterungen und Änderungen an Benutzungsschnittstellen können somit einfach und ohne großen Mehrfachaufwand auf einer Vielzahl von Plattformen konsistent durchgeführt werden.

Um das Ziel einer durchgängigen modellbasierten Entwicklungsumgebung zu erreichen ist es insbesondere notwendig, weitere Transformationsprozesse zu spezifizieren und in entsprechende Transformatoren zu integrieren. Hierbei ist es zuerst notwendig, mögliche Zielmodalitäten wie bspw. sprachbasierte oder haptische Benutzungsschnittstellen zu definieren und geeignete Modelle sowie (XML-basierte) Sprachen auszuwählen oder zu erarbeiten. Dabei müssen Standardsprachen Verwendung finden, die in Forschung und Industrie genutzt werden. Des Weiteren müssen auch Rückwärtstransformationen (Reverse Engineering) entwickelt und integriert werden. Dies gewährleistet die Erhaltung der Konsistenz der Modelle, wenn Optimierungen auf einer konkreteren Ebene stattfinden, da entsprechende Änderungen automatisch in abstraktere Modelle integriert werden. Diese Punkte führen insgesamt dazu, dass die modellbasierte Architektur und die dazugehörige Werkzeugkette in bestehende industrielle Entwicklungsprozesse integriert werden können.

Des Weiteren kann durch Integration von weitergehenden Mechanismen zur automatisierten Modellevaluation eine Steigerung der Effizienz ermöglicht werden. Bspw. können Model-Checking Algorithmen innerhalb des Dialogmodells integriert werden, die automatisiert prüfen, ob im Dialogmodell Verklemmungen (deadlocks) existieren oder ob alle spezifizierten Zustände der Benutzungsschnittstelle tatsächlich erreichbar sind.

Kapitel 6

Grundlegende Prinzipien und Interaktionstechniken

Nachdem in Kap. 3 die Vorgehensweise bei der ergonomischen Bediensystementwicklung vorgestellt wurde, soll in diesem Kapitel näher auf Gestaltungsdetails und -anforderungen eingegangen werden. Hierfür wird zunächst ein Überblick über relevante Normen und Richtlinien gegeben. In Kap. 6.2 werden die wichtigsten Prinzipien zur Gestaltung von Bediensystemen erläutert. Kapitel 6.3 befasst sich mit der Gestaltung der Interaktion zwischen Mensch und Maschine. Dies umfasst eine Vorstellung möglicher Interaktionsformen (z. B. Kommandosprachen, Dialoge, Menüs), gängiger Interaktionselemente und diverser Interaktionstechnologien (Ein-Ausgabegeräte).

6.1 Normen und Richtlinien

Normen sind allgemein anerkannte bzw. für die Allgemeinheit festgelegte Definitionen, Durchführungsbestimmungen oder Regeln. Die englische Übersetzung „standards“ bezeichnet den Zweck einer Norm vielleicht am Besten. Es sollen hier lediglich Festlegungen getroffen werden, die den Anwendern der Normen als Standard, nicht jedoch als Pflicht dienen. National und international haben sich verschiedene Normungsgremien entwickelt. Diese werden in Tab. 6.1 dargestellt.

Neben den nationalen und internationalen Normungsinstituten existieren weitere Verbände, die eigene Richtlinien herausgeben, welche zumeist den Charakter von Empfehlungen haben und eine Dokumentation von anerkannten Regeln und dem Stand des Wissens aus der jeweiligen Domäne darstellen. Der Verein Deutscher Ingenieure (VDI) ist – zusammen mit dem Verband der Elektrotechnik Elektronik Informationstechnik (VDE) – der größte technische Verein in Deutschland. Aus diesem großen technischen Wissen über alle technischen Bereiche und über die Zusammenarbeit mit Forschung und Industrie werden seit Jahrzehnten Richtlinien generiert, die anderen Ingenieuren dazu dienen sollen, Entwicklungen zu vereinfachen oder zu verbessern.

In Tab. 6.2 werden einzelne Normen und Richtlinien aufgeführt, die bei der Gestaltung von Mensch-Maschine-Systemen Hilfestellung leisten können. Im An-

Tab. 6.1 Nationale und internationale Normungsgremien

Geltungsbereich	Kürzel	Name
National	DIN	Deutsches Institut für Normung
International	CEN	Commission Européen de Normalisation
	IEC	International Electrotechnical Committee
	ISO	International Standards Organization
	ECMA	European Computer Manufacturers Association
	HFS-HCI	Human Factors Society-Human-Computer Interaction Standards Committee
	IEEE	Institute of Electrical and Electronic Engineers
	OSF	Open Systems Foundation
	UI	UNIX International

schluss an die Tabelle werden einige wichtige Normen und Richtlinien detaillierter aufgeführt und kurz erläutert.

6.1.1 DIN EN ISO 9241

Die wichtigste Norm für die Entwicklung von Bediensystemen ist die DIN EN ISO 9241. In dieser Norm wird in derzeit insgesamt 29 Teilen das weltweit verfügbare Wissen zur ergonomischen Gestaltung von Software, Hardware und Arbeitsumgebung gesammelt, bewertet und in einer Norm integriert. Die Norm bietet Hilfestellungen bei der Konzeption, Gestaltung und Bewertung von Bildschirmarbeitsplätzen und definiert Mindestforderungen für die ergonomische Gestaltung von Software. Es muss festgehalten werden, dass die Gestaltungsprinzipien in erster Linie für Anwendungen im Bürobereich entwickelt wurden und nicht vorbehaltlos für Anwendungen in Fertigungsbereichen übernommen werden können; dies gilt vor allem für die Steuerbarkeit und die Individualisierbarkeit. Dennoch sind die zu erfüllenden Eigenschaften so allgemein, dass sie auch auf Bediensysteme übertragen werden können, die neben Bildschirmen weitere Ein- und Ausgabemöglichkeiten besitzen.

Die derzeit veröffentlichten 26 Teile der Norm sind im Folgenden aufgelistet:

- Teil 1: Allgemeine Einführung
- Teil 4: Anforderungen an die Tastatur
- Teil 5: Anforderungen an Arbeitsplatzgestaltung und Körperhaltung
- Teil 6: Leitsätze für die Arbeitsumgebung
- Teil 9: Anforderungen an Eingabemittel – ausgenommen Tastaturen
- Teil 11: Anforderungen an die Gebrauchstauglichkeit – Leitsätze
- Teil 12: Informationsdarstellung
- Teil 13: Benutzerführung
- Teil 14: Dialogführung mittels Menüs
- Teil 15: Dialogführung mittels Kommandosprachen

Tab. 6.2 Normen für die Gestaltung von Mensch-Maschine-Schnittstellen

Art	Nr.	Titel, Stichworte	Inhalt, Bemerkungen
DIN	1450	Leserlichkeit von Schriften	Definitionen, Auflistung der Einflussgrößen auf die Leserlichkeit, Hauptkriterien für die Leserlichkeit, räumliche Leserlichkeitsbereiche, Schriftgrößen
DIN	2137	Alphanumerische Tastaturen: Deutsche Tastatur für Text- und Datenverarbeitung	Belegung mit Schriftzeichen; Tastenanordnung und Maße; Belegung von Zeichtasten; Tastenanordnung und -belegung für tragbare Rechner, geteilte und abgewinkelte Tastaturen
DIN	2340	Kurzformen für Benennungen und Namen	Begriffsdefinitionen, Bildungsregeln, Spezialfälle, alphabetisches Register
DIN	4844	Sicherheitskennzeichnung	Begriffe, Grundsätze und Sicherheitszeichen; Sicherheitsfarben
DIN	5034	Tageslicht in Innenräumen	Allgemeine Anforderungen; Grundlagen; Berechnung; Messung
DIN	5035	Beleuchtung mit künstlichem Licht	Begriffe und allgemeine Anforderungen; Richtwerte für Arbeitsstätten in Innenräumen und im Freien; Notbeleuchtung; Messung und Bewertung; Beleuchtung von Räumen mit Bildschirmarbeitsplätzen und mit Arbeitsplätzen mit Bildschirmunterstützung
DIN	6164	DIN-Farbenkarte	System der DIN-Farbenkarten
DIN	19227	Grafische Symbole und Kennbuchstaben für die Prozessleitechnik: Darstellung von Aufgaben	Kennbuchstaben zur Darstellung der funktionalen Arbeitsweise, grafische Symbole mit Bedeutungen, Gerätesymbole
DIN	19235	Meldung von Betriebszuständen	Begriffsdefinition, Meldungscodierung durch Blinken, Töne, Farben
DIN	33402	Körpermaße des Menschen	Begriffe, Messverfahren; Werte; Bewegungsraum; Grundlagen für die Bemessung von Durchgängen, Durchlässen und Zugängen
DIN	33403	Klima am Arbeitsplatz und in der Arbeitsumgebung	Grundlagen; Einfluss auf den Wärmehaushalt des Menschen; Beurteilungen
DIN	33404	Akustische Gefahrensignale	Begriffe, Anforderungen, Prüfung, Gestaltungshinweise
DIN	33414	Ergonomische Gestaltung von Warten	Sitzarbeitsplätze: Begriffe, Grundlagen, Maße; kognitive Faktoren; Gestaltungskonzept; Gliederungsschema, Anordnungsprinzipien
DIN EN	66001 894	Informationsverarbeitung Nutzerinteraktion m. Anzeigen & Stellteilen	Sinnbilder und ihre Anwendung Grundlagen; Anzeigen; Stellteile

Tab. 6.2 (Fortsetzung)

Art	Nr.	Titel, Stichworte	Inhalt, Bemerkungen
EN	29241	Ergonomische Anforderungen für Büro-tätigkeiten mit Bildschirmgeräten	Anforderungen an die Arbeitsaufgaben; Anforderungen an visuelle Anzeigen; Farbdarstellungen; Grundsätze der Dialoggestaltung; Informationsdarstellung; Nutzerführung; Dialog m. Menü; Kommandosprache; Bildschirmformulare (identisch zur ISO 9241)
EN	60073	Kodierung von Anzeigegeräten und Bedienelementen durch Farben und ergänzende Mittel	Deutsche Übersetzung der Internationalen Norm IEC 73
EN	60447	Grund und Sicherheitsregeln für die Mensch-Maschine-Schnittstelle	Kennzeichnung, Bedienungsgrundsätze, Betätigung und Wirkung
IEC	73	Bedeutung von Farben	Kennfarben für Leuchtmelder und Druckknöpfe
IEC	16336	Mensch-Maschine-Schnittstelle	Kodierungsgrundsätze für Anzeigen
ISO	9241	Ergonomische Anforderungen für Büro-tätigkeiten mit Bildschirmgeräten	Allgemeine Einführung; Anforderungen an die Arbeitsaufgaben; Anforderungen an visuelle Anzeigen; Anforderungen an Arbeitsplatzgestaltung und Körperhaltung; Anforderungen an die Arbeitsumgebung; Anforderungen an visuelle Anzeigen bezüglich Reflexionen; Anforderungen an Farbdarstellungen; Anforderungen an Eingabegeräte; Grundsätze der Dialoggestaltung; Angaben zur Benutzbarkeit; Informationsdarstellung; Nutzerführung; Dialogführung mittels Menüs, Kommandosprache, indirekter Manipulation und Bildschirmformularen
ISO	14915	Software-Ergonomie für Multimedia-Benutzungsschnittstellen	Gestaltungsgrundsätze und Rahmenbedingungen; Multimedia-Navigation und Steuerung; Auswahl und Kombination von Medien
ISO	20282	Einfachheit der Handhabung von Produkten des täglichen Gebrauchs	Gestaltungsanforderungen im Kontext von Anwendungs- und Benutzermerkmalen; Prüfverfahren für öffentlich zugängliche Produkte
VDE	199	Bedeutung von Farben	Identisch zur IEC 73
VDI/ VDE	3699	Prozessführung mit Bildschirmen	Grundlagen und Begriffe; vorgestaltete Darstellungen; Fließbilder; Kurven; Meldungen;
VDI/ VDE	3850	Nutzergerechte Gestaltung von Bediensystemen für Maschinen	Grundlagen; Interaktionsgeräte für Bildschirme; Dialoggestaltung für Touchscreens

- Teil 16: Dialogführung mittels direkter Manipulation
- Teil 17: Dialogführung mittels Bildschirmformularen
- Teil 20: Leitlinien für die Zugänglichkeit der Geräte und Dienste in der Informations- und Kommunikationstechnologie

- Teil 110: Grundsätze der Dialoggestaltung
- Teil 151: Leitlinien zur Gestaltung von Benutzungsschnittstellen für das World Wide Web
- Teil 171: Leitlinien für die Zugänglichkeit von Software
- Teil 300: Einführung in die Anforderungen an elektronische optische Anzeigen
- Teil 302: Terminologie für elektronische optische Anzeigen
- Teil 303: Anforderungen an elektronische optische Anzeigen
- Teil 304: Prüfverfahren zur Benutzerleistung für elektronische optische Anzeigen
- Teil 305: Optische Laborprüfverfahren für elektronische optische Anzeigen
- Teil 306: Vor-Ort-Bewertungsverfahren für elektronische optische Anzeigen
- Teil 307: Analyse- und Konformitätsverfahren für elektronische optische Anzeigen
- Teil 400: Grundsätze und Anforderungen für physikalische Eingabegeräte
- Teil 410: Gestaltungskriterien für physikalische Eingabegeräte

6.1.2 VDI 3850

Mit der Richtlinie 3850 „Nutzergerechte Gestaltung von Bediensystemen für Maschinen“ soll Entwicklern von Bediensystemen ein Gestaltungsrahmen an die Hand gegeben werden, mit dessen Hilfe sie ein Bediensystem schnell und nutzerfreundlich erstellen können. Damit soll dem Entwickler jedoch kein Standardbediensystem aufgezwungen werden, vielmehr werden ihm genügend gestalterische Freiräume zur Erarbeitung eigenständiger Bediensysteme gelassen.

Die Regeln und Empfehlungen dieser Richtlinie beziehen sich vor allem auf informationsergonomische und weniger auf hardwareergonomische Gesichtspunkte. Die Richtlinie besteht aus drei Teilen, die die folgenden Inhalte haben:

- **Blatt 1: Grundlagen**

Das Blatt 1 behandelt die Grundlagen der Maschinenbediensystemgestaltung und bildet die Basis für die weiteren Blätter. Die Inhalte sind im Einzelnen: Vorgehen bei der Gestaltung von Bediensystemen, Grundlagen zur Gestaltung von Bediensystemen, Informationscodierung und Aufmerksamkeitssteuerung, Bildzeichengestaltung für Bildschirme und Bedienfelder, Abkürzungen, Informations- und Meldeklassen, Grundlagen der Dialoggestaltung (s. Abb. 6.1).

- **Blatt 2: Interaktionsgeräte für Bildschirme**

Das Blatt 2 wendet sich speziell an Entwickler und Anwender, die ein geeignetes Interaktionsgerät für die Maschinenbedienung suchen. Die Richtlinie bietet Auswahlkriterien und Gestaltungsempfehlungen für industrietaugliche und dem Arbeitsumfeld angepasste Interaktionsgeräte. Die Auswahlkriterien beziehen sich auf das spezielle Interaktionsgerät. Sie sind nach den beeinflussenden Faktoren Aufgabentypen, ergonomischen Merkmalen und Umweltbedingungen

Informationsklassen und ihre Merkmale		Darstellungsattribute		Visualisierungs-ort	Zusätzliches Audiosignal	Zeitpunkt	Benutzerreaktion	Initiative
Prio.	Bezeichnung	Hintergrundfarbe	Vordergrundfarbe	Umrahmung	Schattierung	Blinken	Beginn	Ende
hoch	Störungsinformationen	rot schwarz	schwarz rot	sollte, sollte, bis das Lesen quitt. wurde		fest, nicht	mit Alarmereignis belebung und Quittierung	quitieren, beheben
	Alarne							
	Warnungen	gelb schwarz gelb		fest sollte kann ergänzt werden	fest, nicht	kurzzzeitig	mit Warnereignis Wegfall oder Quittierung	System
	Statusinformationen normal	weiß schwarz weiß		kann nein	fest, nicht	überdeckbar	dauern	System
	gestört	rot schwarz gelb						
	Hinweisinformationen	blau weiß blau		fest sollte kann nein				divers: auswählen bestätigten
	Bedienfehlerinformationen						kann, kurzzeitig	System
niedrig	Allgemeine Informationen						mit Bedienfehler	System
	Hilfeinformationen							Benutzer

Abb. 6.1 Informationsklassen nach VDI/VDE 3850

unterteilt. Die Gestaltungsempfehlungen beziehen sich auf die Verbindung des Interaktionsgerätes mit dem Gesamtsystem.

- **Blatt 3: Dialoggestaltung für Touchscreens**

Das Blatt 3 trägt dem Punkt Rechnung, dass in industriellen Anwendungen ein deutlicher Trend zur Touchscreentechnologie zu erkennen ist. Deswegen soll dem Entwickler von Bediensystemen eine Richtlinie zur Verfügung gestellt werden, die ihn bei der Gestaltung eines touchscreenspezifischen Dialoges unterstützt, aber weiterhin genügend gestalterische Freiräume zur Erarbeitung eigenständiger Lösungen belässt.

6.1.3 VDI 3699

Zu der VDI/VDE-Richtlinie 3850 stellt die Richtlinie 3699 „Prozessführung mit Bildschirmen“ das Pendant für die Verfahrenstechnik dar. Diese Richtlinie soll Basis für die sachgerechte Umsetzung der Bildschirmarbeitsplatzverordnung in Warten der Verfahrenstechnik sein. Nutzen soll diese Richtlinie sowohl Betreibern und End-Nutzern als auch Herstellern von Bedien- und Beobachtungssystemen der Prozessleittechnik.

Die Richtlinie besteht im Einzelnen aus den folgenden Blättern:

- **Blatt 1: Begriffe**

Blatt 1 der Richtlinie stellt die in den folgenden Blättern verwendeten Begriffe vor und schafft ein grundlegendes Verständnis für die Begrifflichkeiten.

- **Blatt 2: Grundlagen**

Blatt 2 der Richtlinie erläutert Ziele der Prozessführung, Grundlagen der Gestaltung von Mensch-Maschine-Schnittstellen und die Organisation von Leitstationen.

- **Blatt 3: Fließbilder**

Blatt 3 der Richtlinie nennt Regeln und gibt Empfehlungen zur Gestaltung von Fließbildern auf Bildschirmen sowie zur Prozessführung mit Hilfe von Fließbildern.

- **Blatt 4: Kurven**

Blatt 4 der Richtlinie nennt Regeln und gibt Empfehlungen zur Darstellung von Kurven auf Bildschirmen und zur Bedienung von Kurvenbildern zur Prozessführung. Beispiele für unterschiedliche Varianten von Kurvenbildern zeigen, wie sich die genannten Regeln und Empfehlungen praxisgerecht umsetzen lassen.

- **Blatt 5: Meldungen**

Blatt 5 der Richtlinie gibt Empfehlungen und Regeln zur Darstellung von Meldungen auf Bildschirmen und zur Bedienung von Meldesystemen. Weitere Empfehlungen gelten der Darstellung einzelner Meldungen sowie dem Aufbau von Meldeseiten und Meldeprotokollen.

- **Blatt 6: Bedienverfahren und Bediengeräte**

Blatt 6 der Richtlinie beschreibt unterschiedliche Bedienverfahren und Bedienarten für verschiedene Anwendungszwecke und Bediengeräte. Ergonomisch günstige Bedienverfahren unter Beachtung geltender Normen werden genannt.

6.2 Gestaltungsprinzipien für Bediensysteme

Im folgenden Abschnitt dieses Kapitels werden die grundlegenden Prinzipien für die Bediensystemgestaltung vorgestellt. Wichtigstes Kriterium für die Anordnung der Komponenten ist die richtige und eindeutige Übermittlung von Informationen, so dass das Übersehen und Verwechseln von Informationen sowie die daraus resultierenden Betätigungsfehler mit hoher Wahrscheinlichkeit ausgeschlossen werden. Des Weiteren soll durch die Gestaltung des gesamten Bediensystems die Informationsverarbeitung unterstützt werden.

6.2.1 Grundsätze der Dialoggestaltung

Teil 110 der DIN EN ISO 9241 enthält sieben Grundsätze, die unabhängig von einer bestimmten Software als allgemeine Leitlinien bei der Leistungsbeschreibung, Gestaltung und Bewertung von Dialogsystemen angewendet werden sollten. Die Einhaltung dieser Prinzipien in einer konkreten Anwendung kann nicht mit einer Ja/Nein-Entscheidung beurteilt werden. Wie gut sie umgesetzt sind, muss systemspezifisch beurteilt werden.

6.2.1.1 Aufgabenangemessenheit

Anwendungen sind aufgabenangemessen, wenn sie den Nutzer bei der Erledigung seiner Arbeitsaufgabe unterstützen, ohne ihn mental oder körperlich jedoch unnötig zu belasten. Dies umfasst eine Anpassung der Dialogreihenfolge, Ein- und Ausgabeform und Informationsbereitstellung an die Eigenschaften der zu erfüllenden Arbeitsaufgabe.

Die Reihenfolge der in einem Dialog dargestellten Informationen und zu leistenden Eingaben sollte sich an der Relevanz für die Erfüllung der damit verknüpften Arbeitsaufgabe orientieren. Wichtige Informationen/Eingabeelemente sollten gut sichtbar an oberster Stelle platziert werden, während weniger häufig benötigte eher am Ende des Dialoges stehen sollten.

6.2.1.2 Erwartungskonformität

Eine Anwendung verhält sich erwartungskonform, wenn der Dialogablauf den Erwartungen der Nutzer entspricht, die sich aus Erfahrungen mit bisherigen Arbeitsabläufen, der bisherigen Benutzung des Systems sowie anderer Systeme ergeben. Als zentraler Punkt ist hier die Konsistenz der Benutzungsoberfläche hinsichtlich Informationsdarstellung und Dialogverhalten (anwendungsintern und anwendungsübergreifend) zu nennen. Abweichungen von für den Benutzer gewohnten kulturellen und sprachlichen Konventionen sowie Vokabular sind zu vermeiden. Weiterhin sollten unmittelbare und passende Rückmeldungen auf Benutzerhandlungen erfolgen.

Eine konsistente Präsentation (z. B. Platzierung gleicher Informationen an den gleichen Stellen) und konsistente Interaktion (z. B. wiederkehrende Ablaufschemata, einheitliche Funktionstastenbelegung) machen ein System für den Nutzer erwartungskonform.

6.2.1.3 Selbstbeschreibungsfähigkeit

Ein Dialog ist selbstbeschreibungsfähig, wenn für den Nutzer zu jeden Zeitpunkt offensichtlich ist, in welchem Dialog und an welcher Stelle im Dialog er sich befindet. Weiterhin müssen an dieser Stelle alle Handlungsoptionen und Interaktionsmöglichkeiten für den Nutzer transparent sein, ggf. durch zusätzliche Informationen.

Der Nutzer muss jederzeit wissen, in welchem Zustand sich das System befindet und erkennen können, wie weit er mit seiner Aufgabe fortgeschritten ist. Deshalb müssen bei längeren Vorgängen Zwischenausgaben bzw. Vollzugsmeldungen den Stand der Bearbeitung anzeigen. Auch muss der Nutzer wissen, dass das System seinen Auftrag bearbeitet und nicht etwa auf weitere Eingaben wartet (Feedback).

6.2.1.4 Fehlertoleranz

Ein Dialog sollte sich dadurch auszeichnen, dass Fehler weitgehend verhindert und fehlerhafte Eingaben bzw. zu einem falschen Zeitpunkt ausgeführte Funktionen leicht zu korrigieren sind. Der Benutzer soll durch geeignete Erläuterungen bei der

Fehlerbeseitigung unterstützt werden und vor Handlungen mit schwerwiegenden Konsequenzen gewarnt und zur Bestätigung aufgefordert werden.

Folgenschwere und irreversible Anweisungen wie z. B. das Löschen von Daten in der Datenbank müssen vor der Ausführung noch einmal bestätigt werden, um deren versehentliche Ausführung zu vermeiden.

6.2.1.5 Steuerbarkeit

Ein Dialog wird als steuerbar bezeichnetet, wenn der Benutzer in der Lage ist, den Dialogablauf zu starten sowie seine Richtung und Geschwindigkeit zu beeinflussen. Unterbrechungen sollten möglich sein und die Wiederaufnahme des Dialogs sollten wenn möglich nach einer Unterbrechung an einem vom Benutzer zu wählenden Punkt stattfinden können. Wenigstens der letzte Dialogschritt sollte rückgängig gemacht werden können, soweit die ausgeführten Handlungsschritte dies zulassen. Falls Datenänderungen vorgenommen werden, sollten die Originaldaten ggf. weiterhin einsehbar bleiben, falls die Arbeitsaufgabe dies erfordert. Ein- und Ausgaben innerhalb eines Dialoges sollten vom Benutzer über jedes verfügbare Ein- und Ausgabemittels durchgeführt werden können.

Der Nutzer muss die Möglichkeit haben, einen Prozess zu unterbrechen, um in einem anderen zu arbeiten. Er darf nur in Ausnahmefällen in einem Bearbeitungszustand festgehalten werden, zum Beispiel, wenn eine bestimmte Eingabe von ihm verlangt wird, damit seine gewünschte Aktion überhaupt ausgeführt werden kann. Er muss also jederzeit wissen, in welchem Prozess er sich gerade befindet (beispielsweise durch veränderten Cursor) und wie er diesen beenden kann.

6.2.1.6 Individualisierbarkeit

Hinter dem Kriterium der Individualisierbarkeit versteckt sich die Flexibilität, dass das System den Benutzern erlaubt, die Mensch-System-Interaktion und die Darstellung von Informationen an ihre eigenen Bedürfnisse und Fähigkeiten anpassen zu können. Dies umfasst die Bereitstellung von Techniken zur Anpassung an charakteristische Eigenschaften von Benutzern, Wählbarkeit zwischen verschiedenen Darstellungsformen, Erläuterungsumfängen, Ein- und Ausgabegeschwindigkeiten und Dialogtechniken. Dialogelemente oder Funktionen sollten hinzugefügt oder neu ge-

ordnet werden können, soweit dies zweckmäßig ist. Jede individuelle Einstellungen muss aber auch wieder rückgängig gemacht werden können und zu Standardeinstellungen zurückgekehrt werden können.

Variabilität: Neulinge, Gelegenheitsnutzer und erfahrene Nutzer sind gleichermaßen zu unterstützen. Hierfür werden parallel alternative Interaktionstechniken für dieselbe Funktion angeboten (z. B. Klappmenü/Kontextmenü/Doppelklick; Klappmenü/Tastenkombination).

Dynamik: Die Anzahl und Art der Objekte in einer Objektmenge ist dynamisch veränderlich. Der Nutzer kann sich die Menge wahlweise als Menge von Ikonen, als Liste oder als Strukturdarstellung (Baum) anzeigen lassen.

6.2.1.7 Lernförderlichkeit

Ein Dialog ist lernförderlich, wenn er den Benutzer beim Erlernen der Nutzung des interaktiven Systems unterstützt und anleitet. Hierzu ist es hilfreich, wenn Regeln und zugrundeliegende Konzept dem Benutzer zugänglich gemacht werden. Geeignete Unterstützung (z. B. durch ein Hilfesystem) sollte dem Benutzer bei seltenen Dialogen und bei der erstmaligen Benutzung verfügbar gemacht werden. Rückmeldungen und Erläuterungen können dabei dazu dienen, dem Benutzer ein konzeptuelles Verständnis vom System zu geben und ihn anhand von erfolgreichen Handlungen das System erlernen lassen. Insgesamt sollte ein minimaler Lernaufwand zur Ausführung der Arbeitsaufgabe angestrebt werden. Dies kann durch Dialog mit minimaler Informationseingabe ermöglicht werden, die zusätzlichen Informationen auf Anforderung bereitstellen.

Die Struktur der Anwendung sollte die reale Welt widerspiegeln. Hierzu ist insbesondere die objektorientierte Dialogstrukturierung geeignet. Abkürzungen und Kurzbefehle sind sinnfällig und griffig zu wählen.

6.2.2 Weitere Prinzipien

Die oben genannten Eigenschaften eines Bediensystems sind nicht unabhängig voneinander zu sehen, vielmehr ist es notwendig, die Vorteile eines Grundsatzes gegenüber denen eines anderen abzuwagen.

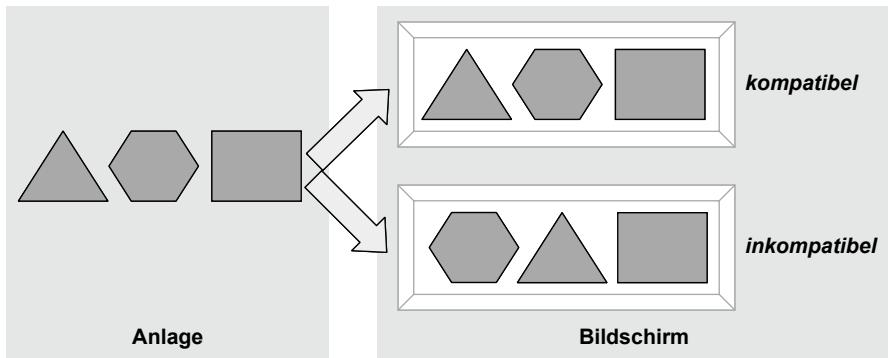


Abb. 6.2 Kompatibilität des Ortes

Um die übergeordneten Eigenschaften in einem Bediensystem zu realisieren, sollten bei der Gestaltung folgende Prinzipien angewendet werden.

6.2.2.1 Kompatibilität

Eine grundlegende Anforderung an ein nutzergerechtes Bediensystem ist seine Kompatibilität, die sich aus der Übereinstimmung der Anordnung in der Realität und in der Darstellung auf dem Bediensystem bzw. der Anordnung in der Realität und in der Vorstellung des Nutzers ergibt.

Bezüglich der Kompatibilität werden drei verschiedene Ausprägungen unterschieden:

- Kompatibilität des Ortes:
Die Anordnung der Bedien- und Beobachtungseinheiten entspricht der Anordnung in der realen Anlage (vgl. Abb. 6.2).
- Kompatibilität der Richtung:
Die Anordnung und die Funktionalität der Anzeige- und Bedienelemente entspricht dem Material-, Energie- und Informationsfluss (vgl. Abb. 6.3).
- Kompatibilität der Folge:
Die Anordnung der Bedienelemente entspricht der zeitlichen Reihenfolge ihrer Bedienung (vgl. Abb. 6.4).

6.2.2.2 Konsistenz

Konsistenz ist gegeben, wenn die Gestaltung von Darstellungen, Strukturen, Begriffen, (Dialog-)Abläufen einheitlich ist und dadurch widerspruchsfreie und zusammenhängende Gedankengänge gefördert werden. Ein konsistentes Bediensys-

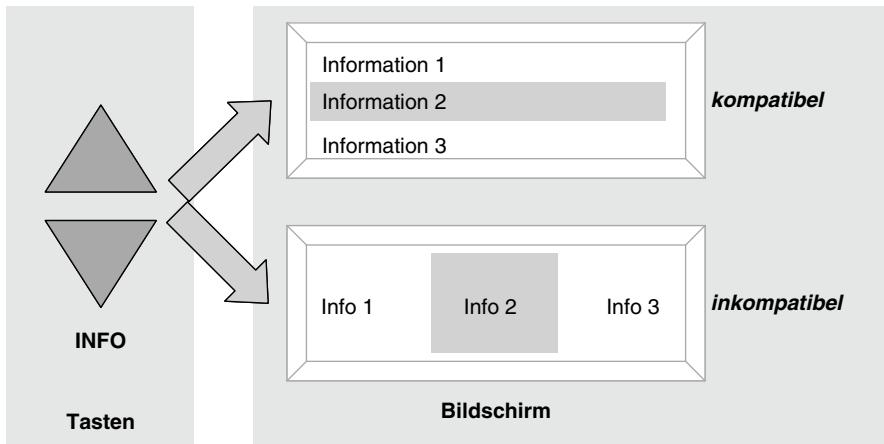


Abb. 6.3 Kompatibilität der Richtung

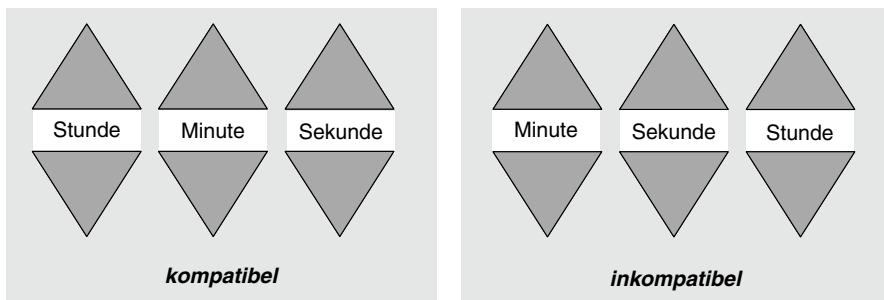


Abb. 6.4 Kompatibilität der Folge

tem ist durch Ähnlichkeit im Erscheinungsbild und im Layout der Komponenten sowie deren Funktionen gekennzeichnet.

- Eine Aktion sollte immer die gleiche Auswirkung haben, unabhängig vom Modus des Bediensystems. Unter gleichen Randbedingungen sollte das System immer gleich reagieren.
- Ein konsistentes Design ist Voraussetzung für ein Bediensystem mit vorhersehbarem Erscheinungsbild und Verhalten z. B. im Hinblick auf Informationsdarstellung, Informationsmanipulation und Navigation.
- Konsistenz des Bediensystems muss gegeben sein und sollte auch systemübergreifend angestrebt werden.

Bei einer Zeitschaltuhr wird beispielsweise eine einheitliche und durchgängige Gestaltung (= Konsistenz) angestrebt in den:

Darstellungen: Die Uhrzeit befindet sich immer an derselben Stelle, ganz gleich in welchem Modus.

Strukturen: ungerade Programmnummern sind immer Einschaltzeiten, gerade Programmnummern immer Ausschaltzeiten.

Begriffen: „MO“ bezeichnet z. B. immer den Montag und nie den Monat.

Abläufen: Die Uhrzeit wird immer in der Reihenfolge Wochentag, Stunde, Minute eingegeben.

6.2.2.3 Gruppierung

Gruppierung ist eine Art der strukturierten Darstellung und beschreibt die räumlich gemeinsame Anordnung zusammengehöriger Elemente (Abb. 6.5). Kriterien für die Zusammengehörigkeit können sein:

- Inhaltliche und funktionale Zusammengehörigkeit
- Häufigkeit der Bedienung
- Reihenfolge der Bedienung

Abbildung 6.5 zeigt die Gruppierung von Interaktionselementen an einer Maschine.

6.2.3 Gestaltgesetze

Die Gestaltgesetze beschreiben, nach welchen Prinzipien bestimmte Gestaltungselemente vom Menschen wahrgenommen werden. Sie sind aus rein empirischen Betrachtungen/Versuchen entstanden und insgesamt existieren über 100 dieser Gesetze. Sie sind erstmals Anfang des 20. Jahrhunderts im Kontext der kognitiven Psychologie entdeckt und formuliert worden. Im Rahmen der Bediensystemgestaltung können die Gestaltgesetze einen Leitfaden für die Anordnung von Bedienelementen und Gestaltung der Bedienoberfläche insgesamt genutzt werden, um eine möglichst optimale Wahrnehmung durch den Benutzer zu gewährleisten.

6.2.3.1 Ganzheitliches Erfassen

Formzusammenhänge werden in ihrer Gesamtform als Gestalt wahrgenommen, die sich von ihrer Umgebung abhebt. Die einzelnen Teile treten zugunsten des Gesamteindrucks zurück. Auch wenn Teile geändert oder ausgetauscht sind oder wenn Formteile fehlen, ergänzt der Wahrnehmungsvorgang die Gestalt zu einem einheitlichen Ganzen.

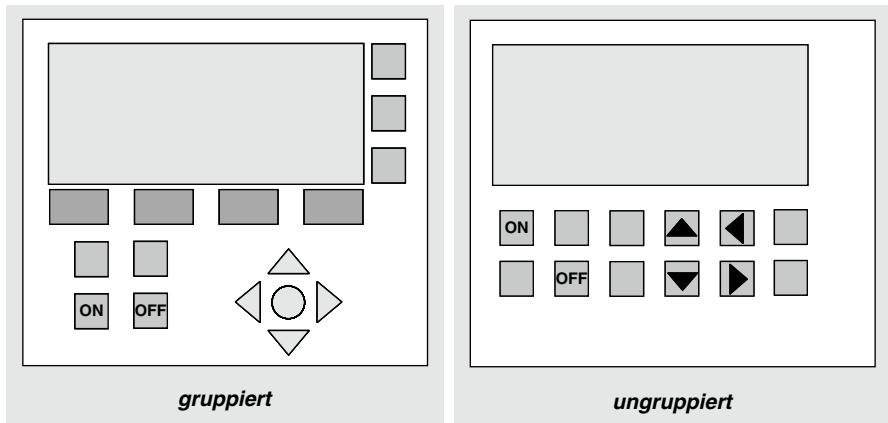


Abb. 6.5 Gruppierung nach funktionaler Zusammengehörigkeit

6.2.3.2 Prägnanz

Formen und Figuren werden bevorzugt wahrgenommen, wenn sie prägnant sind, d. h. wenn sie keinen zufälligen, sondern logischen Aufbau haben und geordnet, einfach und ausgewogen sind, bzw. das Wesentliche einzelner Gegenstände dem Bewusstsein in möglichst klarer, regelmäßiger, optimal ausgeprägter Weise erscheinen lassen. Gefüge, die aus gleichartigen Formteilen zusammengesetzt sind, wirken übersichtlich. Der Mensch hat in seiner Wahrnehmung das Bedürfnis nach Klarheit, Übersichtlichkeit und Ordnung. Komplexe Formen werden vereinfacht und gestaltlose Elemente übersehen (vgl. Abb. 6.6).

6.2.3.3 Nähe

Nahe liegende Formen schließen sich zu einer Einheit zusammen (vgl. Abb. 6.7). Auch ähnliche und ungleiche Formteile haben die Tendenz, durch Nähe als einheitliche Form zu erscheinen. Durch Nähe können Gruppen gebildet werden oder Einzelemente können in Korrespondenz zueinander treten. Von der Wahrnehmung werden die Zwischenräume durch Linien (Brückenlinien) scheinbar gefüllt.

6.2.3.4 Geschlossenheit

Geschlossene Formen lassen sich gegenüber offenen leichter als Figur wahrnehmen. Offene und unvollständige Gefüge werden beim Wahrnehmungsvorgang geschlossen und vervollständigt (s. Abb. 6.8). Was der Mensch als geschlossene Figur sieht, muss nicht von einer Linie begrenzt sein.

Abb. 6.6 Prägnanz – Wahrnehmung eines Dreiecks mit einem rechteckigen Balken statt zweier unregelmäßiger Vielecke

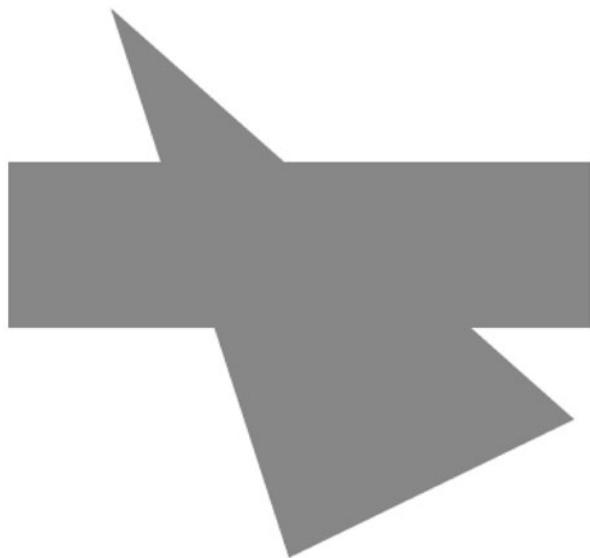


Abb. 6.7 Nähe



Abb. 6.8 Geschlossenheit



Abb. 6.9 Figur-Grund-Beziehung. (Rachui 1998)



6.2.3.5 Figur-Grund-Beziehung

Das menschliche Wahrnehmungssystem übermittelt bei der Aufgliederung des Wahrnehmungsbereiches nur das Wesentliche an das Gehirn und entlastet damit den kognitiven Bereich. U. a. wird dazu das Wahrnehmungsfeld in Figur und (Hinter-)Grund aufgeteilt. Eine Figur, die Hintergrund und Zwischenraum sein kann, hebt sich von ihrem Umfeld ab, wenn sie durch Farbe, prägnante Form, Größe, Helligkeit oder Bewegung auffällt. Figur und (Hinter-)Grund stehen in Beziehung zueinander, können aber nicht gleichzeitig wahrgenommen werden. Bei nicht eindeutiger Figur-Grund-Beziehung entscheidet die Wahrnehmung nach dem für sie leichter Erkennbaren (vgl. Abb. 6.9).

6.2.3.6 Sinnbedeutung

Die Wahrnehmung versucht, der Form oder Gestalt einen Sinn zu geben, sie trennt das Gestalthafte vom Formlosen. Das Wesentliche wird aussortiert und erhält eine Bedeutung. Ist die Bedeutung einmal erkannt, wird sie schnell wieder erkennbar. Dem Menschen fällt es schwer sich von der einmal gebildeten Vorstellung zu lösen (vgl. Abb. 6.10). Das Bekannte wird dem Unbekannten gegenüber vorgezogen. Das Wahrgenommene, das der konkreten Vorstellung zugeordnet wird, bezieht sich auf die individuelle, menschliche Seherfahrung, auf das Bekannte, das es möglich macht, der Gestalt einen Sinn zu geben.

Abb. 6.10 Sinnbedeutung.
(Birbaumer 1991)

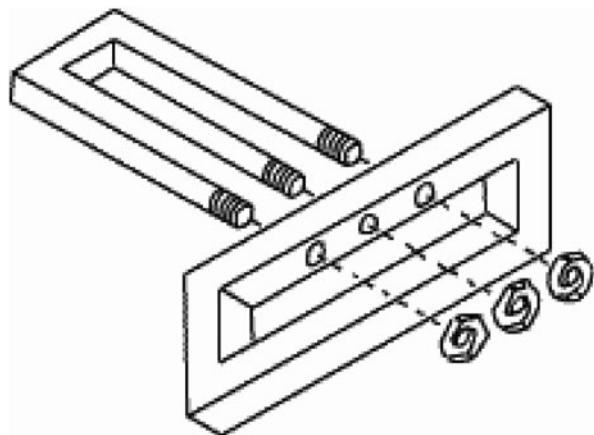
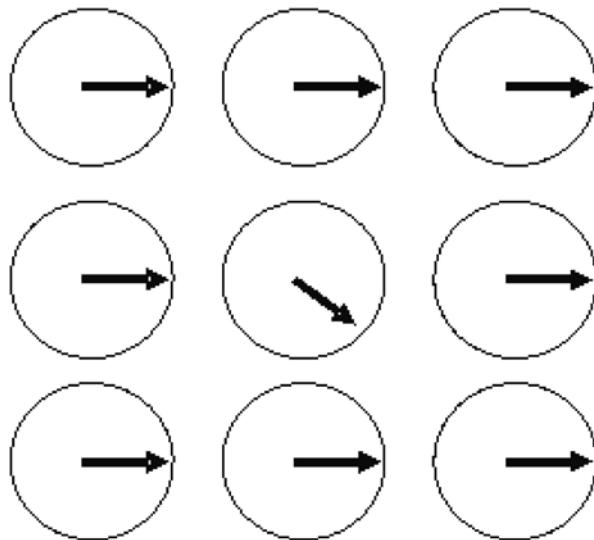


Abb. 6.11 Gleichheit



6.2.3.7 Gleichartigkeit (Gleichheit)

Elemente einer bildlichen Darstellung erscheinen auch über beträchtliche Zwischenräume hinweg als zusammengehörig, wenn Gleichartigkeit hinsichtlich Farbe (wirkt am stärksten), Form, Größe und Verhalten (Bewegung) vorhanden ist. Weicht ein Element stark von den anderen ab, wird es sehr schnell erkannt (s. Abb. 6.11).

6.2.3.8 Trennungsverfahren

Um einige Elemente oder Gruppen aus der Benutzeroberfläche herauszuheben oder zu trennen, können mehrere Verfahren angewendet werden:

- Trennung durch Abstand
- Abheben durch Farbe
- Markieren durch Form
- Veränderung der Größe
- Einziehen von Trennlinien

6.3 Interaktionsgestaltung

Neben der Anordnung der Elemente auf dem Bildschirm ist auch die ergonomische Gestaltung der Interaktion, also der eigentlichen „Kommunikation“ zwischen Benutzer und System ein wichtiger Aspekt, der zur Benutzbarkeit einer Maschine beiträgt. Im Folgenden werden einige für technische Systeme wichtige Interaktionsformen erläutert, die Interaktionselemente für die Eingabe von Informationen in das System diskutiert und ein Überblick über gängige Interaktionstechnologien zur Ein-/Ausgabe von Informationen gegeben.

6.3.1 Interaktionsformen

Mit Interaktionsform ist die softwareabhängige Form der Informationseingabe in das Bediensystem gemeint (Oberquelle 1994). Allgemein können Interaktionsformen also als das bezeichnet werden, was der Nutzer bei der Durchführung einer Interaktionsaufgabe auf dem Bildschirm sieht. Die Interaktionsform muss an die jeweilige Arbeitsaufgabe und die entsprechenden Rahmenbedingungen angepasst sein. Im Folgenden werden die wichtigen Interaktionsformen Kommandosprache, Formulare, Windows-Icon-Menu-Pointer und die direkte Manipulation näher erläutert.

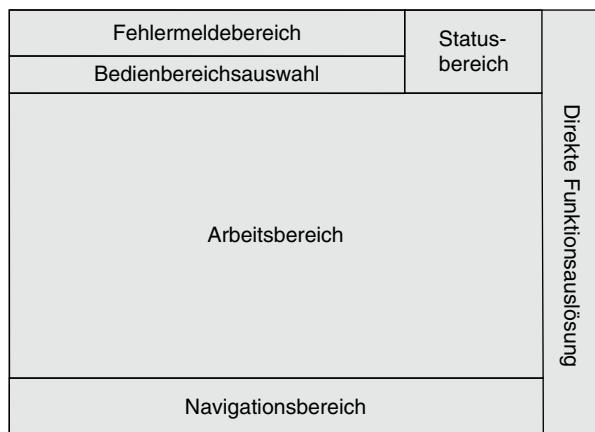
6.3.1.1 Kommandosprache

Die Bedienung von Maschinen mittels der direkten Übermittlung von Kommandos stellte eine der ersten weit verbreiteten Interaktionsformen dar. Der Befehlssatz ist dabei eingeschränkt auf einen vordefinierten Satz von Anweisungen und wird in Form von Funktionstasten, einzelnen Buchstaben, Abkürzungen oder textuellen Kommandos direkt in der Maschine eingegeben.

Diese Interaktionsform findet heute noch Einsatz beim Remote-Zugriff auf Systeme zum Beispiel zum Zwecke der Fernwartung oder als Ergänzung zu menübasierten Systemen, um einen schnelleren Zugriff zum System für erfahrene Nutzer bereitzustellen.

Vorteile dieser Interaktionsform sind der direkte Zugang zur Systemfunktionalität und damit einhergehende Schnelligkeit der Bedienung, sowie die erhöhte Fle-

Abb. 6.12 Beispielhaftes Maskenlayout eines Maschinenbediensystems



xibilität, die durch eine Kombination von mehreren Kommandos erreicht werden kann. Problematisch ist die Benutzbarkeit und Erlernbarkeit zu bewerten. Der Kommandosatz unterscheidet sich von System zu System und muss auswendig gelernt werden.

6.3.1.2 Maskendialoge (Formulare)

Bei Maschinenbediensystemen können im Gegensatz zum Bürobereich Aktionen auch von der Maschine bzw. dem technischen System ausgehen. Aus diesem Grund spielt die Kontrolle über den Prozess eine wesentliche Rolle. Früher wurden Kommandodialoage an dieser Stelle eingesetzt, heute erfüllen Maskendialoge die gestellte Aufgabe am besten. Der Vorteil von Masken gegenüber Kommandosprachen liegt in der kontinuierlichen Anzeige des Systemzustands. Abbildung 6.12 zeigt ein beispielhaftes Maskenlayout für ein Maschinenbediensystem.

Als Masken werden zweidimensionale Anzeigen des Zustands von interaktiven Systemen bezeichnet, die die gesamte Bildschirmfläche belegen. Der Bildschirm kann in einzelne Bildbereiche aufgeteilt werden. Den jeweiligen Bildbereichen werden bestimmte eigene Funktionen und Aufgaben zugeordnet. Sie lassen sich grob unterteilen in Anzeige des Status (informiert über den Systemzustand z. B. Bedienbereich und aktueller Zugriffslevel), Bildbereich (meist in Form von Formularen), Steuerungsbereich (z. B. Softkeyleiste) und Meldebereich. Eine detaillierte Beschreibung der unterschiedlichen Bereiche ist nachfolgend zu finden:

Fehlermeldebereich

- Aufgabe und Bedeutung

Störungsinformationen beschreiben Zustände eines Systems, bei denen eine oder mehrere Funktionen ganz ausgefallen oder beeinträchtigt sind. Diese Informa-

tionen dürfen auf dem Bildschirm nicht überdeckt werden und müssen gegebenenfalls vor anderen, weniger wichtigen Informationen dargestellt werden. Sie lassen sich in Alarme, Störungen und Warnungen unterteilen (s. Kap. 3.4.6)

- Darstellung

Da Störungen unter Umständen ein unmittelbares Eingreifen des Benutzers erfordern, besitzen sie für den Benutzer höchste Priorität. Sie sollte sich von den übrigen Bereichen deutlich abheben (Farbe, Schrift, Einrahmungen) und darf keinesfalls überdeckt werden (DIN 19235).

Generell gilt, dass der Mensch im westlichen Kulturkreis dem oberen Bildschirmbereich eine höhere Aufmerksamkeit widmet als dem unteren Bildschirmbereich. Der Störmeldebereich wird entsprechend seiner Bedeutung für den Benutzer am oberen Rand des Bildschirms positioniert, wo der Fokus der größten Aufmerksamkeit des Nutzers liegt. Da hier in der Regel keine Bedienung erfolgt (Ausnahmen sind redundante kurzzeitige Bedienhandlungen), besteht keine Gefahr der Überdeckung von angezeigten Informationen durch die Hand des Nutzers.

Statusbereich

- Aufgabe und Bedeutung

Der Statusbereich visualisiert permanent den Zustand von Auftrag, Maschine und Prozess. Primär ist der Statusbereich ein Anzeigebereich. Optional können redundante Bedienhandlungen wie das Einstellen eines Wertes über diesen Bereich durchgeführt werden. Mit seiner Hilfe kann der Nutzer außerdem die Maschine identifizieren und sich orientieren.

- Darstellung

Der Statusbereich wird im oberen Bereich des Bildschirms positioniert, wo der Fokus der größten Aufmerksamkeit des Nutzers liegt. Damit ist der Statusbereich hinreichend hoch im Aufmerksamkeitsbereich des Nutzers. Da hier in der Regel keine Bedienung erfolgt, besteht keine Gefahr der Überdeckung von angezeigten Informationen durch die Hand des Nutzers. Ebenso macht die Position oberhalb der Bedienbereichsauswahl dem Anwender deutlich, dass der Inhalt nicht auf den speziellen Bedienbereich bezogen ist.

Bedienbereichsauswahl

- Aufgabe und Bedeutung

Bei einer Aufteilung des Bediensystems in verschiedene unabhängige Bedienbereiche, die unterschiedliche Kontexte des Nutzers im Rahmen der Mensch-Maschine-Interaktion darstellen, ist eine geeignete Umschaltmöglichkeit vorzusehen, die einerseits den aktuell angewählten Bedienbereich darstellt und andererseits einen schnellen Wechsel ermöglicht.

- Darstellung

Die Position des Bereichs zum Bedienbereichsauswahl erfolgt im oberen linken Bildschirmbereich unterhalb des Störmelde- und Statusbereichs. Damit erfolgt bei der Bedienung keine wesentliche Überdeckung von wichtigen Bildinhalten und die Tasten sind schnell zugreifbar. Außerdem befinden sich die Schaltflächen in einem Bereich mit ausreichend hoher Aufmerksamkeit des Nutzers.

Navigationsbereich

- Aufgabe und Bedeutung

Die Navigation beschreibt alle Hilfsmittel, die zur Erschließung der Informationen und zur Nutzung der maschinenspezifischen Funktionen erforderlich sind. Mit Hilfe der Navigation bewegt sich der Benutzer innerhalb der Informations- und Funktionsstruktur. In der Industrieautomation kommen diese Bereiche kaum vor, da die Masken hier vorgegeben sind.

- Darstellung

Die Position des Bereichs zur Navigation erfolgt am unteren Bildschirmrand aus folgenden Überlegungen:

- Bei der Touchbedienung erfolgt keine Überdeckung von wichtigen Bildinhalten.
- Es entstehen keine Konflikte für Rechts- bzw. Linkshänder (im Gegensatz zur Anordnung am rechten oder linken Bildschirmrand).
- Die längere Horizontalachse des Bildschirmformats entspricht der breiten Menüstruktur besser als die kurze Vertikalachse.
- Die restliche Bildschirmfläche (Arbeitsbereich) kann in ihrer vollen Breite für darzustellende Informationen genutzt werden. Die darzustellenden Informationen (z. B. Maschinendarstellungen) sind meist eher breit als hoch.

Globaler Aktionsbereich (direkte Funktionsauslösung)

- Aufgabe und Bedeutung

Der Aktionsbereich für globale Funktionen dient zur Aktivierung von direkten Bedien-Funktionen, die unabhängig von der aktivierte Maske, aber abhängig vom Zustand der Maschine sind. Hierzu gehören z. B. Funktionen wie Stopp und Aggregat-Funktionen beim Betrieb sowie Start und Klappen-Öffnen/Schließen beim Stillstand der Maschine.

- Darstellung

Dem Benutzer zur Verfügung stehende Funktionen werden aktiv dargestellt. Die Sichtbarkeit der Funktionen zu jeder Zeit dient dem konsistenten Erscheinungsbild.

Die Positionierung des globalen Aktionsbereichs kann abhängig von der Händigkeit des Nutzers (z. B. Abfrage über Log-In) am rechten oder linken Bildschirmrand erfolgen. Für Rechtshänder wird der globale Aktionsbereich am rechten Bildschirmrand positioniert, für Linkshänder am linken Bildschirmrand

(da „nur“ ca. 10–15 % der Bevölkerung linkshändig ist, könnte evtl. auf diese Funktionalität verzichtet werden). Eine Position des globalen Aktionsbereichs am unteren Rand des Arbeitsbereichs ist nicht sinnvoll, da hier keine Abgrenzung zum Navigationsbereich erfolgen würde. Zudem könnte bei Bedienhandlungen in diesem Bereich der Navigationsbereich überdeckt werden. Bei einer Position am oberen Rand des Arbeitsbereichs würde der gesamte Arbeitsbereich bei Touchbedienung überdeckt werden.

Arbeitsbereich

- **Aufgabe und Bedeutung**

In diesem Bereich wird die eigentliche Aufgabe bearbeitet. Hier werden die kontextspezifischen Funktionen und Informationen dargestellt, die der Nutzer zur Erledigung seiner Aufgaben benötigt. Er ist gleichzeitig Anzeige- und Bedienbereich. Bedient werden hier die einzelnen Funktionen des aktiven Kontexts. Die Inhalte des Arbeitsbereichs sind abhängig von der ausgewählten Menüebene. Da der Inhalt veränderlich ist, muss der Nutzer die Möglichkeit haben, sich zu orientieren. Durch die Orientierungsinformation kann er erkennen, wo er sich gerade befindet. Realisiert wird dies z. B. durch eine präzise, sinnvolle Bezeichnung des Kontextes und durch die Angabe eines Pfades, der den übergeordneten Kontext darstellt.

- **Darstellung**

Der Arbeitsbereich ist der größte zusammenhängende Bereich. In ihm wird abhängig von der aktiven Funktionsgruppentaste die spezifische Interaktionsmaske dargestellt. Die Positionierung des Arbeitsbereichs als größten zusammenhängenden Bereichs erfolgt im Zentrum der Bedienoberfläche über dem Navigationsbereich und unter dem Statusbereich.

Im oberen Bereich befindet sich eine Kopfzeile, in der eine präzise, sinnvolle Bezeichnung des Kontextes und des Pfades, der den übergeordneten Kontext darstellt, dargestellt wird.

Ein Beispiel für ein Bediensystem mit den genannten Bereichen findet sich in Abb. 6.13

6.3.1.3 Windows, Icons, Menus, Pointers (WIMP)

Diese Interaktionsform bezeichnet die im PC-Markt dominante Interaktionsform, in der Informationen auf dem Bildschirm in Fenstern (Windows) angezeigt werden und durch Icons repräsentiert werden. Diese Darstellung wird über in Menüs organisierte Kommandos gesteuert, die durch ein Zeigegerät angewählt werden können.

Vorteile dieser Interaktionsform sind die durch die hohe Verbreitung im Heimcomputer-Segment einhergehende Vertrautheit und leichte Erlernbarkeit auch für unerfahrene Nutzer. Im industriellen Umfeld hat sich die WIMP-Interaktionsform bisher aber nicht durchsetzen können, da die meisten Maschinen nicht mit einem Zeigegerät ausgestattet sind (Mäuse sind verschmutzungsanfällig).

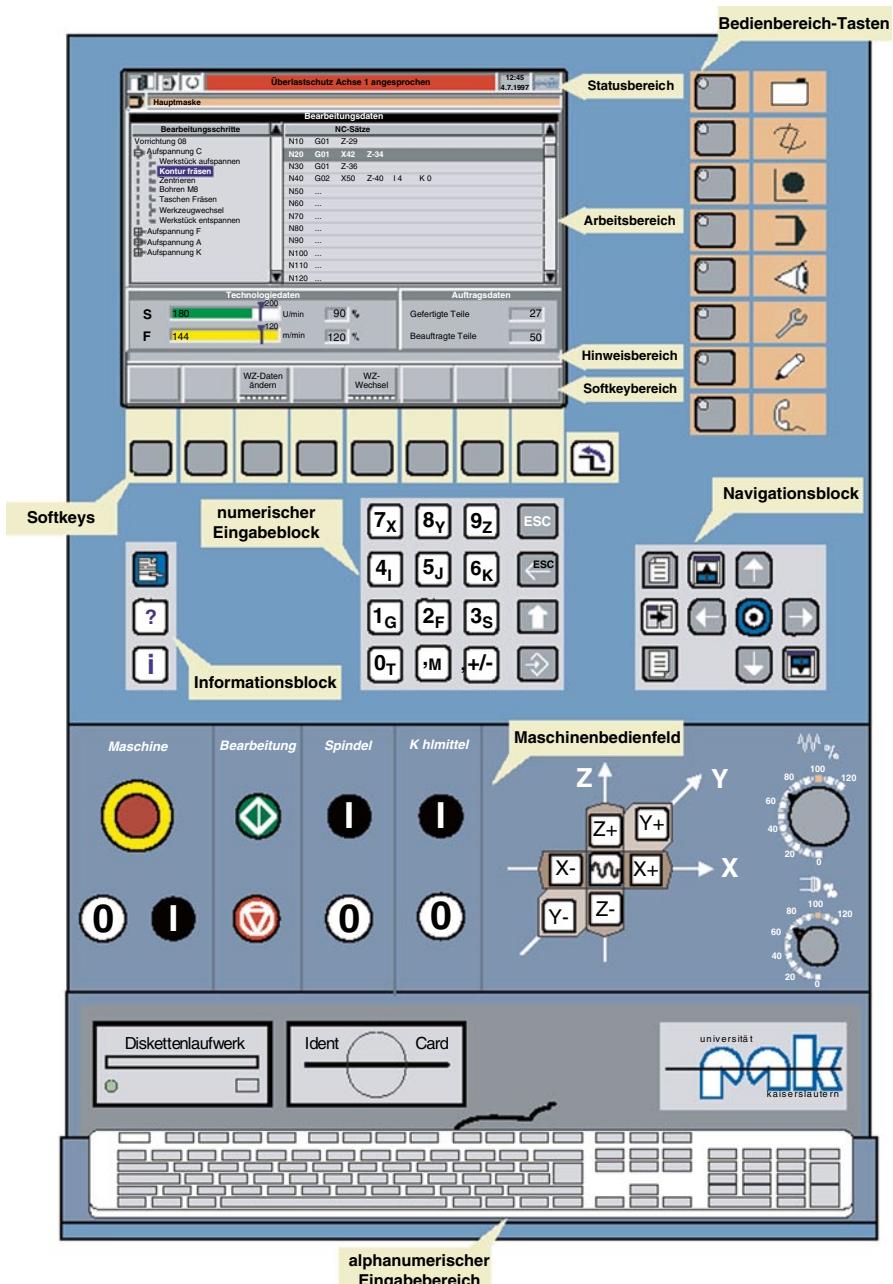


Abb. 6.13 Beispiel für ein Bediensystem. (HÜMNOS 1997)

Eine Abwandlung der WIMP-Oberflächen stellen die sogenannten Point-and-Click Interfaces dar, die mit einem einzelnen Click bereits Aktionen auslösen. Die Verwendung dieser Interaktionsform bietet sich insbesondere für Touchscreens an.

6.3.1.4 Direkte Manipulation

Die Bezeichnung „Direkte Manipulation“ existiert seit etwa Anfang der Achtziger Jahre und bezeichnet eine Interaktionsform, die durch folgende Eigenschaften charakterisiert ist (Dix 2004):

- Alle für den Nutzer relevanten Objekte sind innerhalb des Systems sichtbar
(Beispiel: Ein Betriebssystem verfügt über ein Konzept Zwischenablage. Diese ist innerhalb des Systems visuell als Klemmbrett repräsentiert, dass der Nutzer aufrufen kann und sich die ausgeschnittenen und dort gespeicherten Elemente als „Schnipsel“ ansehen kann.)
- Direkt auf diesen Objekten kann der Nutzer Handlungen ausführen. Die Handlungen sind inkrementell und der Nutzer bekommt ein sofortiges Feedback über die Auswirkungen seiner Handlung.
(Beispiel: Zu löschen Objekte eines Betriebssystems können in einen Papierkorb verschoben werden. Sie verschwinden daraufhin an der ursprünglichen Stelle und das Papierkorbsymbol verändert sich von einem leeren Papierkorb zu einem gefüllten Papierkorb.)
- Jede Handlung kann rückgängig gemacht werden, in dem der Nutzer eine komplementäre Handlung ausführt.
- Jede Handlung ist eine gültige Handlung, es gibt keine syntaktisch inkorrekten Handlungen.
- Komplexe Kommandos werden durch Handlungen ersetzt, die direkt die sichtbaren Elemente manipulieren.
(Beispiel: Das Verschieben einer Datei in einen anderen Ordner wird statt über einen speziellen Befehl unter Angabe der Datei mit vollständigem Pfad und dem Zielpfad durch das einfache „Anfassen“ und Verschieben des entsprechenden Datei-Symbols in den Zielordner realisiert.)

6.3.2 Interaktionselemente

6.3.2.1 Taster (Button)

Bei Aktivierung eines Tasters wird die ihm zugewiesene Aktion ausgeführt. Dabei kann die Aktion je nach Einsatzfall einmal durchgeführt werden wie z. B. bei der Bestätigung für einen Befehl (OK-Button), oder solange aktiviert werden bis die Taste losgelassen wird (Tipp-Betrieb).

Tab. 6.3 Schaltzustände von Tastern

Bedienbar	Der Taster kann betätigt werden und löst eine bestimmte Funktion aus.	
Betätigt	Beim Betätigen des Tasters verändert dieser sein Aussehen, um ein optisches Feedback für die Bedienhandlung zu liefern.	
Nicht bedienbar	Der Taster ist in dieser Situation nicht bedienbar.	

Tab. 6.4 Schaltzustände von Schaltern

Bedienbar und deaktiv	Der Button kann betätigt werden und schaltet dann einen bestimmten Zustand ein.	
Betätigt	Beim Betätigen des Buttons verändert dieser sein Aussehen, um ein optisches Feedback für die Bedienhandlung zu liefern.	
Bedienbar und aktiv	Der Schalter ist eingeschaltet/aktiv und bedienbar.	
Bedienbar und aktiv (Schaltzustand verzögert)	Der Schalter wurde aktiviert, aber der Prozess kann den Schaltzustand erst nach einer bestimmten Zeitspanne herstellen. Das rechte obere Eck blinkt	
Nicht bedienbar und deaktiv	Der Schalter ist in dieser Situation nicht bedienbar und hat den Schaltzustand „Aus“.	
Nicht bedienbar und aktiv	Der Schalter ist eingeschaltet, kann aber nicht bedient werden.	

Wird ein Taster aktiviert (gedrückt), so verändert er sein Aussehen und seinen internen Status. 3D-Buttons sinken in die Oberfläche ab. Zur Kennzeichnung verschiedener Buttons mit gleicher Grundform, können Beschriftungen oder Piktogramme auf den Buttons enthalten sein. Ihr Design ist weitgehend frei, wird jedoch durch Randbedingungen wie Übersichtlichkeit und Erkennbarkeit beeinflusst.

Grundsätzlich lassen sich für den Taster drei Schaltzustände unterscheiden, die in Tab. 6.3 dargestellt sind.

Da Taster ein universell einsetzbares Interaktionselement (Funktionsaufruf, Dialogaktivierung, Maskenwechsel uvm.) sind, können keine Maßbeschränkungen gesetzt werden.

6.3.2.2 Schalter (Button)

Schalter werden im Gegensatz zu Tastern eingesetzt, wenn ein Zustand über die eigentliche Bedienhandlung hinaus aktiviert bzw. deaktiviert werden soll. Das Aktivieren/Deaktivieren setzt immer eine Bedienhandlung des Benutzers voraus.

Schalter unterscheiden sich grundsätzlich nicht von Tastern. Um dem Benutzer die Schalterfunktionalität zu visualisieren, wird eine Analogie aus der Elektrotechnik herangezogen, die LED-Codierung. Hierbei wird am Schaltelement eine Leuchtfäche (rund oder eckig) angebracht, die den Schaltzustand durch Aufleuchten visualisiert. Durch Farb- oder Blinkcodierung kann der Schaltzustand noch detaillierter ausgedrückt werden. Grundsätzlich lassen sich für den Schalter fünf bzw. sechs Schaltzustände unterscheiden (s. Tab. 6.4).

Da Schalter – wie Taster auch – ein universell einsetzbares Interaktionselement (Funktionsaktivierung, Dialogaktivierung, Maskenwechsel, uvm.) sind, können keine Maßbeschränkungen gesetzt werden.

6.3.2.3 Navigationstasten

Navigationstasten entsprechen ihrer Funktion nach der Gruppe der Taster. Daher werden sie ebenso gestaltet und durch ein Symbol abgegrenzt. In Analogie zu den Windows-Menüs werden drei Punkte eingesetzt, um zu verdeutlichen, dass das Drücken des Tasters zu weiteren Informationen führt.

Grundsätzlich lassen sich für Navigationstasten drei Schaltzustände unterscheiden (s. Tab. 6.5).

6.3.2.4 Schaltergruppe

Schaltergruppen mit gegenseitiger Auslösung werden eingesetzt, wenn das Aktivieren eines Schaltzustandes das Deaktivieren eines anderen Schaltzustandes bedingt (s. Tab. 6.6). Schaltergruppen sind für den Benutzer leichter zu verstehen und einfacher.

Tab. 6.5 Schaltzustände von Navigationstasten

Bedienbar	Der Taster kann betätigt werden und löst eine bestimmte Funktion aus.	
Betätigt	Beim Betätigen des Tasters verändert dieser sein Aussehen, um ein optisches Feedback für die Bedienhandlung zu liefern.	
Nicht bedienbar	Der Taster ist in dieser Situation nicht bedienbar.	

Tab. 6.6 Schaltzustände von Schaltergruppen

Schaltergruppe	Das Betätigen eines Schalters setzt den bisher aktiven Schalter zurück. Da Schaltergruppen bevorzugt bei Funktionen von hoher Bedeutung eingesetzt werden, kann eine Hinterlegung der Tasten zur Farbcodierung erfolgen.	
----------------	---	--

cher zu bedienen als z. B. Mehrfachschalter, da für jeden Schaltzustand ein diskreter Schalter existiert.

Im Gegensatz zu den Radiobuttons lösen die Elemente von Schaltergruppen direkt Funktionen im Bediensystem oder an der Maschine aus, währen mit Radiobuttons nur Optionen bzw. Parameter für die Funktionsabarbeitung festgelegt werden.

Schaltergruppen bestehen aus mehreren Schaltern, die horizontal oder vertikal aneinandergereiht werden. Wird ein Schalter gedrückt, wird der bisher aktive Schalter zurückgesetzt. Aussehen und Verhalten ist ansonsten identisch zu den Schaltern.

6.3.2.5 Anzeigen

Zur sachgerechten Bedienung der Maschine benötigt der Benutzer Informationen über Maschinen- und Prozessparameter. Das Maschinen-Bediensystem muss hierzu Werte anzeigen und ggf. deren Manipulation erlauben.

Tab. 6.7 Schaltzustände von Anzeigen

Wertanzeige, passiv	Beschriftung, Wertefeld und Einheit werden durch einen Rahmen gruppiert.	
Wertanzeige, aktiv	Der angezeigte Wert kann manipuliert werden. Als 3D-Eingabefeld ausgeprägt.	
Wertanzeige, deaktiv	Ein Wert wird angezeigt, ist aber in diesem Kontext nicht verstellbar. Anzeige als deakt. 3D-Feld.	
Wertanzeige, aktiviert	Als vertieftes 3D-Eingabefeld mit schwarzer Schrift ausgeprägt.	
Wertanzeige (bewertet)	Ein Wert wird angezeigt und Bedingungen entsprechend bewertet.	

Reine Anzeigen sind nicht bedienbare Elemente und daher in 2D dargestellt. Sofern der Benutzer den Anzeigewert durch numerische oder alphanumerische Eingaben verändern kann, wird das Element in 3D ausgeführt.

Eine Wertanzeige umfasst immer eine Wertbezeichnung, den Wert selbst und, sofern erforderlich, eine Einheit. Die Beschriftung und die Einheit stellen nicht veränderliche Informationen dar, so dass diese auf dem Hintergrund platziert werden. Der Wert ist eine vom System oder Benutzer veränderliche Information und wird daher in einem weißen Wertefeld angezeigt. Zahlen werden immer rechts angeordnet, dies vereinfacht die Informationsaufnahme.

Neben der absoluten Wertänderung durch den numerischen oder alphanumerischen Eingabeblock existiert die Möglichkeit, den Wert inkrementell zu verändern. Hierzu dient der an dem Bedienfeld angebrachte Dreh-Druck-Steller. Trotz einer internen Unterscheidung zwischen Soll- und Ist-Werten werden dem Benutzer nur die Istwerte präsentiert und wenn er eine Manipulation vornimmt, erfolgt diese am entsprechenden Soll-Wert. Eventuelle Zwischenzustände, wie sie beispielsweise eintreten, wenn eine Heizung mehrere Minuten braucht, um den Soll-Wert zu erreichen, werden durch Blinken des Anzeigewerts dargestellt.

In Tab. 6.7 sind die unterschiedlichen Interaktionselemente vom Typ „Anzeige“ aufgelistet und deren Schaltzustände dargestellt.

Die Größe der Anzeige- und Eingabeelemente kann maskenspezifisch angepasst werden. Hierbei sollte dem Entwickler eine gewisse Freiheit eingeräumt werden, allerdings immer im Hinblick auf eine konsistente und nutzergerechte Gestaltung der Masken.

Tab. 6.8 Schaltzustände von Auswahlfeldern

Bedienbar und nicht selektiert	Die Checkbox kann betätigt werden und selektiert eine bestimmte Option ein.	<input type="checkbox"/> Beschriftung
Betätigt	Beim Betätigen der Checkbox verändert diese ihr Aussehen, um ein optisches Feedback für die Bedienhandlung zu liefern.	<input checked="" type="checkbox"/> Beschriftung
Bedienbar und selektiert	Die Checkbox ist selektiert und bedienbar.	<input checked="" type="checkbox"/> Beschriftung
Nicht bedienbar und nicht selektiert	Die Checkbox ist in dieser Situation nicht bedienbar und ist nicht selektiert.	<input type="checkbox"/> Beschriftung
Nicht bedienbar und selektiert	Die Checkbox ist selektiert, kann aber nicht bedient werden.	<input checked="" type="checkbox"/> Beschriftung

6.3.2.6 Auswahlfeld (Checkbox)

Auswahlfelder werden eingesetzt, wenn der Benutzer Optionen auswählen kann. Bei der Auswahl sind alle Auswahlkombinationen möglich. Im Gegensatz zum Schalter wird bei der Aktivierung eines Auswahlfeldes keine Funktion ausgelöst, sondern nur eine Option für die Funktionsverarbeitung selektiert.

Übertragen auf die Anwendung wird die Funktion mit einem Taster ausgelöst, nachdem zuvor mit Hilfe von Auswahlfeldern die Parameter gesetzt wurden, die bei der Funktionsausführung von Bedeutung sind.

Auswahlfelder sind 3D-Felder, die mit einer Beschriftung versehen sind. Sie sind im Allgemeinen quadratisch und im selektierten Zustand mit einem Häkchen versehen. Auswahlfelder können nur einen von zwei verschiedenen Schaltzuständen einnehmen (Ein/Aus).

Kommen mehr als zwei Auswahlfelder für denselben Funktionskontext zum Einsatz, so sind sie durch einen Rahmen zu gruppieren.

Grundsätzlich lassen sich für Auswahlfelder fünf Zustände unterscheiden (s. Tab. 6.8).

6.3.2.7 Exklusiv-Auswahlfeld (Radiobutton)

Exklusiv-Auswahlfelder werden eingesetzt, wenn der Benutzer aus einer Gruppe von mindestens zwei Optionen genau eine Option auswählen kann. Es ist nur die Auswahl einer einzigen Option möglich. Im Gegensatz zum Schalter wird bei der Selektion eines Exklusiv-Auswahlfeldes keine Funktion ausgelöst, sondern nur eine Option für die Funktionsverarbeitung eingestellt.

Exklusiv-Auswahlfelder sind 3D-Felder, die mit einer Beschriftung versehen sind. Sie sind kreisförmig mit weißem Hintergrund und im selektierten Zustand

Tab. 6.9 Schaltzustände von Exklusiv-Auswahlfeldern (Checkbox)

Bedienbar und nicht selektiert	Der Radiobutton kann betätigt werden und selektiert eine bestimmte Option. Dabei wird der bisher selektiert Radiobutton der Gruppe zurückgesetzt.	 Beschriftung
Betätigt	Beim Betätigen des Radiobuttons verändert dieser sein Aussehen, um ein optisches Feedback für die Bedienhandlung zu liefern.	 Beschriftung
Bedienbar und selektiert	Der Radiobutton ist selektiert und bedienbar.	 Beschriftung
Nicht bedienbar und nicht selektiert	Der Radiobutton ist in dieser Situation nicht bedienbar und hat den Schaltzustand „Aus“.	 Beschriftung
Nicht bedienbar und selektiert	Der Radiobutton ist eingeschaltet, kann aber nicht bedient werden.	 Beschriftung

schwarz gefüllt. Exklusiv-Auswahlfelder können nur einen von zwei verschiedenen Schaltzuständen einnehmen (Ein/Aus). Wird eine Option gewählt, so wird diese selektiert und die bisher selektierte Option in der Gruppe zurückgesetzt.

Eine Gruppe von Exklusiv-Auswahlfeldern sollte sich immer nur auf einen Funktionskontext beziehen und durch einen Rahmen gruppiert werden. Auch wenn die Checkbox durch Antippen der Beschriftung aktiviert werden kann, wird diese nicht durch einen 3D-Rahmen – analog zum Taster – hervorgehoben. Da mit der Aktivierung keine Funktion ausgelöst wird, stellt dies für den Nutzer kein Problem dar und er kann die Auswahlfelder nicht mit Schaltern verwechseln, bei denen die weiße Leuchtdiode den Auswahlfeldern ähnlich sieht. Grundsätzlich lassen sich für Exklusiv-Auswahlfelder fünf Schaltzustände (s. Tab. 6.9) unterscheiden.

6.3.2.8 Register (Tab control)

Die Register nutzen eine Metapher aus dem Bürobereich. Hier werden Registerblätter eingesetzt, um z. B. in Ordnern Papierdokumente strukturiert abzulegen. Da bei industriellen Anwendungen regelmäßig auf sehr viele Informationen auf zumeist sehr begrenztem Bildschirmraum zugegriffen werden muss, ist diese Analogie für den Benutzer leicht nachzu vollziehen. Der Benutzer hat außerdem ständig den Überblick über die möglichen Bildschirmteile und kann schnell auf diese zugreifen.

Die Registerlaschen werden ober- oder unterhalb der Anzeigefläche gruppiert, die mit ihrer Hilfe bedient wird. Sie sind als 3D-Elemente ausgeprägt und die jeweils aktive Registerlasche wird hell oder farbig hervorgehoben und verbunden mit dem Rand des Anzeigebereiches dargestellt (Tab. 6.10).

Tab. 6.10 Register

Register	<p>Durch Anklicken der Registerlasche wird die entsprechende Bildschirmsicht aufgerufen.</p>	<input type="button" value="Produktion"/> <input type="button" value="Diagnose"/> <input type="button" value="Hilfe"/> <input type="button" value="Konfiguration"/>
----------	--	---

Aus der Praxis: „Identifikation der benötigten Interaktionsform“

Grundlage für die Gestaltung der Visualisierungskomponenten ist die Frage, welche Aufgabe der Nutzer durchzuführen hat. Diese Frage sollte auf einer hohen Abstraktionsebene gestellt werden, um daraus Klassen abzuleiten. Beispiele für diese Aufgaben sind:

- Aktivieren/Deaktivieren eines Zustandes
- Funktionsauslösung
- Schrittweise Veränderung eines angezeigten Wertes
- Exklusive Auswahl zwischen verschiedenen Optionen

Jede Aufgabe lässt sich zumeist noch funktional untergliedern. Diese funktionale Untergliederung führt zu den Bestandteilen einer Visualisierungskomponente:

Die Aufgabe „Aktivieren/Deaktivieren eines Zustandes“ hat als Bestandteile die Aktivierung eines deaktiven Zustandes und die Deaktivierung eines aktiven Zustandes.

„Ableitung von Visualisierungs-Klassen“

Aus den identifizierten Interaktionsformen lassen sich Visualisierungs-Klassen ableiten, die sich für die spezifischen Anforderungen der Bedienaufgabe eignen. Hierzu können Analogien aus der realen Welt oder aber bereits bekannte Interaktions-Klassen herangezogen werden.

Für die Aufgabe „Aktivieren/Deaktivieren eines Zustandes“ eignet sich z. B. die Visualisierungs-Klasse „Schalter“.

„Repräsentationsform entwerfen“

Beim Entwurf der Repräsentationsform der Visualisierungskomponente muss neben der Funktionsrealisierung auch die Funktionsvisualisierung berücksichtigt werden (s. Abb. 6.14). Der Nutzer muss die Zustände und Interaktionsmöglichkeiten der Visualisierungskomponente erkennen können. Außerdem muss Platz reserviert werden, der es erlaubt, den Kontext, also die Komponente zu visualisieren, auf die sich die Bedienaufgabe bezieht.

Auf diese Weise müssen vor der Realisierung des Bediensystems alle Bedienaufgaben in Form von Visualisierungskomponenten repräsentiert werden. Diese können in Form einer Bibliothek abgelegt werden und dann an den spezifischen Einsatzort angepasst werden.

Abb. 6.14 Repräsentationsform „Schalter“

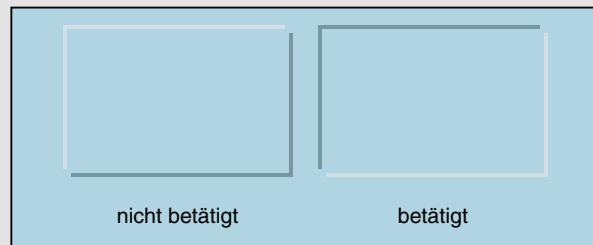
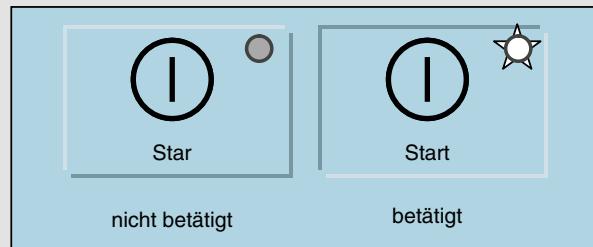


Abb. 6.15 Spezifische Ausprägung des „Schalters“



„An spezifischen Einsatz anpassen“

Die Anpassung an den spezifischen Einsatz umfasst das Verknüpfen der Visualisierungskomponente mit dem Kontext. Dies geschieht insbesondere durch die Platzierung z. B. in unmittelbarer Nähe der betroffenen Komponente auf dem Display. Durch Text- oder Bildinformationen, die der Visualisierungskomponente zugeordnet werden, wird dem Nutzer dessen Funktion vermittelt (Abb. 6.15).

6.3.3 Interaktionstechnologien

Interaktionstechnologien lassen sich unterteilen in Eingabe- und Ausgabetechnologien und solchen, die Ein- und Ausgabe vereinen (z. B. Touchscreen). Im Folgenden werden zunächst einige Eingabetechnologien vorgestellt, dann detailliert auf die Touchscreen-Technologie und ihre Einsatzmöglichkeiten im industriellen Umfeld eingegangen und abschließend aktuelle Ausgabetechnologien diskutiert.

6.3.3.1 Tastaturen und Tasten

Tastaturen und einzelne Tasten bezeichnet man als nicht koordinatengebende Interaktionsgeräte. Sie werden zur Eingabe von textuellen Daten (Buchstaben oder Zahlen) sowie Kommandos verwendet.

Tasten existieren in verschiedensten Bauformen (z. B. Folientasten, Silikon-schaltmatten), meist werden jedoch durch eine mechanische Einwirkung durch den

Finger Kontakte bewegt, wodurch ein elektrischer Kreis geschlossen oder geöffnet und daraufhin eine Funktion ausgelöst wird. Nach dem Drücken der Taste geht diese wieder in ihre Ausgangslage zurück.

Als Spezialfall können die häufig eingesetzten Softkeys betrachtet werden. Hierbei handelt es sich um normale Tasten, die sich um einen Bildschirm gruppieren oder um Tastenfelder eines Touchscreens, deren Belegung sich abhängig vom dargestellten Bild ändern kann. Dies bietet den Vorteil, dass sich die Anzahl der Tasten insgesamt reduzieren lässt, da eine Hardware-Taste mehrere Funktionen übernehmen kann, ohne dass die Übersichtlichkeit und Benutzbarkeit darunter leidet.

6.3.3.2 Zeigegeräte

Koordinatengebende Interaktionsgeräte lassen sich in ein- (z. B. Handrad), zweidimensionalen (z. B. Maus) und mehrdimensional (z. B. 6D-Maus) wirkende Interaktionsgeräte einteilen

Im Folgenden werden einige koordinatengebende Eingabetechniken (s. Abb. 6.16) kurz beleuchtet, nähere Informationen zur Auswahl von Interaktionsgeräten im industriellen Umfeld finden sich in (Krauss 03).

- **Maus**

Die Maus ermöglicht zweidimensionale Bewegungen (i. d. R. auf einer flachen, horizontalen Fläche), die auf dem Bildschirm in einen sich bewegenden Zeiger umgesetzt werden. Die Umsetzung der Signale erfolgt über mechanische, optische oder induktive Sensoren. Während der Bewegung liegt die Hand auf der Maus, was eine große Genauigkeit bei kleinem Zeitaufwand ermöglicht. Mäuse verfügen in der Regel über eine bestimmte Anzahl von Tasten, welche z. B. zur Funktionsauslösung verwendet werden.

Mäuse finden vor allem im Bürobereich Anwendung, da hier die benötigten verschmutzungsarmen, horizontalen Arbeitsflächen vorhanden sind. Im industriellen Umfeld werden Mäuse aufgrund der Verschmutzung und meist nicht vorhandener Rollflächen selten eingesetzt.

- **Trackball**

Beim Trackball wird eine rollbar gelagerte Kugel mit Hilfe der Hand bewegt. Insgesamt gibt es viele Analogien zur Maus. Die Signalaufnahme erfolgt hier ebenfalls mechanisch oder optisch, wodurch ein Zeigeelement auf dem Bildschirm bewegt wird.

Der Trackball benötigt keine ebene Fläche auf der er bewegt werden muss und eignet sich somit (bei entsprechender Kapselung) auch für den industriellen Einsatz. Durch die Kombination mit Tasten ist auch beim Trackball eine Funktionsauslösung möglich.

- **Mousepad (Fingermaus)**

Das Mousepad wird direkt mit dem Finger bedient. Dabei wird die Position des Fingers auf einer Sensorfläche erfasst und an ein Zeigeelement auf dem Bildschirm weitergegeben. Schaltelemente befinden sich unter oder neben der Sensorfläche.



Abb. 6.16 Koordinatengebende Eingabegeräte (v. o. l.: Maus, Trackball, Joystick, Mousepad, Mousestick, Mousebutton, Grafiktablett). (Quellen: LOGITECH, MICROJOYSTICK, MOBILE BUTTON, SMARTTABLET)

Mousepads eignen sich aufgrund nicht vorhandener beweglicher Teile gut für mobile Anwendungen. Außerdem können sie im industriellen Umfeld eingesetzt werden, da sie nicht sehr verschmutzungsanfällig sind.

- **Joystick**

Der Joystick kann in zwei Freiheitsgraden ausgelenkt werden, wobei entweder die Kraft (isotonisches Wirkprinzip) oder die Auslenkung (isometrisches Wirkprinzip) gemessen werden kann. Der Joystick kann meist mit der ganzen Hand umfasst werden und bietet somit eine gute Kontrolle. Schaltelemente werden sowohl am Hebel als auch am Fuß des Joysticks angebracht.

Der Joystick findet vor allem Anwendung bei Zielverfolgungsaufgaben oder zum Steuern von räumlichen zweidimensionalen Bewegungen.

- **Mousestick (Maus-Steuerknüppel) und Mousebutton (Maus-Knopf)**

Der Mousstick und der Mousebutton sind an die Funktionsweise eines Joysticks angelehnt. Hierbei wird das Zeigeelement analog zur Kraftaufbringung oder zur Auslenkung des Knopfes am Interaktionsgerät bewegt. Beim Mousebutton handelt es sich um einen flachen Knopf, während der Mousestick an einen kleinen Joystick erinnert. Schaltelemente werden um diese Elemente angeordnet.

Beide Eingabegeräte können im industriellen Umfeld eingesetzt werden, da sie keine nennenswerten beweglichen Teile haben und durch Kapselung auch verschmutzungsunempfindlich gehalten werden können.

- **Grafiktablett**

Auf einem Grafiktablett wird mittels eines Zeigestiftes innerhalb eines Positionsreichs und eines Schaltbereichs navigiert. Im Positionsreich wird das Zeigeelement auf dem Bildschirm bewegt und im Schaltbereich werden Aktionen ausgelöst. Eingesetzt werden Grafiktablets vor allem im CAD-Bereich.

6.3.3.3 Touchscreens

Derzeit ist bei Hardware-Konzepten in industrieller Umgebung ein deutlicher Trend festzustellen, die Touchscreen-Technologie vermehrt einzusetzen.

Bei der Touchscreen-Technologie erfolgt die Interaktion mit dem technischen System direkt durch Berühren des Bildschirms. Sie erlaubt dem Anwender eine Interaktion mit dem technischen System ohne die Verwendung von Tastatur und Maus. Der Finger wird somit buchstäblich mit dem in Verbindung gebracht, was auf dem Bildschirm dargestellt wird. Ausgelöst wird eine Funktion durch die unmittelbare Berührung einer sensiblen Bildschirmoberfläche durch den Nutzer, was eine intuitive Bedienung ermöglicht. Die sogenannten Multi-Touch-Bildschirme haben in den letzten Jahren die Bedienmöglichkeiten erheblich erweitert, da sie die mehrere Berührungen gleichzeitig erkennen können und somit die Manipulation von Objekten mittels verschiedener Gesten erlauben. Diese einfache und nutzerfreundliche Form der Bedienung ist ideal für Menschen, die nicht unbedingt mit Computern vertraut sind. Die Touchscreen-Technologie kann somit eingesetzt werden, um schnelleres Lernen oder einfachere Informationsbeschaffung zu ermöglichen.

Deshalb ergeben sich für Touchscreens weitreichende Einsatzgebiete. Sie sind sowohl im industriellen als auch im Labor-Bereich zunehmend wirtschaftlich interessant. Die grafisch-interaktive Benutzung erleichtert die einfache Handhabung von Systemen. Beliebige Schalt- bzw. Touchflächen können unter Beachtung gewisser Gestaltungsrichtlinien frei auf der Oberfläche platziert werden. Grafisch gestaltete Bildschirmmasken mit logisch zugeordneten Touchobjekten erhöhen die Anschaulichkeit. Dieser Aspekt führt auch zu einem immer größeren Einsatz der Touchscreen-Technologie im Konsum-Bereich (z. B. Fahrkartautomaten). So mit eignen sich Touchscreens generell für Systeme, die von einer vielfältigen Nutzerschicht bedient werden.

Vor- und Nachteile eines Touchscreens Nachfolgend werden die Vorteile des Touchscreens den Nachteilen gegenübergestellt. Aus den Nachteilen ergeben sich die Randbedingungen für den Touchscreen-Einsatz, die in einem späteren Kapitel erläutert werden.

Vorteile

- Direkte Bedienung

Die Bedienung des Systems erfolgt direkt durch Berühren des Bildschirms. Ausgelöst wird eine Funktion durch den Fingerdruck des Bedieners. Dadurch wird eine intuitive Bedienung ermöglicht. Das Eingabemedium ist gleichzeitig Ausgabemedium, dadurch ist eine direkte Hand-Augen-Koordination möglich. Es ermöglicht eine unmittelbare Rückkopplung zwischen der Eingabe des Nutzers und der Ausgabe des Systems. Für direkt manipulative Anwendungen ist der Touchscreen sehr gut geeignet.

- Freie Platzierung

Anders als bei Bediensystemen ohne Touchscreen sind hier keine zusätzlichen, hardwareseitig ausgeführten Tasten (z. B. Funktions- oder Softkeytasten) notwendig. D. h. auch die auf dem Display abgebildeten Bedienelemente müssen

nicht mehr bei den Hardwaretasten gruppiert werden. Alle zum jeweiligen Zeitpunkt verfügbaren Funktionen werden auf dem Bildschirm angezeigt.

- Flexibilität

Die freie Platzierung der Oberflächenobjekte auf der Displayfläche ermöglicht Flexibilität in der Oberflächengestaltung, Unabhängigkeit von Hardwareelementen und eine optimale Anpassung an die mentale Informationsverarbeitung des Menschen. Dem Menschen stehen immer nur die unmittelbar zur Ausführung seiner Aufgabe nötigen Funktionen zur Verfügung. Die flexible Eingabetechnik über Berühren ist ein dem Menschen angepasstes natürliches Verhalten.

- Nutzerfreundlichkeit

Durch die direkte Bedienung des Touchscreens und das sich intuitiv erschließende Benutzungskonzept entsteht beim Bediener schnell Vertrautheit mit dem System. Durch den notwendigen übersichtlichen Aufbau werden Hemmschwellen beim Anwender beseitigt, so dass die Akzeptanz des Systems sowie die Identifikation mit dem System zunehmen. Daraus resultiert die große Beliebtheit von Touchscreens bei den Bedienern. Die Einlernzeiten und das notwendige Training werden minimiert.

- Schnelligkeit

Mit dem Finger ist ein schnelleres Auslösen einer Funktion möglich als mit anderen Eingabetechniken.

- Keine zusätzliche Hardware

Dies ist – bei einer reinen Touchscreen-Realisierung – ein entscheidendes finanzielles Argument. Hardwarekomponenten (z. B. Tastaturen, Joystick usw.) müssen bei entsprechender Touchscreen-Funktionalität nicht, bzw. nur beschränkt, vorhanden sein. Damit entfallen auch aufwändige landesspezifische Beschriftungen der Hardware-Tasten.

Nachteile

- Verschmutzung

Durch die direkte Berührung mit dem Finger lagern sich Fett und andere Rückstände auf der Oberfläche ab. Verstärkt wird dies in industriellen Arbeitsumgebungen. Insbesondere durch Kühlsmiermittel, Ölnebel oder andere Industrieprodukte kann es zu einer Verschmutzung oder Beschädigung des Bildschirms kommen.

- Verschleiß

Verschleiß kann mechanisch durch Abnutzung (eine Stelle am Display wird immer wieder betätigt), Schlag-, Stoßbelastung und/oder chemisch durch die in Verschmutzungen enthaltenen Basen oder Säuren erfolgen.

- Optik

Um die Gefahren von Verschmutzung und Verschleiß zu mindern, können vandalismusichere Scheiben verwendet werden. Bei einer dickwandigen Glasfront tritt aber bei von 0 Grad abweichenden Blickwinkeln Parallaxe auf, welche sich störend auswirkt, wenn Positionen auf dem Display mit einem Gegenstand markiert werden und sich der erwünschte Auslöser in der Lage aber leicht versetzt befindet. Erfolgt die Touchscreen-Funktionalität über eine dem Display vorgelagerte

Folie, können sich die Transparenz, die Helligkeit und der Kontrast verschlechtern. Des Weiteren können bei ungünstigen Lichtverhältnissen Blendeffekte auftreten (z. B. bei Aufstellung im Freien).

- Anordnung

Die Anbringung des Touchscreens muss abhängig von den optischen Kriterien erfolgen und der Bildschirm muss im Greifraum des Nutzers angeordnet werden. Daraus ergeben sich evtl. Einschränkungen der Bewegungsfreiheit. Zur Bedienung kann u. U. häufig das Heben der Hand nötig sein, woraus sich eine schnellere Ermüdung ergibt.

- Alphanumerische Eingaben

Alphanumerische Eingaben müssen bei reiner Touchscreen-Funktionalität über den Bildschirm erfolgen. Dazu muss ein Tastatursatz auf dem Bildschirm geschaffen werden. Erfolgt dies über eine einblendbare Tastatur, muss dafür Sorge getragen werden, dass diese nicht andere Bereiche (z. B. Fehlermeldungen) überdeckt und dass potentiell erforderliche Eingaben möglich sind (Tastatur-, Tasten- und Fingergröße beachten).

- Kalibrierung

Aufgrund des Versatzes zwischen tatsächlichen Koordinaten und den vom Sensor detektierten Koordinaten infolge Änderungen der Bildgeometrie muss der Touchscreen von Zeit zu Zeit justiert werden. In Zusammenhang mit dem Parallaxefehler kann sich dieses Merkmal potenzieren und dazu führen, dass der Bediener neben den Positionspunkt drücken muss. Auch muss in diesem Zusammenhang eine Abstimmung von Sichtbereich und Greifbereich erfolgen.

- Bedienfunktionalität

Zur Steuerung eines auf Touchscreen-Technologie beruhenden Systems ist im Gegensatz zur Windows-Funktionalität eine völlig neue Oberfläche notwendig. Dies bedeutet einen Mehraufwand bei der Gestaltung des Eingabebereichs: Es steht nicht ständig die gesamte Funktionalität des Systems zur Verfügung (z. B. Drag & Drop und Slider-Bedienung), wodurch der Dialog durch Darstellung aller Funktionen hierarchischer und umfangreicher werden kann.

- Bedienbarkeit

Darunter lassen sich die physischen und physikalischen Schwierigkeiten zusammenfassen. Der Anwender erhält bei Betätigung einer Taste keine taktile Rückmeldung. Diese Rückmeldung muss anderweitig gewährleistet werden, z. B. durch ein akustisches oder visuelles Signal. Des Weiteren ist zu prüfen, ob beim Tragen von Handschuhen eine sichere Bedienung erfolgen kann.

- Verdeckung

Abhängig von der Größe des gewählten Displays und dem Bedienkonzept besteht die Gefahr, dass Informationen durch die Hand verdeckt werden können, wobei Rechtshänder andere Informationen verdecken als Linkshänder. Für die Selektion von kleinen Icons oder Feldern (z. B. Slider bei Bildlaufleisten) ist der Touchscreen ungeeignet.

Technologie Touchscreens ermöglichen ein Bedienen durch Zeigen auf die gewünschte Stelle in einem angezeigten Bild auf dem Bildschirm. Am Bildschirm angebrachte Sensoren geben Signale an eine Elektronik, die rechnergerechte Koor-

dinatenpaare ermittelt, d. h. Bilderzeugung und Koordinatenerfassung sind nicht miteinander gekoppelt.

Die verschiedenen Technologien und ihre Ausprägungen werden in diesem Kapitel kurz dargestellt.

- Resistiv:
Touchscreens nach dem resistiven Prinzip bestehen aus zwei übereinanderliegenden leitfähigen Schichten, die durch eine isolierende Schicht in der Mitte getrennt sind. Sobald auf die obere Schicht mittels Finger oder anderem Zeigegerät ein Druck ausgeübt wird, wird der Kontakt zwischen den leitenden Schichten hergestellt, wobei sich der elektrische Widerstand der Schicht im Verhältnis der Entfernung des Druckpunktes zu den gegenüberliegenden Seiten aufteilt. Dieses Widerstandsverhältnis (und damit die Lage des Druckpunktes) kann über einen Spannungsteiler ermittelt werden.
- Kapazitiv:
Kapazitive Touchscreens bestehen aus einer Glasplatte, die mit einer leitenden Beschichtung versehen ist und an deren vier Ecken eine Spannung angelegt ist. Durch eine Berührung mit einem Finger oder einem Zeigegerät aus leitendem Material wird der Stromkreis geschlossen und ein Strom wird über den Körper abgeleitet. Die Finger-Berührung teilt die Spannung zu jeder Seite proportional zur Entfernung von der Ecke. Ein Controller berechnet aus dem Spannungsverhältnis die Position des Fingers.
- Akustisch (SAW – Surface Acoustic Wave):
Eine Glasfläche über dem Display wird mit einem in eine Schallwelle umgewandelten elektrischen Signal beaufschlagt und dadurch in Schwingung versetzt. Die Welle breitet sich in der Glasfläche aus und wird wieder durch einen Empfänger auf der gegenüberliegenden Seite registriert. Durch einen aufgesetzten Gegenstand (Finger) wird die Welle ganz oder teilweise absorbiert und beim Empfänger trifft ein deutlich gedämpftes Signal ein. Über die Dämpfung lassen sich dann Lage des Berührpunktes und Anpresskraft ermitteln.
- Optisch (Infrarot):
An zwei Rändern des Displays werden Infrarot LEDs vertikal und horizontal angeordnet. Auf der gegenüberliegenden Seite jeder LED wird ein Lichtsensor angebracht, so dass eine Berührung des Displays zur Unterbrechung des Lichtstrahls zwischen LED und Lichtsensor führt (Lichtschrankenprinzip). Der Druckpunkt innerhalb der Infrarotmatrix kann anhand der unterbrochenen horizontalen und vertikalen Strahlen ermittelt werden.
- Dispersive-Signal-Technology:
Diese Ausprägung eines Touchscreens wurde von der Firma 3 M entwickelt und beruht auf der Messung der Stärke der Biegewelle, die sich innerhalb einer Glasplatte ausbreitet, wenn die Oberfläche mit dem Finger oder einem Stift berührt wird. Piezoelektrische Sensoren nehmen diese Welle an den Ecken der Plattenrückseite auf und wandeln sie in ein elektrisches Signal. Da hier die Ausbreitung der Welle innerhalb der Platte zur Bestimmung der Position genutzt wird (statt

Tab. 6.11 Eigenschaften der verschiedenen Touchscreen Technologien. (Schenk und Rigoll 2010, Holzinger 2003)

	Optisch	Akustisch	Resistiv	Kapazitiv	Piezo	DST
Auflösung	–	+	+	0	0	+
Griffel	Beliebig	Weich, energie-absorbierend	Beliebig	Leitend	Beliebig	Beliebig
Lichtdurchlässigkeit	++	+	0	+	+	++
Kalibrierung	Nie	Einmalig	Einmalig	Wiederholt	Einmalig	?
Haltbarkeit	+	+	0	+	+	?
Vorteile	?	Unempfindlich gegen Kratzer	?	?	Robust	Unempfindlich gegen Kratzer und Verschmutzung, geeignet für große Formate gut abdichtbar
Nachteile	Verschmutzungsempfindlich	Schlecht abdichtbar	Kratzempfindlich	Kratzempfindlich	Schlecht abdichtbar	

auf der Oberfläche), ist diese Technologie unempfindlich gegen Schmutz und leichte Beschädigungen der Oberfläche.

- **Piezoelektrisch:**

Piezoelemente wandeln eine auf sie einwirkende mechanische Belastung in eine proportionale Spannung um. Sie werden bei piezoelektrischen Touchscreens genutzt, um den Touchdruck in Spannungssignale umzusetzen. Sie sind jeweils an den Ecken des Touchscreens angebracht und der Druck (bzw. die Spannung) teilt sich abhängig von der Lage des Berührpunktes auf die vier Piezoelemente auf. Aus den Verhältnissen an den Piezoelementen gemessenen Spannungen kann die Position der Berührung errechnet werden.

Zum Abschluss dieser übergreifenden Technologiebetrachtung werden in Tab. 6.11 die Funktionsprinzipien, ihre Ausprägungen sowie die Vor- und Nachteile der Technologien zusammenfassend gegenübergestellt.

Touchscreen-Einsatz Soll ein Bediensystem mit einem Touchscreen realisiert werden, sind einige Randbedingungen und Richtlinien zu beachten. Diese Besonderheiten werden im folgenden Abschnitt behandelt und sind z. T. in VDI/VDE 3850 Blatt 3 und DIN EN ISO 9241 enthalten. Auf Basis von Flachbildschirmen müssen die hier zusammengestellten Kriterien beachtet werden:

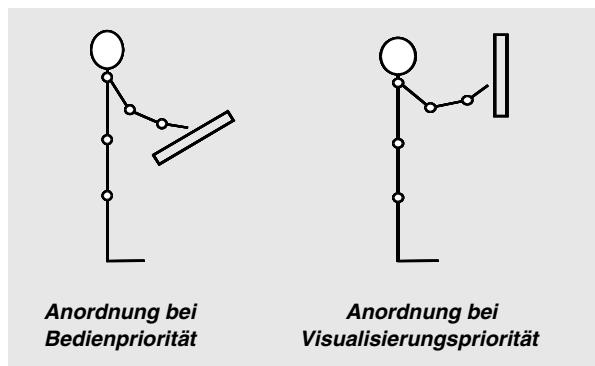
- **Einbau und Positionierung:**

Der Einbau des Displays sollte bündig mit der Gehäuseoberfläche erfolgen, um Verschmutzungen v. a. in den Ecken zu vermeiden und um eine leichte Reinigung zu gewährleisten. Ein nahtloser Einbau des Displays in eine Art Pult prädestiniert eine Abstützung des Bedieners mit den Armen auf diesem Pult. Fehlbedienung infolge solcher unabsichtlichen Berührung durch „Ablegen der Hände“ auf den Bildschirm (abhängig von der Einbaulage) können z. B. durch eine Schaltfunktion zur Deaktivierung der Touch-Funktionalität vermieden werden.

Aus der Praxis: Bei stark beanspruchten Touchscreen-Konzepten in der Industrie wird häufig vergessen, dass die Bildschirme zum Teil auch während des Betriebes gereinigt werden müssen. Wird dieser Punkt nicht bedacht, ist es nicht möglich, den Touchscreen zu reinigen ohne eine (oder mehrere) Funktionen durch die Berührung auszulösen. Daher ist eine Reinigungsfunktion vorzusehen, die die Touch-Sensitivität auch während des Betriebs für kurze Zeit deaktiviert, damit die Glasoberfläche gereinigt werden kann.

- Bei Druck auf die Displayoberfläche durch den Bediener ist sicherzustellen, dass das Display so stabil eingebaut ist, dass keine Schwingungen oder Verschiebungen auftreten können. Des Weiteren sollte das Display im Greifraum des Nutzers angeordnet sein und die Bedienung sollte ermüdfrei erfolgen können (z. B. durch Armauflage).

Abb. 6.17 Bedien- vs. Visualisierungspriorität



Einen wesentlichen Einfluss auf die Positionierung des Touchscreens hat die Entscheidung, ob eine Visualisierung- oder eine Bedienpriorität vorliegt (Abb. 6.17). Muss der Nutzer eher von beliebigen Stellen des Arbeitsplatzes aus einen Überblick über den Produktionsprozess erhalten, so ist das Gerät senkrecht in Augenhöhe anzubringen (Visualisierungspriorität). Muss er in erster Linie die Maschine bedienen, so ist der Pulteinbau in Ellbogenhöhe vorzuziehen, da dieser ein ermüdungsfreieres Arbeiten ermöglicht (Bedienpriorität)

Aus der Praxis: Der Einbau eines Touchscreens (wie auch eines anderen Bildschirms) in ein Pult führt häufig zu störenden Reflexionen der Hallenbeleuchtung auf der Glasoberfläche. Hier müssen entsprechende Maßnahmen getroffen werden, um diese negativen Effekte zu minimieren. Häufig helfen bereits geringe Änderungen des Einbauwinkels, um die Reflexionen zu beheben.

Der Touchscreen muss so angeordnet sein, dass der Bediener (zur Vermeidung von Fehlbedienungen infolge Parallaxe) möglichst im rechten Winkel auf den Bildschirm schaut. Zur Datenkontrolle (z. B. Arbeitsfortschritt) soll der Bildschirminhalt im Blickfeld des Nutzers gut einsehbar sein.

- **Hardwareauslegung:**
Die Hardware (Display) ist nach den unter Verschleiß und Optik genannten Kriterien auszulegen. D. h. es ist auf Transparenz, Helligkeit auch bei Tageslicht, ausreichenden Kontrast, Blendeffekte, Sichtbereiche und Greifbereiche zu achten.
- **Auflösung:**
Die Fehlerrate wird durch eine hohe Auflösung positiv beeinflusst: Je höher die Auflösung ist, umso genauer kann die Fingerposition ermittelt werden, da sie aus einer größeren Anzahl von Punkten errechnet wird. Je genauer das physikalische Touch-Feld mit einer grafischen Darstellung auf dem Display übereinstimmt, umso geringer wird die Fehlerrate.

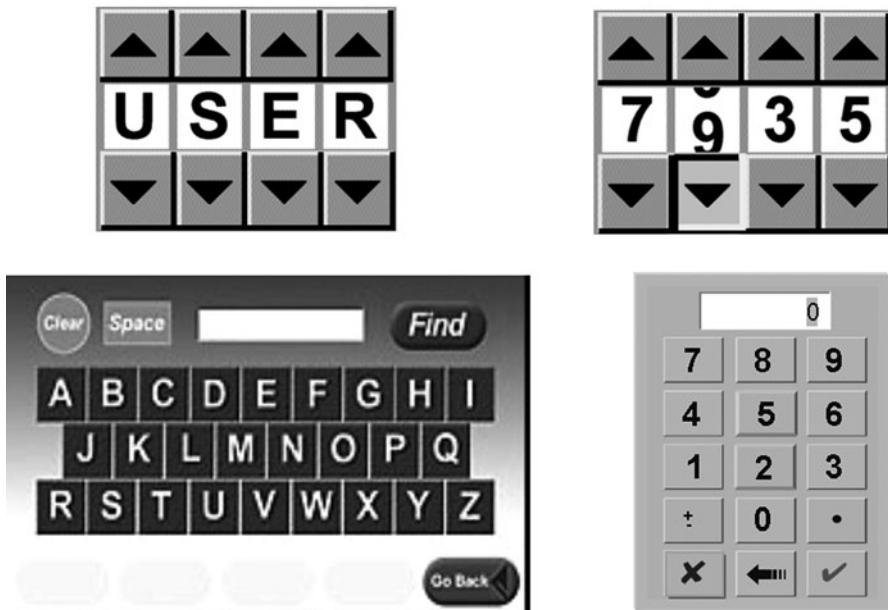


Abb. 6.18 Einfache und komplexe alphanumerische und numerische Eingabe

- Alphanumerische Eingaben:

Bei häufigen Dateneingaben ist ein externes Eingabeelement wie eine alphanumerische Tastatur oder ein Dreh-Druck-Steller notwendig. Soll die Dateneingabe auf dem Bildschirm erfolgen, müssen entsprechende On-Screen-Tastaturen automatisch bei Bedarf zur Verfügung gestellt werden. Bei geringen Dateneingabemengen mit eingeschränktem Wertebereich kann eine Auswahl der Zeichen (Zahlen, Buchstaben) über ein sog. Toggle-Wheel-Switch erfolgen (s. Abb. 6.18).

Spezielle Gestaltungsrichtlinien für Touchscreen Folgende Richtlinien erleichtern die Gestaltung eines ergonomischen Bediensystems auf Touchscreen-Basis.

Anordnung der Touch-Felder Der Nutzer nimmt zwischen den Bedienungsbewegungen eine „Grundhaltung“ (oder „Ruhestellung“) ein. Diese liegt bei Rechtshändern im rechten unteren Bildschirmquadranten und ca. 8 cm von der Oberfläche entfernt. Da Felder in diesem Bereich nur kleine Bewegungen erfordern, liefern sie die kürzesten Bedienzeiten. D. h. häufig zu bedienende Funktionen sind am besten in den rechten unteren Bereich zu legen.

Größe von Touch-Feldern Die Informationsfläche der jeweiligen Anzeige soll größer gleich der Fingerfläche des Bedieners sein (vgl. Abb. 6.19) wobei auch „Werkerfinger“ oder „Knöchelbedienungen“ und damit mögliche Verdeckungen berücksichtigt werden müssen. Ebenso ist das Tragen von Handschuhen zu beach-

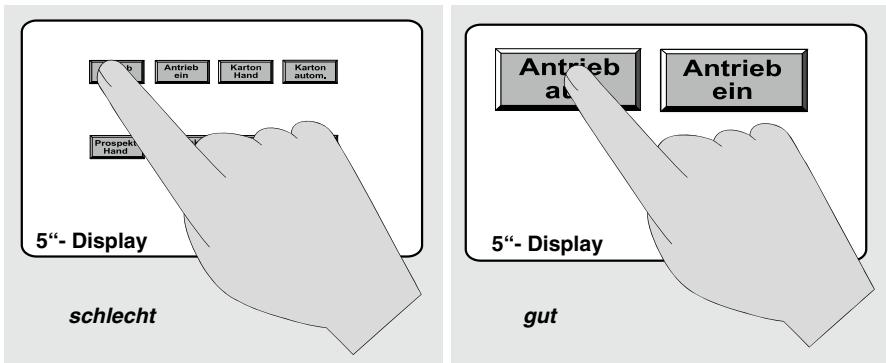


Abb. 6.19 Buttongröße bei Touchscreen-Verwendung

Tab. 6.12 Mindestmaße für Touch-Felder

	Größe [mm]
Touchfeldhöhe	10
Touchfeldbreite	20
Touchfeldabstand:	Horizontal 10
	Vertikal 5 oder 10

ten, wodurch die Fingerfläche zusätzlich vergrößert wird. Weiteres Kriterium ist die verwendete Technologie des Touchscreen. Insofern geben die in Tab. 6.12 dargestellten Werte nur Anhaltspunkte an. Es ist zu empfehlen, für andere Technologien im Zweifelsfall größere Felder zu verwenden.

Beschriftungen von Touch-Feldern Beschriftungen sollten in den oberen Randbereich der Felder gelegt werden. Dadurch können die Nutzer ein Feld berühren, ohne die Beschriftung zu verdecken (s. Tab. 6.13).

Globale Funktionen und maximal die Navigationsfunktionen der ersten Menüebene können durch Bildzeichen beschriftet werden, ohne den Nutzer zu überfordern. Weitere Menüebenen und die Interaktionselemente des Arbeitsbereichs sollten durch Symbole mit Text beschriftet werden. Das Symbol erleichtert dem Nutzer die Zuordnung der Bedienfunktion und durch die Textinformation kann diese konkretisiert werden. Kann aus Platz- oder Kapazitätsgründen kein Einsatz von Symbol und Beschriftung erfolgen, so ist die rein textuelle Beschriftung der rein symbolischen vorzuziehen.

Touchscreen-Dialog Aufgrund des unangenehmen Gefühls beim Verschieben eines Elements auf dem Display mit dem Finger können „Drag and Drop“-Funktionalitäten und Windows-Objekte wie z. B. Slider, Dreh- oder Schieberegler nur schwer realisiert werden. Ebenso ist ein Doppelklick nur sehr schwer zu realisieren, die rechte Maustaste existiert nicht und mehrere Tasten können nicht gleichzeitig gedrückt werden.

Tab. 6.13 Beschriftung von Touchfeldern

Art	Erläuterung	Beispiel
Nur Text	Damit die Beschriftung des Interaktionselements bei der Bedienung nicht verdeckt wird, muss diese in deren oberem Bereich angebracht werden.	
Nur Symbol	Bei Verwendung von Symbolen zur Beschriftung der Interaktionselemente kann dessen gesamte Fläche genutzt werden.	
Symbol und Text	Kommen Symbol und Text kombiniert zum Einsatz, sollten 2/3 der Fläche des Interaktionselements für das Symbol und 1/3 für den Text vorgesehen werden	

Rückmeldung Die Art und Geschwindigkeit der Rückmeldung hat beim Touchscreen beträchtlichen Einfluss auf das Nutzerverhalten und die Leistung. Dies resultiert v. a. aus der fehlenden taktilen Rückmeldung, die dem Nutzer im Allgemeinen am vertrautesten ist. Am besten geeignet erscheint eine Kombination aus auditiver (z. B. Click) und visueller (z. B. 3D-Effekte, ausgeprägte Schatten, deutliche Änderung der Kontraste in der Helligkeit (Leuchtdichte), Farbsättigung, Farbton der Buttons) Rückmeldung. Dabei teilt die auditive Rückmeldung dem Nutzer mit, dass eine Funktion vom System ausgeführt wird bzw. eine Eingabe erkannt wurde. Die visuelle Rückmeldung gibt am Touchscreen Auskunft, welche Funktion, bzw. welches Feld vom System erkannt worden ist.

Funktionsauslösung Für die Funktionsauslösung am Touchscreen existieren das Prinzip der Funktionsauslösung beim Berühren des Displays und das Prinzip der Funktionsauslösung beim Verlassen (Loslassen) des Touch-Feldes. Von der Art der Funktionsauslösung wird auch die Fehlerrate beeinflusst. Funktionsauslösung beim Verlassen des Touchscreens ermöglicht die Korrektur einer fehlerhaften Positionierung, bevor der Bediener den Finger vom Display abhebt, und vermindert damit eine fehlerhafte oder ungewünschte Funktionsauslösung.

Grafische Gestaltung Es ist empfehlenswert, helle Hintergrundfarben für einen Touchscreen zu verwenden, damit Reflexionen, die aufgrund von Fingerabdrücken auf dem Bildschirm noch verstärkt werden, nicht so sehr ins Gewicht fallen. Ebenso werden die Auswirkungen von Spiegelungen reduziert. Auch schwach gemusterte oder gerasterte Hintergründe lenken das Augenmerk von den Reflexionen zu der eigentlichen Oberfläche. Dies gilt insbesondere für Bereiche, in denen keine Icons oder Menüpunkte dargestellt sind.

6.3.3.4 Sprachein- und -ausgabe

Die Vision aus Science-Fiction Filmen, dass ein Nutzer mit einem Computersystem in Form eines Gesprächs interagiert und die zu erfüllenden Aufgaben mitteilt und ebenfalls in natürlicher Sprache Feedback vom System erhält, ist von der Realität noch weit entfernt.

Dies liegt vor allem in der Tatsache begründet, dass die Artikulation von Sprachkommandos eine vergleichsweise hohe kognitive Leistung erfordert, verglichen mit klassischen Click-and-Point-Bedienungen. Sprachverarbeitung findet im selben Teil des Gehirns statt, in dem auch Problemlösung und Planung von Handlungen stattfinden, während die Hand/Auge Koordination über andere Bereiche gesteuert wird, und daher ohne große Beeinträchtigungen parallel (nebenher) ausgeführt werden kann, wie Untersuchungen zeigen. Außerdem ist die Erkennung von Sprache in störbehafteten, lauten industriellen Umgebungen noch nicht problemlos möglich, sowie der Umgang mit wechselnden Nutzern mit hohem Einlernaufwand verbunden. Daher muss der Einsatz dieser Interaktionstechnologie sorgfältig abgewogen werden.

Nichtsdestotrotz kann ein Einsatz von Sprachebedienung sinnvoll sein, wenn:

- die Hände des Nutzers nicht frei sind
- Mobilität erforderlich ist
- die Augen nicht auf einen Bildschirm gerichtet sein können, weil die Arbeitsaufgabe die kontinuierliche Beobachtung anderer Elemente erfordert
- die Umgebungsbedingungen die Nutzung eines anderen Eingabeelementes ausschließt.

6.3.3.5 Displays

Bildschirme sind die Hauptquelle von Feedback des Systems zum Nutzer. Im Folgenden werden einige wichtige Bildschirm-Technologien kurz vorgestellt und auf die im industriellen Umfeld interessanten Head Mounted Displays eingegangen (Abb. 6.20).

- **Kathodenstrahlröhren-Bildschirme (CRT)**

Bei den sogenannten Röhrenmonitoren werden in der Bildröhre („Braunsche Röhre“) durch einen in horizontaler und vertikaler Richtung auslenkbaren Elektronenstrahl Bilder auf der Leuchtschicht sichtbar gemacht. Solche Bildschirme haben einen großen Betrachtungswinkel, ein hohes Auflösungsvermögen und gute Farbe- und Kontrasteigenschaften. Allerdings haben sie auch einen sehr großen Raum- und Energiebedarf, weshalb sie gegenüber anderen sparsameren Technologien deutlich an Popularität eingebüßt haben.

- **LC-Displays (LCD)**

LC (Liquid Crystal = Flüssigkristall) sind eine organische Substanz in molekularer Größe. Optisch zeigen sie kristalline Eigenschaften (Transparenz), solange die Moleküle gleich ausgerichtet sind. Beim Anlegen einer elektrischen Spannung ändert sich die Struktur und damit die Lichtdurchlässigkeit.



Abb. 6.20 Ausgabegeräte (v. l.: Bildschirm (CRT), LC-Display (LCD), Head-Mounted Display (HMD)). (Quellen: EIZO, MICROOPTICAL)

Im Gegensatz zu Röhrenbildschirmen haben LCDs nur einen geringen Raum- und Energiebedarf. Nachteilig ist allerdings der eingeschränkte Temperaturbereich von ca. 0 bis 50 °C (Spezialdisplays: -20 bis 70 °C).

- **Plasma-Displays**

Bei Plasmabildschirmen wird das Bild durch Gasentladungen zwischen zwei Glasplatten des Displays erzeugt. Einzelne kleine Kammern sind gefüllt mit Edelgasen, die bei Anlegen einer elektrischen Spannung Licht emittieren, je nach genauer Zusammensetzung des Gases in Grün, Blau oder Rot. Drei Kammern ergeben dabei jeweils zusammen einen Bildpunkt. Plasmabildschirme haben wie LCDs einen geringeren Raumbedarf, allerdings benötigen sie mehr Energie.

- **Head-Mounted Displays (HMD)**

Bei HMD's handelt es sich um Bildschirmsysteme, die auf oder am Kopf getragen werden und die Bilder direkt vor die Augen projizieren. Die Bilder werden entweder so dargestellt, dass der Nutzer von ihnen vollständig umgeben ist und dadurch komplett in eine virtuelle Realität (Virtual Reality) eintauchen kann oder es findet eine Überlagerung von realem und virtuellem Bild (sog. Erweiterte Realität, Augmented Reality) statt. Die Head-Mounted Displays sind meist als Brille oder als auf eine Brille aufsteckbares Modul ausgeführt oder in eine Art Helm integriert.

Aus der Praxis: Bei der Verwendung von Monitoren ist ganz allgemein auf die Umgebungsbedingungen zu achten. Hier können vor allem die Lichtverhältnisse einen ständig wechselnden und zum Teil auch störenden Einfluss haben.

Die Leuchtdichte (vom Auge wahrgenommene Helligkeit) sollte einen Wert von etwa 35 cd/m² besitzen. Außerdem sollten Spiegelungen und Reflexionen des Umgebungslichts vermieden werden. Dies kann zum einen durch die richtige Aufstellung (parallel zu Fenstern oder anderen Lichtquellen) und zum anderen durch Hilfsmittel wie entspiegelte Oberflächen oder Anti-Reflex-Beschichtungen erreicht werden.

6.3.3.6 Evaluation von Interaktionsgeräten

Alle vorgestellten Interaktionsgeräte haben ihre speziellen Vor- und Nachteile insbesondere dann, wenn sie im industriellen Umfeld eingesetzt werden sollen. Die Frage der Entwickler, welche Geräte am besten einzusetzen sind, kann nicht generell beantwortet werden. Umgebungsbedingungen, Bedienhaltung und Arbeitsaufgabe haben hier einen starken Einfluss.

Um die Gebrauchstauglichkeit verschiedener Eingabegeräte vergleichen zu können, müssen diese mit standardisierten Untersuchungsmethoden getestet werden. Hierzu gibt es verschiedene Modelle und Testverfahren. Zwei der wichtigsten Verfahren werden im Folgenden erläutert. Zum einen handelt es sich hierbei um den international verbreiteten aber aus dem Jahr 1954 stammenden Teststandard FITTS LAW (Fitts 1954), zum anderen um die wesentlich modernere Untersuchungsmethode DEVICE (Krauß 2003).

FITTS LAW Fitts leitete aus der informationstechnischen Betrachtung des Shannon-Theorems ab, dass zur Ausführung einer Zielbewegung eine bestimmte Informationsmenge notwendig ist, die im Wesentlichen vom Logarithmus dualis des Verhältnisses zwischen Bewegungsentfernung und Zielgröße abhängt.

Bildlich gesprochen ist die Bewegungszeit abhängig zwischen dem Abstand zweier Ziele und der Zielbreite. In den damaligen Versuchen mussten die Probanden mit einem Metallstift abwechselnd zwei Platten berühren. Dabei wurden die Zeiten zwischen den Berührungen gemessen. Dieser Test wurde mit Platten verschiedener Breite und Höhe durchgeführt.

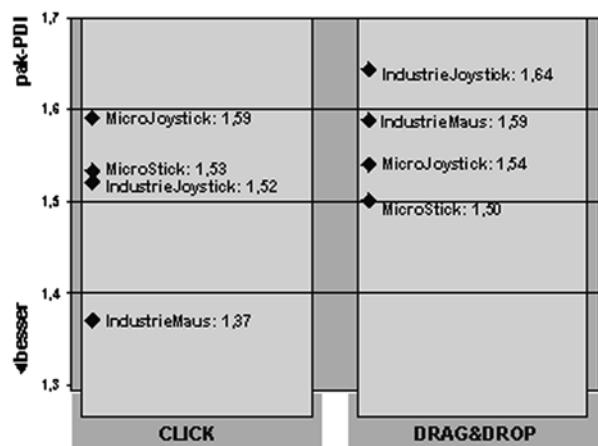
Umgesetzt auf Interaktionsgeräte muss das Fitts-Experiment auf eine Bedienoberfläche abgebildet werden, auf der Probanden mit verschiedenen koordinatengebenden Interaktionsgeräten zwischen Feldern verschiedener Größe „hin- und her springen“. Durch Mausklicks wird dabei jeweils die Zeitmessung ausgelöst und gestoppt. Bewertungskriterium für die Geräte ist der so genannte Leistungsindex, der aus der Bewegungszeit und dem Schwierigkeitsgrad (abhängig von Größe und Abstand der Felder) über alle Versuchspersonen hinweg gemittelt wird.

Die Testmethode nach Fitts wurde nicht für die heutige Problematik der Bildschirminteraktion entwickelt. Ihre Anwendung hierfür ist zwar prinzipiell möglich, liefert aber wenig aussagekräftige Ergebnisse.

DEVICE Die Ende der 90er-Jahre entwickelte Testmethode DEVICE orientiert sich im Gegensatz zu FITTS LAW speziell an der Interaktionsproblematik mit koordinatengebenden Geräten und Bildschirmen. Sie ist eine relative Methode, bei der verschiedene Versuchspersonen standardisierte aufgabentypische Testaufgaben (z. B. Eintrag aus Liste wählen, verschiedene Fenster anwählen, Markieren) mit verschiedenen Interaktionsgeräten bearbeiten müssen. Dabei werden die Bearbeitungszeit und die Fehlerrate zur Auswertung herangezogen. Alle Werte beziehen sich relativ auf die Versuchsergebnisse einer normalen Büro-Maus (=100 %), da sich diese bis jetzt als das weit verbreitetste und gebräuchlichste Interaktionsgerät bei den koordinatengebenden Interaktionsgeräten etabliert hat.

DEVICE hat zum Ziel, die folgenden vier Fragestellungen zu beantworten:

Abb. 6.21 Bewertung von Interaktionsgeräten mit dem pak-pdi



- Wie gut kommen die Nutzer mit einem Interaktionsgerät zurecht?
- Mit welcher Geschwindigkeit können Nutzer ein Interaktionsgerät bedienen?
- Wie sicher erreichen Nutzer eine bestimmte Zielposition?
- Wie genau können Nutzer das Interaktionsgerät bedienen?

Gemessen werden jeweils die Zeit, die benötigte Versuchsanzahl und die Positioniergenauigkeit. Diese Werte ergeben den **pak-PDI** (Pointing Device Index) und sind ein Kennwert für die Qualität und die Tauglichkeit eines Interaktionsgeräts (s. Abb. 6.21). Dabei werden noch der pak-PDI_C für Click-Aufgaben und der pak-PDI_D für Drag&Drop-Aufgaben unterschieden. Als Faustformel ergibt sich:

Je kleiner der Kennwert (pak-PDI) ist, desto besser ist das Interaktionsgerät für den spezifischen Untersuchungskontext (Nutzergruppe, Bedienhaltung, Gerätelocation, etc.) geeignet, um die jeweilige Aufgabe zu erfüllen, bzw. desto eher kann es als Alternative zur Standardmaus in Betracht gezogen werden.

Kapitel 7

Zukünftige Interaktionstechniken

7.1 Multimodale Interaktion

In den bislang betrachteten Usewaresystemen gab es weitgehend nur zwei Interaktionsformen: über den visuellen Kanal übermittelte die Maschine dem Menschen die Informationen über ihren Status und über den haptischen Kanal (Tastatur und Maus, aber auch Touchscreen) nahm sie die Eingabedaten des Nutzers entgegen. Der auditive Kanal wurde allenfalls minimal verwendet, um den Menschen über Pieptöne oder ähnliche Geräusche Feedback über Fehler zu vermitteln. Diese Technik bezeichnet man als unimodale Interaktion (s. Abb. 7.1).

In Kap. 2 wurde aber bereits deutlich gemacht, dass der Mensch besonders gut und vor allem dann fehlerfrei kommuniziert, wenn möglichst viele Sinneskanäle beim Informationsaustausch beteiligt sind und Redundanzen schaffen, die vor allem der Fehlererkennung dienen. Eine solche Kommunikation basiert auf der sog. multimodalen Interaktion, bei der mehrere Sinne bidirektional genutzt werden (s. Abb. 7.2).

Über den visuellen Kanal sieht der Mensch den Bildschirm und die Maschine sieht den Nutzer und wertet z. B. seine Gestik und seinen Blickfokus aus. Über den auditiven Kanal hört der Mensch die Maschine und die Maschine kann den Menschen hören. Hier wird vor allem die natürlich-sprachliche Interaktion stattfinden, in der Mensch und Maschine quasi natürlich miteinander kommunizieren. Über den haptischen Kanal erfährt die Maschine den Menschen z. B. über Tasten, Maus, Joystick o. ä. Umgekehrt versorgt sie den Menschen über Kraft- bzw. Bewegungs-rückkopplung mit den entsprechenden haptischen Feedbackinformationen. Weitere Sinnesmodalitäten wären in ferner Zukunft sicher auch denkbar.

Auch wenn dieses Szenario futuristisch klingen mag, so belegen viele technische Entwicklungen den Weg zur multimodalen Interaktion. Wenn die Interaktion durch die Multimodalität menschähnlicher und damit hoffentlich intuitiver wird, wird der Mensch auch immer weniger den Computer als das zentrale Element einer Maschinen- bzw. Anlagensteuerung wahrnehmen. Der Computer wird unsichtbar und mehr und mehr zum stillen Diener im Hintergrund.

Die Vorreiterrolle in der Nutzung dieser Technologien übernimmt sicher nicht der industrielle Bereich sondern der Konsumbereich. Vielerorts sind Smart Homes

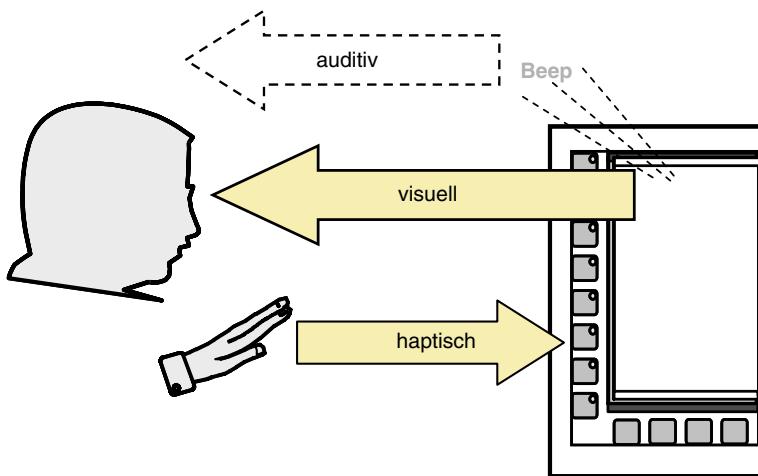


Abb. 7.1 Unimodale Interaktion

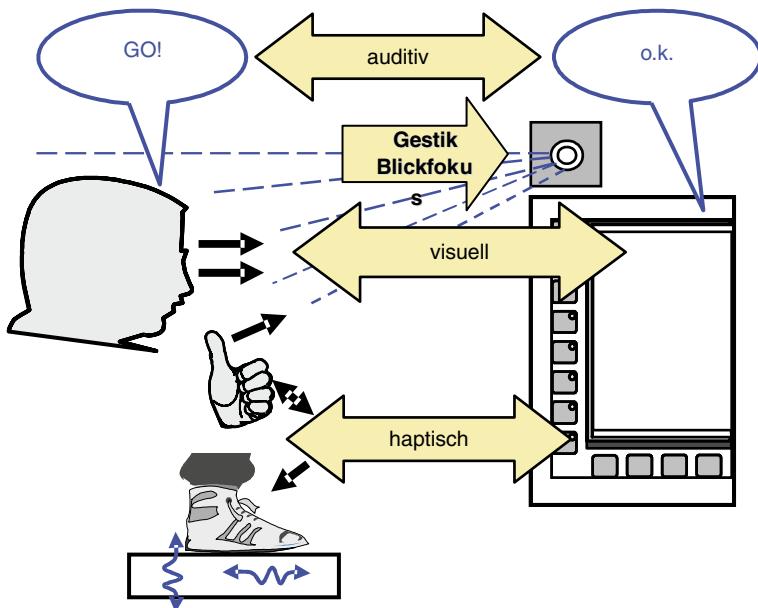


Abb. 7.2 Multimodale Interaktion

entstanden, in denen der neue Umgang zwischen Mensch und Technik erprobt und dokumentiert werden soll. Alle Geräte in diesen Häusern sind miteinander vernetzt, die beteiligten Computer sind größtenteils unsichtbar in die Geräte des alltäglichen Gebrauchs so integriert, dass der Mensch sie nicht mehr wahrnimmt, und die Interaktion zwischen Mensch und Computer wird durch die Nutzung multimodaler Techniken menschähnlich. Wie häufig am Beginn eines neuen Technikzeitalters werden viele Begriffe erfunden, die die neuen Technologien umschreiben sollen. So spricht man im amerikanischen beispielsweise von Ubiquitous Computing (allgegenwärtig), von Pervasive Computing (alles durchdringend) oder von Invisible Computing (unsichtbar). Im europäischen Raum scheint sich der Begriff Ambient Intelligence (umgebende Intelligenz) zu etablieren, der ursprünglich von der Fa. Philips geprägt wurde, seit einiger Zeit aber vor allem durch entsprechende Programme der EU-Forschungsförderung eine übergeordnete Bedeutung erlangt hat.

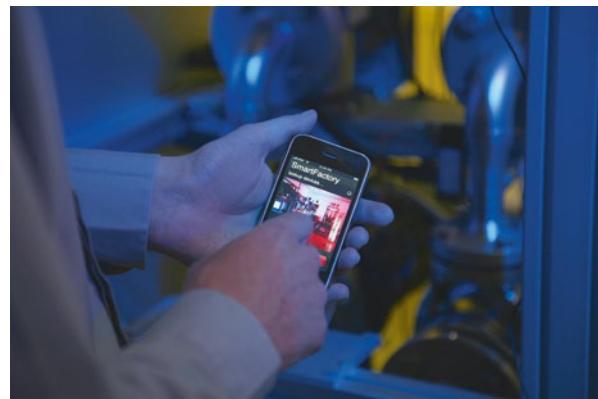
Kennzeichnend für dieses neue Technikzeitalter sind eine Reihe von Basisentwicklungen, die, wie schon angeführt, Bausteine zu einer neuen Welt der Interaktion zwischen Mensch und Technik sind.

7.2 Mobilität durch Miniaturisierung

Kleinste Computer – sog. embedded Systems – machen z. Z. enorme Fortschritte in Bezug auf ihre Leistungsfähigkeit und ihren Stromverbrauch. Damit sind sie die Hardwarebasis für eine neue Generation von mobilen Geräten, die extrem miniaturisiert sind, so wenig Leistung brauchen, dass sie über Tage oder Wochen aus einer Batterie arbeiten und obendrein so unempfindlich sind, dass sie in Kleidungsstücke (Wearables) integriert werden können. Anders als heutige PC's werden sie spezialisierte Aufgaben wie etwa die Navigation des Menschen in einer fremden Umgebung oder die Informationsbereitstellung in speziellen Arbeitsaufgaben übernehmen. Damit sie dieses erfüllen können, müssen sie mit anderen spezialisierten Computern und dem Nutzer kommunizieren können. Einen ersten Schritt in diese Richtung gehen heutige Smartphones (Abb. 7.3). Durch die Verwendung von modernen Funktechnologien und hochauflösenden Touchscreens, erlauben Smartphones den mobilen Zugriff auf nahezu beliebige Anwendungsdaten (z. B. Email-, Webbrowsere, Terminkalender, ...) und Multimedia-Inhalte (MP3, Video, ...). Im Umfeld der Fabrik bedeutet dies, dass Smartphones in Zukunft als Basis für mobile Bediensysteme bei der Wartung und Diagnose eingesetzt werden können. Mittels der Integration von modernen Sensorsystemen (Kamera, Beschleunigungssensoren, ...) ermöglichen diese Geräte zusätzlich die Unterstützung des Nutzers mittels neuer Interaktionsformen. So kann die integrierte Kamera des Smartphones genutzt werden um Umgebungsgestände selbstständig zu erkennen und dem Nutzer anschließend zu erklären.

Somit wird der zukünftige Anlagenbediener sehr viel freier agieren können als dies heute der Fall ist. Er wird sich mit Hilfe mobiler Geräte von seinem Arbeitsplatz entfernen können und dabei trotzdem ausreichend Möglichkeiten zur Inter-

Abb. 7.3 Mobile Interaktion in der Fabrik mittels eines Smartphones



aktion mit seinem Prozess bzw. Anlage haben. Dieses ist die Basis für neue Formen der Arbeitsgestaltung, die zu mehr Verantwortung und Zufriedenheit führen kann.

7.3 Neue Interaktionstechniken

Kleinste Computer für mobile Anwendungen können weder über eine ausladende Tastatur noch über die heute üblichen Touch-Bildschirme mit Schrifterkennung bedient werden, da die Hand im Prozess benötigt wird und ein Mensch in Bewegung keine sensomotorisch anspruchsvollen Aufgaben mit entsprechender Präzision erledigen kann. Neue Interaktionsformen wie Sprach- oder Gesteninteraktion sind gefordert. Dabei darf sich die Kommunikation nicht mehr am Computer orientieren, wie es z. B. bei den heute verwendeten Kommandosprachen der Fall ist, sondern sie muss sich am Menschen orientieren. Der Mensch kommuniziert aber in seltenen Fällen ausschließlich über die Sprache, stattdessen nutzt er – häufig unbewusst – parallel mehrere Sinneskanäle, die redundante oder für das Verständnis ergänzende Informationen liefern. „Wechselseitig mal dieses Werkzeug aus“ kann erst verstanden werden, wenn man auch eine entsprechende Zeigehandlung und den Blickfokus der sprechenden Person auswertet.

In der Erkennung wie auch Generierung natürlichsprachiger Dialoge ist die Technik mittlerweile weit fortgeschritten, so dass man sie bereits heute zumindest in abgegrenzten Anwendungsbereichen ausreichend sicher einsetzen kann. Vor allem in telefonischen Auskunftssystemen oder im Bereich der Call-Center-Automatisierung werden mittlerweile Sprachdialogsysteme eingesetzt, die natürliche Sprache trotz all ihrer Unwägbarkeiten mit hoher Sicherheit erkennen können (Gorin 2003). Der Vergleich mit sprachgesteuerten Diktiersystemen für die Textverarbeitung macht hierbei wenig Sinn. Diese Systeme sind darauf konzipiert, alle gesprochenen Worte mit hoher Präzision aber ohne ein explizites Sinnverständnis zu erkennen. Im Gegensatz dazu sollen Sprachinteraktionssysteme vor allem den Sinn des Gesprochenen erkennen und hierzu müssen sie – genauso wie der Mensch –

bei weitem nicht alle Wörter erkennen. Sie setzen dabei genau wie der Mensch auf Kontextwissen und nutzen Wahrscheinlichkeiten bestimmter Worte und deren Kombinationen in bestimmten Anwendungsfällen. Außerdem nutzen solche Systeme bei Nichterkennung die gleichen Strategien wie der Mensch: sie fragen erneut nach und steigern so die Erkennungsrate drastisch.

Bei der visuellen Erkennung menschlicher Interaktionsmerkmale wie etwa die Gestik- und Mimikerkennung steht die Anwendbarkeit noch ganz am Anfang. Gerade hier behindern die Umgebungsbedingungen des Arbeitsplatzes sowie die bislang verfügbaren Sensoren eine sichere und vor allem praktikable Erkennung. Die zumeist auf Videosensorik basierenden Systeme sind nicht nur hinderlich im Gebrauch sondern auch zu empfindlich gegenüber Beleuchtungsveränderungen. Eine Ausnahme bilden hier Erfassungssysteme für den Blickfokus allerdings auch nur dann, wenn sich die Blickfokuserfassung z. B. auf den Bildschirmbereich beschränkt. Allgemein einsetzbare Blickverfolgungssysteme (Gaze Tracker) sind zwar erhältlich, jedoch so unhandlich und empfindlich, dass ihre Anwendung bislang auf Laborumgebungen beschränkt bleiben sollte. Im Gegensatz dazu gab es bei der Gestikerkennung beachtliche Fortschritte. Spielekonsole wie die Nintendo Wii und Smartphones wie das iPhone haben gezeigt, dass sich die menschlichen Bewegungen von Armen und Händen zur Interaktion sinnvoll einsetzen lassen.

7.4 Displaytechnologien

7.4.1 *Flexible Displays*

Da der visuelle Kanal für den Menschen nach wie vor der wichtigste ist, werden visuelle Darstellungen eine hohe Bedeutung behalten. Im Bereich der Displaytechnologie werden große Fortschritte zu erwarten sein. Flexible Displays auf organischer Basis werden gerade im Bereich mobiler Systeme neue platzsparende Realisierungsformen gestatten. (s. Abb. 7.4)

Während man sich in heutigen technischen Systemen vorwiegend auf 2D-Darstellungen beschränkt und somit den menschlichen Sinn der Tiefensensibilität (s. Kap. 2) ignoriert, geht der Trend in der Konsumwelt eindeutig in Richtung 3D-Darstellung. Doch sind die für die 3D-Wahrnehmung erforderlichen Hilfsmittel wie Shutterbrillen o. ä. im industriellen Umfeld ungeeignet.

7.4.2 *3D Monitore (Autostereoskopische Displays)*

1995 begann die prototypische Entwicklung von Monitoren, die ohne externe Hilfsmittel (wie z. B. 3D-Brillen) dreidimensionale Bilder erzeugen konnten. 1998 wurde auf der CeBit der erste serienreife Prototyp eines solchen Monitors vorgestellt. (s. Abb. 7.5).

Abb. 7.4 Siemens ePYRUS.
(Quelle: Siemens)



Abb. 7.5 3D-Bildschirm



Autostereoskopische Monitore können ohne zusätzliche Hilfsmittel 3D-Daten darstellen (daher autostereoskopisch). Dabei werden dem Betrachter ständig zwei Perspektiven präsentiert, die den 3D-Eindruck erzeugen. Ein Problem stellt die Nachführung dieser Perspektiven bei einer Bewegung des Betrachters dar. Hier können zum einen verschiedenen Perspektiven ausgegeben werden, wodurch mehrere Personen auf einem Monitor 3D-Bilder wahrnehmen können. Zum anderen können durch den Einsatz einer Blickverfolgungskamera ständig die Positionen der Augen des Betrachters ermittelt werden. Diese Daten bewegen eine Prismenscheibe

Abb. 7.6 Die virtuelle Fabrik als 3D-Bediensystem



vor dem Monitor, die den 3D-Eindruck hervorruft. Dadurch ist gewährleistet, dass trotz einer Kopfbewegung der 3D Eindruck erhalten bleibt, da das Bild ständig nachgeführt wird. Der Anwender hat den Eindruck als könne er die sichtbaren Objekte räumlich von allen Seiten betrachten und sogar anfassen. Mittlerweile sind lauffähige DirectX-Treiber auf dem Markt, die es erlauben, nahezu alle Anwendungen in 3D zu betrachten. Es wird damit gerechnet, dass bald die ersten 3D Bildschirme für Laptops und Handhelds zur Verfügung stehen werden. 3D-Monitore können mittlerweile bis zu 42 Zoll groß sein und eignen sich damit auch für Präsentationszwecke.

Im Allgemeinen unterstützen 3D-Darstellungen den Menschen in seiner Kognition ohne ihn zusätzlich zu belasten. Für den industriellen Bereich sind hier vor allem Prozess- oder Bauteilansichten für Techniker und Maschinenbediener die Hauptanwendungsbereiche. Es muss allerdings auch sehr deutlich gesagt werden, dass der Mensch Tiefeninformationen nicht nur über seine beiden Augen empfindet. Eine Darstellung auf einem 3D-Monitor alleine bringt somit nichts, wenn nicht auch die anderen Randbedingungen in der Darstellung eingehalten werden (s. Kap. 2.2.1).

Durch Einsatz solcher Bildschirme ließen sich Anlagen- oder Bauteildarstellungen dem Nutzer sehr realitätsnah präsentieren. Darüber hinaus bietet sich damit aber auch die Möglichkeit, die Bediensysteme selbst in drei Dimensionen zu gestalten. Damit könnte man den Übergang von der bislang dominierenden 2D-Schreibtisch (Desktop)-Metapher zur 3D-Arbeitsraum (Workspace)-Metapher vollziehen (Oortmann 2007) (Abb. 7.6).

Dabei würde man den Aufgabenbereichen des Nutzers jeweils Räume zuweisen, in denen dann z. B. die heutigen „Windows“ an den Wänden hängen würden. So könnten in einem Raum z. B. NC-Programme archiviert sein, in einem anderen wären Betriebsdaten verfügbar und im Kontrollraum würde sich die Anlagenüberwachung und -steuerung abspielen. Damit ein solches 3D-Szenario sinnvoll nutzbar wäre, müssten allerdings geeignete Interaktionstechniken zur Navigation in solchen

Räumen verfügbar sein. Mit einer 2D-Maus wird dieses sicher nicht gelingen. Die heute schon verfügbaren 6D-Mäuse (SpaceMouse), die 3 Translationen und 3 Rotationen an den Rechner ausgeben, wären zwar prinzipiell geeignet, doch bestätigen Praxisversuche, dass die menschliche Feinmotorik mit der simultanen Steuerung von sechs Freiheitsgraden i. A. überfordert ist.

Auch die autostereoskopischen Displays werden z. Z. miniaturisiert. So wurde beispielsweise mit dem Nintendo 3DS eine mobile Spielekonsole auf den Markt gebracht, die über ein autostereoskopisches Display verfügt. Diese Entwicklung zeigt, dass auch bei mobilen Bediensystemen der Einsatz von 3D-Effekten zukünftig eine größere Bedeutung spielen wird.

7.4.3 *Virtuelle Realität (VR)*

Virtuelle Realitäten sind heute schon längst zu einer ernst zu nehmenden Technologie herangereift. Entwickelt wurde die Virtual Reality bereits in den 40er-Jahren aus mechanischen Simulatoren, die der Ausbildung von Fliegern diente. Unter virtueller Realität wird stark vereinfacht die Schaffung von künstlichen Welten verstanden, die von leistungsstarken Computern erzeugt werden. In ihnen ist mit entsprechender Hardware die Bewegung, die Interaktion mit Elementen und letztlich auch die Gestaltung möglich. Der Nutzer kann innerhalb dieses Raumes in alle Richtungen blicken und sich sogar bewegen. Doch was die VR-Technologie ausmacht, ist nicht allein die visuell perfekte Illusion einer vom Computer generierten künstlichen Welt in 3D – vielmehr ist es die interaktive Einbeziehung des Anwenders, die sie auszeichnet. Virtuelle Realität ist also die Übersetzung der Wirklichkeit in einen Datenraum, in dem sich der Anwender bewegen und den er in Echtzeit umgestalten kann. Neben den Anwendungen in der Fahrzeugentwicklung werden im Zuge des Digital Manufacturing ganze Fabriken mit all ihren komplexen Strukturen und Abläufen am Computer geplant. Bereits vor der Grundsteinlegung können Spezialisten virtuelle Rundgänge durch Montagehallen machen, die erst Jahre später Wirklichkeit werden. Es wird so sichergestellt, dass alle Anlagen optimal zusammenarbeiten und der Produktionsfluss störungsfrei und effizient ablaufen kann. Simulationsmethoden leisten darüber hinaus einen wertvollen Beitrag zur Gestaltung ergonomischer Arbeitsplätze, zu optimalen Logistikstrukturen und Maschinenanordnungen.

In der „realen“ Welt sind die Menschen umgeben von mehr oder weniger festen Gegenständen, die in der virtuellen Welt nicht existieren. Einen großen Teil der Orientierung erhält der Nutzer durch die Wahrnehmung über seine Sinnesorgane. Dem Menschen stehen Technologien wie der Datenhandschuh oder der Datenanzug zur Verfügung, um den virtuellen Objekten eine Materialität zu verleihen.

Der Datenhandschuh ermöglicht dem Anwender in der virtuellen Welt Gegenstände anzufassen, zu bewegen und dem Computer Anweisungen zu geben. Um die Hand zum Mund zu führen, braucht die Person sie in der realen Welt nicht zu sehen. Für viele Aktionen ist aber eine Auge-Hand Koordination sehr wichtig, so auch in der virtuellen Welt. Datenhandschuhe geben die Möglichkeit, die Position und Orientierung einer Hand mittels Trackingsystem zu messen und diese im virtuellen

Abb. 7.7 Augmented-Reality Anwendung. (Quelle: DFKI/AV)



Raum zu visualisieren. Doch die Visualisierung der Position reicht nicht aus. Für Kommunikationsprozesse sind beispielsweise Gesten, also Handbewegungen sehr wichtig. Handbewegungen können ebenfalls mit dem Datenhandschuh ermittelt und visualisiert werden. Mit dieser Technologie sind letztlich komplexere Interaktionen mit Objekten im Cyberspace, wie das Greifen eines Gegenstandes möglich.

Mit Hilfe des Datenanzuges wird der Körper des Nutzers in die virtuelle Welt eingebunden. Der Mensch kann sich durch diesen mit Sensoren ausgestatteten Anzug in der künstlichen Umgebung bewegen. Der Vorteil besteht hier in der Wahrnehmung des eigenen Körpers in der künstlichen Realität. Bewegungen wie das Gehen, Springen und Knien können vom Computer registriert und in die virtuelle Welt übertragen werden.

Wie die geschilderten Beispiele zeigen, erfordert die VR-Technik eine Gerätetechnik, die im industriellen Werkstattumfeld nicht eingesetzt werden kann. Daher wird sich ihr Einsatz auf die der Produktion vorgelagerten Bereiche der Planung beschränken.

7.4.4 *Augmented Reality (AR)*

AR, die „erweiterte Realität“ ist im weiteren Sinne eine Einblendung von Zusatzinformationen zu einem real beobachteten Objekt auf einem Computerbildschirm. Im Gegensatz zur virtuellen Realität wird die Welt nicht ersetzt sondern ergänzt. Idealerweise wird eine Koexistenz zwischen natürlicher und augmentierter Realität erreicht. Die eingeblendeten Informationen können kontextabhängig sein, d. h., das reale Objekt wird mit einem vom computergenerierten, meist räumlichen virtuellen Objekt visuell überlagert (Abb. 7.7). Augmented Reality hilft dem Nutzer die Umgebung zu verstehen, indem sie diese Umgebung dort erweitert, wo sie für den Nutzer interessant ist.

Ein wichtiger Bestandteil für das AR ist das Tracking-System. Hierbei handelt es sich um eine Objektverfolgung mit Hilfe einer an den Computer angeschlossenen Kamera oder anderer Geräte. AR benötigt das Tracking um reale Objekte digitalisieren zu können und ihnen virtuelle Objekte zuzuordnen. Ein weiterer wichtiger

Bestandteil sind die Ausgabesysteme. Hierzu werden so genannten See-Through-Head-Mounted Displays verwendet. Im Gegensatz zu anderen Head-Mounted-Displays wird hier nicht direkt auf einen Bildschirm geschaut, sondern auf einen teil-durchlässigen Spiegel, durch den sowohl die reale Welt als auch das elektronische Bild gesehen werden kann, welches vom Display über den Kombinator ins Auge geleitet wird.

7.5 Netzwerktechnologien

Die zukünftige Rechnerwelt wird sehr viel stärker auf kleineren dedizierten Rechnern sowie auf deren Vernetzung basieren. Zukünftige Netzwerke werden einen vorwiegend drahtlos arbeitenden allgegenwärtigen (ubiquitous) und spontanen Zugang bieten und sich dabei selbst organisieren. Wenn wir zukünftig durch unsere Umwelt gehen, so werden sich unsere technischen Systeme ständig und von uns unbemerkt in diese Netze einbinden und den Datenaustausch untereinander wie auch mit weltweiten Datennetzen ermöglichen. Wenn ein Mitarbeiter beispielsweise seine Firma betritt, wird die Anwesenheit automatisch erfasst, die Zugangsbe-rechtigung überprüft werden und am Arbeitsplatz wird er automatisch den Zugriff auf seine Maschinensteuerung, den Produktionsplan o. ä. haben. Doch was rein informationstechnisch Sinn macht, wird neue Regelungen bzgl. des Missbrauchs der Daten erfordern. So wird sich sicher die Frage nach dem Schutz der Daten und der Persönlichkeitsrechte in ganz neuer und besonders ausgeprägter Form stellen. Zusätzlich bedeutet die neue Mobilität eine Komplexitätssteigerung für den Nutzer bei der Auswahl von verfügbaren Diensten und Funktionen. Da die feste physische Zuordnung eines Dienstes zu einem Ort aufgrund des Wegfalls des Kabels verloren geht, werden neue Methoden zur Beschreibung der Dienste benötigt. Location-based Services helfen hierbei die Dienste mit ihren Ortsinformationen und Eigen-schaften zu annotieren und so die Auswahl gewünschter Dienste zu erleichtern.

7.6 Agentensysteme

Einen weiteren wichtigen Bereich für die Gestaltung zukünftiger Useware-Systeme stellt das Gebiet der Wissensverarbeitung dar, denn in den meisten Fällen lässt sich eine verlässliche und effiziente Erkennung multimodaler Informationen nur dann erzielen, wenn man über Wissen und Metawissen der jeweiligen Anwendungsdo-mäne bzw. Arbeitsaufgabe verfügt. Hier wird es noch großer Anstrengung bedürfen, Wissen anwendungsbezogen so effizient zu erfassen und zu speichern, dass man daraus ausreichend sichere Entscheidungen ableiten kann. Dieses ist das klassische Feld der Künstlichen Intelligenz (KI).

Eine Klasse von Systemen, die Wissen erfassen, auswerten und den Menschen wissensbasiert unterstützen, sind Agentensysteme. Agenten sind abgegrenzte Soft-

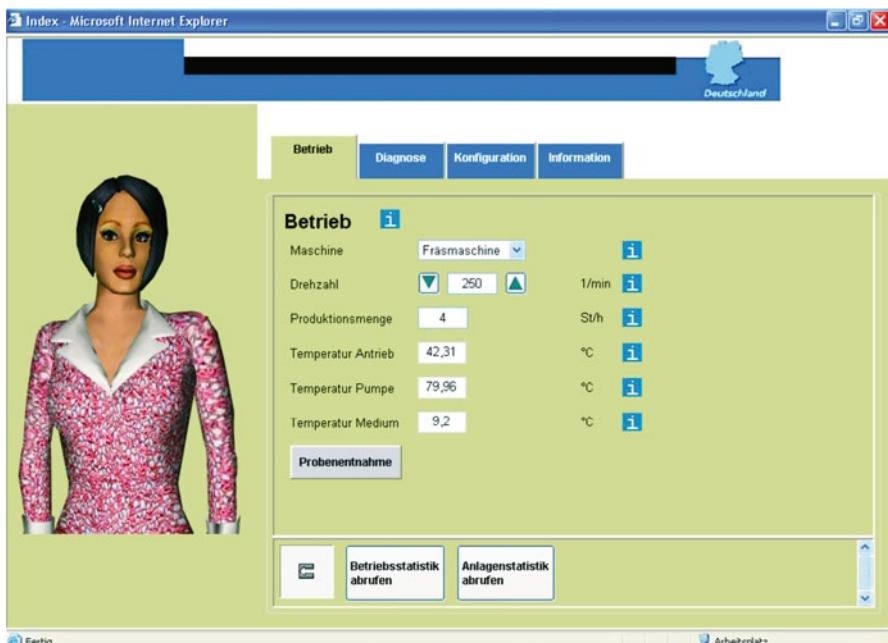


Abb. 7.8 Avatar als Unterstützung bei der Maschinenbedienung

wareentitäten, die lern- und kommunikationsfähigfähig sind, über spezialisiertes Anwendungswissen verfügen, und eigenständig Entscheidungen treffen bzw. vorbereiten können. In zukünftigen Systemen wird es eine Vielzahl solcher Agenten geben (sog. Multiagentensysteme), die den Umgang des Menschen mit technischen Systemen vereinfachen werden. So können komplexe Routineaufgaben wie das häufig zeitaufwändige Herauf- und Herunterfahren von Anlagen in der Großchemie oder der Energieerzeugung Agenten übertragen werden. Aber auch die permanente Prozessanalyse oder die Beobachtung des Nutzerverhaltens mit dem Ziel, unerlaubte Vorgehensweisen zu erkennen, werden insbesondere in komplexen Großanlagen zu einer erhöhten Sicherheit und einer einfacheren Nutzung beitragen (Ferber 2001).

Damit Agenten den Nutzer wirklich unterstützen, müssen sie „ihren“ Nutzer möglichst gut kennen. Wie die allseits bekannte Microsoft-Büroklammer uns immer wieder lehrt, wird ein Anfänger für häufigere Hinweise auf Bedienfehler sicher dankbar sein, wohingegen sich ein erfahrener Nutzer eher gestört fühlt. Dementsprechend ist es ein wichtiges Grundprinzip von Software-Agenten, das Nutzerverhalten zu beobachten, daraus zu lernen und darauf aufbauend die Kommunikation anzupassen.

Obwohl Agenten zunächst reine Softwarebausteine sind, wird ihre Erscheinungsform dem Nutzer gegenüber häufig mit der Nachbildung menschähnlicher Gestalten verknüpft. Diese virtuellen Charaktere werden Avatare genannt. Sie tauchen als bildhafte Darstellung eines Menschen oder in Form einer Comicfigur auf dem Display auf (Abb. 7.8).

Dabei ist es nur natürlich, dass damit auch die verbale und emotionale Kommunikation menschähnlich werden muss. Der Nutzer wird mit „seinem“ Avatar sprechen und erwartet langfristig nicht nur eine verständliche Sprache sondern auch die jeweils passende Gestik und Mimik. Avatare sind heute bereits als Softwarebausteine von diversen Anbietern erhältlich. Ihr Einsatz ist jedoch über einige wenige eher publicity-wirksame Anwendungen nicht hinausgekommen. Die Hauptgründe dürften einerseits in der mangelnden Emotionalität ihres Ausdrucks liegen und andererseits im mangelnden Wissen über die notwendigen Steuerinformationen. Nach einer kurzen Phase der Begeisterung empfindet der Mensch rasch Langeweile, da insbesondere die emotionalen Kommunikationselemente, die jede Mensch-Mensch-Kommunikation erst interessant machen, bei den heutigen Avataren fehlen oder nur rudimentär entwickelt sind. Damit stellt sich in der Kommunikation ein stereotyper Ablauf ein, der den Nutzer wenig anspricht. Sicher sind hier in der Zukunft schnelle Verbesserungen in der mimischen und auch sprachlichen Darstellung emotionaler Zustände zu erwarten, das weitaus schwierigere Problem liegt aber in der Steuerung solcher Ausdrucksformen. Hierzu müsste man nämlich auch den Kommunikationspartner Mensch mit seinen Emotionen erkennen und die Kommunikation entsprechend gestalten.

7.7 Emotionale Systemgestaltung

Emotionen spielen eine sehr große Rolle im Verhalten der Menschen untereinander. So spricht man sogar von der emotionalen Intelligenz, die nach dem amerikanischen Kommunikationswissenschaftler Goleman einer der wichtigsten Erfolgsfaktoren in der menschlichen Gesellschaft ist (Goleman 1997). Sieht man vielleicht von emotionalen Autos oder Uhren einmal ab, so waren Emotionen in der Technikgestaltung hingegen über viele Jahrzehnte kaum ein Thema. Doch auch hier hat sich die Welt verändert. Nachdem die Produktfunktionalität und -qualität für den Kunden heute vielfach keine Differenzierung mehr bieten, werden Produkte mehr und mehr emotional vermarktet. Das richtige Handy-Outfit zum Abendkleid, der richtige Klingelton zum Rendezvous oder das neon-erleuchtete PC-Gehäuse sind Produkte dieser Strategie.

Doch auch im Maschinen- und Anlagenbau werden die Produkte emotionaler. So wissen große Hersteller, dass emotionale Elemente im Maschinendesign mittlerweile eine genauso hohe Bedeutung bei Kunden haben, wie die technische Leistungsfähigkeit. Man spricht heute vom „Joy-of-use“, den ein Nutzer eines technischen Gerätes bei der Nutzung empfinden soll. Und in einer Welt, die emotionaler wird, wird sich dieser Trend verstärken.

Joy-of-use war bislang eher auf äußere Merkmale des Produktdesigns beschränkt. Formen, Farben und High-tech-Elemente sollten den Nutzer emotional stimulieren und ihm Freude beim Umgang mit seinen Geräten vermitteln. Nachdem aber nahezu alle Hersteller diesen Trend erkannt haben und mit entsprechenden Produkten auf dem Markt sind, wird der Schwerpunkt des Joy-of-use mehr und



Abb. 7.9 Avatare in der Maschinenbedienung

mehr auf den Bereich der Kommunikation zwischen Nutzer und Maschine gelegt werden. Und die technische Entwicklung bereitet hierfür die entsprechende Plattform. Die steten Fortschritte in der Verarbeitung und auch Generierung natürlicher Sprache werden in absehbarer Zeit nahezu umgangssprachliche Dialoge zwischen Mensch und Maschine ermöglichen. Avatare werden diese Dialoge um die visuelle Erscheinungsform ergänzen. Den Vorreiter spielt hier die Mobilkommunikation, die im UMTS-System Avatare und Sprachinteraktion verstärkt einsetzen will, um Kunden emotional zu begeistern. So werden virtuelle Charaktere mit entsprechender Sprache die bislang langweiligen SMS-Kurznachrichten in der Form „ich bin so down, ruf mich doch mal an ...“ überbringen, und dabei natürlich auch emotionale Kommunikationselemente wie Freude, Trauer, Lustlosigkeit in audio-visueller Darstellung beinhalten.

Auch wenn dieses Beispiel nicht unmittelbar auf den Maschinen- und Anlagenbau zu übertragen sein wird, so kann man sich sicher auch hier entsprechende Kommunikationsformen vorstellen. Sicher wird eine Warnung in einer technischen Anlage gesprochen von einem plötzlich auf dem Bildschirm auftauchenden virtuellen Polizisten mit erhobener Hand anders empfunden als eine reine Textmeldung, eine hübsch aussehende und freundlich sprechende virtuelle Dame könnte den Bediener auf Änderungen im Produktionsablauf angenehmer hinweisen als der garstige Meister, der den gleichen Inhalt aus 5 m Entfernung herüber brüllt. Und die lieblose textuelle Hilfemeldung könnte vielleicht durch Charaktere animiert eine viel größere Wirkung zeigen. (s. Abb. 7.9). Hiermit wäre nicht nur eine Beeinflussung des Nutzers auf der emotionalen Ebene gegeben, sondern es wären damit auch neue Formen der Aufmerksamkeitssteuerung möglich.

Allerdings fehlen noch sehr viele Voraussetzungen, um Emotionalität vor allem im Bereich der Kommunikationsgestaltung ingenieurmäßig einzusetzen zu können.

Obwohl viele der hier vorgestellten Technologien am Beginn ihrer Entwicklungsgeschichte stehen und für viele noch futuristisch klingen, werden wir ihren Einsatz immer schneller erleben, als dies früher der Fall war. Auch im eher konser-

Abb. 7.10 Die Demonstrationsanlage der SmartFactoryKL



vativen Bereich des Maschinen- und Anlagenbaus werden wir in den nächsten Jahren den Einzug solcher Technologien beobachten. Die treibende Kraft wird immer seltener die technologische Notwendigkeit als vielmehr die Produktdifferenzierung sowie die Joy-of-use-Komponente sein.

Während die in diesem Kapitel vorgestellten Technologien bereits in unserem Alltag bei einer Vielzahl von Konsumgütern wiederzufinden sind, so stellen diese in den heutigen Fabriken noch eher eine Ausnahme dar. Um diese Diskrepanz zu adressieren wurde die Technologieinitiative *SmartFactory^{KL}* (Abb. 7.10) in Kaiserslautern gegründet. Dieses realitätsnahe Labor ermöglicht die Untersuchung der Anwendbarkeit und des Nutzen dieser neuen Technologien in einer wirklichen Produktionsumgebung und somit einen direkten Erkenntnistransfer in reale Fabriken.

Anhand der Demonstrationsanlage wird beispielsweise untersucht, wie Wartungstechniker mittels intelligenter Sensorsysteme in ihren Arbeitsabläufen besser unterstützt werden können. So können beispielsweise Kontextinformationen, die Aufschluss über die aktuelle Position des Benutzers geben, diesen bei der Suche nach einem defekten Feldgerät unterstützen. Der Einsatz von mobilen Bediengeräten, z. B. in Form von Smartphones und Tablet PCs, stellt einen weiteren Forschungsschwerpunkt in der *SmartFactory^{KL}* dar. Diese Bediengeräte ermöglichen einen homogenen Zugriff auf diverse Feldgeräte der Anlage mittels moderner Funktechnologiestandards wie Bluetooth und WLAN. Neuartige Paradigmen, wie z. B. Augmented Reality, gewinnen ebenfalls immer mehr an Bedeutung bei der Mensch-Maschine-Interaktion in modernen Fabriken. Diese helfen den Bruch zwischen der realen und der digitalen Welt schrittweise zu überbrücken und ermöglichen es somit die Bedienung in zukünftigen Fabriken noch einfacher zu gestalten.

Literatur

- Abrams M, Phanouriou C (1999) UIML An XML Language for building device-independent user-interfaces. In: Proceedings of XML'99, Philadelphia (05.-09.12.1999). GCA, Alexandria
- Ali M, Abrams M (2001) Simplifying Construction of Multi-Platform User Interfaces using UIML. In: Proceedings of UIML Europe 2001 conference (08.-09.03.2001), ENSAM, Paris
- Ali M et al (2002) Building multi-plattform user interfaces with UIML. In: Kolski C, Vanderdonckt J (Hrsg) Proceedings of the 4th international conference on Computer-Aided Design of User Interfaces (CADUI'2002). Kluwer, Dordrecht
- Balzert H (1996) Lehrbuch der Software-Technik: Software-Entwicklung. Spektrum, Berlin
- Bauer EW (1981) Humanbiologie. Cornelson-Velhagen & Klasing, Berlin
- Birbaumer N, Schmidt RF (1991) Biologische Psychologie. Springer, Heidelberg
- Blau JR (1987) Entwurf eines Beratungssystems zur Diagnose und Prognose der operationellen Kompetenz des Operators in Mensch-Maschine-Systemen. In: Messen-Steuern-Regeln (msr), Nr. 30
- Bomsdorf B, Szwilus G (2001) UML und Aufgabenmodellierung: Softwaretechnik und HCI im Dialog. In: Oberquelle H, Oppermann R, Krause J (Hrsg) Mensch & Computer 2001: 1. Fach-übergreifende Konferenz. Teubner, Stuttgart
- Bödcher A (2007) Methodische Nutzungskontextanalyse als Grundlage eines strukturierten USE-WARE-Engineering-Prozesses. Fortschritt-Bericht pak, Bd 14. TU Kaiserslautern, Kaiserslautern
- Bourges-Waldegg P (2000) Globalization: a threat to cultural diversity? In: Day D, del Galdo EM, Prabhu GV (Hrsg) Designing for global markets 2, 2nd international workshop on Internationalisation of Products and Systems (IWIPS 2000, 13.-15. July). Backhouse Press, Baltimore
- Bubb H, Eiff W (Hrsg) (1992) Innovative Arbeitssystemgestaltung. Mensch, Organisation, Information und Technik in der Wertschöpfungskette. Wirtschaftsverlag Bachern, Köln
- Bullinger HJ (1994) Ergonomie – Produkt- und Arbeitsplatzgestaltung. Teubner, Stuttgart
- Burmester M, Komischke T (1999) Nutzeranforderungen als Grundlage für die Entwicklung innovativer User Interfaces in der industriellen Prozessführung. In: Arend U, Eberleh E, Pitschke K (Hrsg) Software-Ergonomie '99: Design von Informationswelten. Fachtagung des German Chapter der ACM, der Gesellschaft für Informatik (GI) und der SAP AG, Walldorf, 08.–11.03.1999. Teubner, Stuttgart
- Burmester M, Hassenzahl M, Koller F (2002) Usability ist nicht alles – Wege zu attraktiven Produkten. i-com – Z interaktive kooperative Medien 1(1):32 (Oldenbourg, München)
- Bruder R, Leuchter S, Röse K, Schmidt L, Urbas L (2002) Von MMI-interaktiv Forschung zu Ergonauten: Entwicklung eines nutzeradaptiven Web-Portals. In: USEWARE 2002. Mensch-Maschine-Kommunikation/Design. Tagung Darmstadt, 11./12. Juni 2002, VDI-Bericht 1678, VDI/VDE-Gesellschaft für Mess- und Automatisierungstechnik. VDI, Düsseldorf
- Cadera T, Schäflein-Armbruster R, Schweitzer M (2002) Der erste Anwender: Gemeinsame Ziele, Methoden und Arbeitskriterien. In: technische kommunikation, Nr. 6. Schmidt-Römhild, Lübeck

- Card S-K, Moran T-P, Newell A (1983) *The psychology of human-computer-interaction*. Lawrence Erlbaum Associates, New Jersey
- Charwat H-J (1994) Lexikon der Mensch-Maschine-Kommunikation. Oldenbourg, München
- Day D-L (1996) Cultural bases of interface acceptance: foundations. In: Sasse M-A, Cunningham R-J, Winder R-L (Hrsg) *Proceedings of the British Computer Society Human Computer Interaction Specialist Group, People and Computers XI*. Springer, London
- DeKeyser V (1986) Technical assistance to the operator in case of incident – Some lines of thought. In: Hollnagel G, Mancini G, Woods DD (Hrsg) *Intelligent decision support in process environments*. Springer, Berlin
- DFN (2003) Abschlussbericht des DFN-Projektes „MMI interaktiv/useworld.net – Entwicklung und Betrieb eines föderierten, nutzeradaptiven Web-Portals für Mensch-Maschine-Interaktions-Forschung“. <http://www.useworld.net>
- Diekmann (2000) *Handbuch der Demographie*. Springer, Berlin
- Dinges W (1982) Das Verkehrsflugzeug als Mensch-Maschine-System. Air- Report- Verlag, Rossdorf
- DIN33414 – Ergonomische Gestaltung von Warten. Beuth, Berlin
- DIN66234-8 (1988) Norm DIN 66234 Teil 8 02.88: Bildschirmarbeitsplätze: Grundsätze ergonomischer Dialoggestaltung
- Dix A, Finlay J, Abowd GD, Beale R (2004) *Human-Computer Interaction*. Pearson Eductaion, Harlow
- Dörner D (2003) *Die Logik des Misslingens*. Rowohlt, Berlin
- Dong J, Salvendy G (1999) Designing menus for the Chinese population: horizontal or vertical? *Behav Inf Technol* 18(6):467–471
- Dutke S (1994) *Mentale Modelle: Konstrukte des Wissens und Verstehens*, Bd 4. Hogrefe, Göttingen
- Eibl-Eibesfeldt I (1984) *Die Biologie des menschlichen Verhaltens*. Piper, München
- Eissler R (2001) *Useware-Engineering in der Produktionstechnik: Systemtechnische Methodik zur Entwicklung ergonomischer Bediensysteme für komplexe Maschinen*. Fortschritt-Berichte pak, Bd 4. TU Kaiserslautern, Kaiserslautern
- Ferber J (2001) *Multiagentensysteme – Eine Einführung*. Addison-Wesley, New York
- Fitts PM (1954) The information capacity of the human motor system in controlling the amplitude of movement. *J Exp Psychol* 47(6):381–391
- Gauer OH, Kramer K, Jung R (1972) *Somatische Sensibilität, Geruch und Geschmack: Sinnesphysiologie I*. Bd 11. Urban & Schwarzenberg, München
- Geiser G (1990) Mensch-Maschine-Kommunikation. Oldenbourg, München
- Geis T, Hartwig R (1998) Auf die Finger geschaut: Neue ISO-Norm für nutzergerechte interaktive Systeme. c't 1998(14):168–171 (Heise, Hannover)
- Goleman D, Griese F (1997) *Emotionale Intelligenz*, EQ. DTV, München
- Goldstein EB (1997) *Wahrnehmungspsychologie – Eine Einführung*. Spektrum, Heidelberg
- Gorin A (2003) Semantic information processing of spoken language – how may I help you? In: *Proceedings of the international conference on Intelligent User Interfaces*. ACM, New York.
- Hacker W (1978) *Allgemeine Arbeits- und Ingenieurpsychologie – Psychische Struktur und Regulation von Arbeitstätigkeiten*. Hans Huber, Bern
- Hacker W (1986) *Arbeitspsychologie*. VEB Deutscher Verlag der Wissenschaft, Berlin
- Hamburg K-C, Schweppenhäuser A (1993) Zur Bedeutung psychologischer Arbeits- und Aufgabenanalyse für die Softwaregestaltung. In: Rödinger K-H (Hrsg) *Software-Ergonomie '93: Von der Benutzungsoberfläche zur Arbeitsgestaltung*. Fachtagung des German Chapter der ACM, Bremen, 15.-17.03.1993. Teubner, Stuttgart
- Hamming RW (1987) *Information und Codierung*. VCH, Weinheim
- Hassenzahl M, Beu A, Burmester M (2001) Engineering joy. *IEEE Software* 18(1):70–76
- Herczeg M (1994) *Software-Ergonomie – Grundlagen der Mensch-Computer-Kommunikation*. Addison-Wesley, Bonn
- Heuer H, Kleinbeck U, Schmidt KH (1987) *Motor behaviour – programming, control and acquisition*. Springer, New York

- Heuer J (2003) Expertenevaluation. In: Heinzen S, Vogt P (Hrsg) Usability praktisch umsetzen: Handbuch für Software, Web, Mobile Devices und andere interaktive Produkte. Hanser, München
- Hofvenschiodl E (2003) Determining Cultural Issues in Attitude to and Use of Mobile Phones. In: Peissner M, Röse K (Hrsg) Usability Professionals 2003. Berichtband des 1. Workshops des German Chapters der Usability Professionals Association e. V., 08.-10.09. 2003. Stuttgart
- Holz auf der Heide B (1993) Welche software-ergonomischen Evaluationsverfahren können was leisten? In: Rödiger KH (Hrsg) Software-Ergonomie '93: Von der Nutzer-oberfläche zur Arbeitsgestaltung. Teubner, Stuttgart,
- Holzinger A (2003) Finger instead of mouse: touch screens as a means of enhancing universal access. In: Carbonell N, Stephanidis C (Hrsg) Universal access, theoretical perspectives, practice, and experience. Springer, Berlin
- Hom J (2011) The usability methods toolbox. <http://usability.jameshom.com/>. Zugegriffen: 20. Jan 2011
- Hußmann H, Meixner G, Zühlke D (Hrsg) (2011) Model-Driven Development of Advanced User Interfaces. In: Studies in Computational Intelligence (SCI). Springer, Heidelberg
- HÜMNOS (1997) Style-Guide Werkzeugmaschinen – Ein Handbuch zur Gestaltung von Benutzungsoberflächen für Werkzeugmaschinen. Fraunhofer IRB, Stuttgart
- Ikonen V, Suihkonen R, Jalonens K, Laitakari I, Salovaara A (2003) Developing personal navigation products for professionals – a methodological perspective. In: Evers V, Röse K, Honold P, Coronado J, Day D (Hrsg) Designing for global markets – 5. workshop proceedings of the 5th international workshop on Internationalisation of Products and Systems, IWIPS 2003, 17.–19. July 2003. Berlin
- ISO 9241 – Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten. Beuth, Berlin
- Johannsen G (1993) Mensch-Maschine-Systeme. Springer, Berlin
- Katz D (1969) Der Aufbau der Tastwelt. Johann Ambrosius Barth, Leipzig
- Krauss L (2003) Entwicklung und Evaluation einer Methodik zur Untersuchung von Interaktionsgeräten für Maschinen- und Prozessbediensysteme mit grafischen Benutzungsoberflächen. Fortschritt-Bericht pak, Bd 7. TU Kaiserslautern, Kaiserslautern
- Kuchling H (2001) Taschenbuch der Physik. Fachbuchverlag, Leipzig
- Kurze M (1998) Methoden zur computergenerierten Darstellung räumlicher Gegenstände für Blinde auf taktilen Medien. Freie Universität Berlin, Berlin
- Leuchter S, Urbas L, Röse K (2003) Engineering and evaluation of community support in use-world.net. In: Stephanidis C, Jacko J (Hrsg) Human-computer interaction: theorie and practice (Part II). Proceedings of the HCI International 2003 10th international conference on human-computer interaction, symposium on human interface (Japan) 2003, 5th international conference on Engineering Psychology and Cognitive Ergonomics, 2nd International Conference on Universal Access in Human-Computer Interaction, 22–27 June. Erlbaum, Crete
- Lewin K (1942) Feldtheorien und Lernen. In: Cartwright D (Hrsg) Feldtheorie in den Sozialwissenschaften. Hans Huber, Bern (1963)
- Luyten K, Coninx K (2001) An XML-based runtime user interface description language for mobile computing devices. In: Johnson C (Hrsg) Proceedings of DSV-IS 2001– Design, Specification and Verification of interactive systems. Springer, Wien
- Maletzke G (1996) Interkulturelle Kommunikation: Zur Interaktion zwischen Menschen verschiedener Kulturen. Westdeutscher Verlag, Opladen
- Meixner G, Görlich D, Schäfer R (2008) Unterstützung des Useware-Engineering Prozesses durch den Einsatz einer modellbasierten Werkzeugkette, USEWARE 2008. Tagung, Baden-Baden, VDI-Bericht 2041, VDI/VDE-Gesellschaft für Mess- und Automatisierungstechnik, Düsseldorf, Germany. VDI, Düsseldorf, S219–232
- Meixner G, Schäfer R (2009) Modellbasierte Entwicklung von Benutzungsschnittstellen mit UIML. i-com 8(1):60–67
- Meixner G (2010) Entwicklung einer modellbasierten Architektur für multimodale Benutzungsschnittstellen. Fortschritt-Bericht pak, Bd 21. TU Kaiserslautern, Kaiserslautern

- Matlin MW, Foley HJ (1991) Sensation and perception – the skin senses (Chapter 12). Allyn and Bacon, Needham Heights
- Mörike KD, Betz E, Mergenthaler W (1989) Biologie des Menschen. Quelle & Meyer, Heidelberg
- Moser H (2003) Instrumentenkoffer für die Praxisforschung. Pestalozianum, Zürich
- Müller A (2001) Model-based user interface design using markup concepts. In: Johnson C (Hrsg) Proceedings of DSV-IS 2001 – design, specification and verification of interactive systems. Springer, Wien
- Müller W, Schäfer R, Bleul S (2004) Interactive multimodal user interfaces for mobile devices. Proceedings of the 37th Annual Hawaii International Conference on System Sciences
- Muthig KP (1990) Informationsaufnahme und Informationsverarbeitung. In: Graf Hoyos C, Zimolong B (Hrsg) Ingenieurpsychologie. Hofgrefe, Göttingen
- Myers B (1996) UIML, toolkits, interface builders. Carnegie Mellon University (Human Computer Interaction Institute), Pittsburgh
- Neumann H (2002) Analyse und Entwurf von Softwaresystemen mit der UML. Hanser, München
- Nielsen J (1993) Usability Engineering. 1. Aufl. AP Professional, Boston
- Nielsen J (1994) Heuristic evaluation. In: Nielsen J and Mack R-L (Hrsg) Usability Inspection Methods. Wiley, New York
- Norman D (1989) The design of everyday things. MIT-Press, Cambridge
- Norman D (1983) Some observations on mental models. In: Gentner S Mental models. Lawrence Erlbaum Associates, Hillsdale
- Norman D (1998) The invisible computer. MIT-Press, Cambridge
- Oortmann H (2000) Das Bediensystem von morgen – eine Umfrage. In: Computer&Automation, Nr. 06
- Oortmann H (2007) Workspace – vom integrierten Plattformkonzept für die Produktion zum exemplarischen 3D-Userinterface. Fortschritt-Bericht pak, Bd 13. TU Kaiserslautern, Kaiserslautern
- Ortlib S, Holz auf der Heide B (1993) Nutzer bei der Software-Entwicklung angemessen beteiligen: Erfahrungen und Ergebnisse mit verschiedenen Konzepten. In: Rödinger K-H (Hrsg) Software-Ergonomie '93: Von der Benutzungsoberfläche zur Arbeitsgestaltung. Fachtagung des German Chapter der ACM, Bremen, 15.–17.03.1993. Teubner, Stuttgart
- Oschatz S (2000) Erkennbarkeit (Vorlesung Interfacedesign), Universität Ulm, <http://aws-1.e-technik.uni-ulm.de/idesign/vorlesung/chapter4sl.html>
- Paternò F, Santoro C (2002) One model, many interfaces. In: Kolski C, Vanderdonckt J (Hrsg) Proceedings of the Fourth International Conference on Computer-Aided Design of User Interfaces (CADUI'2002). Kluwer Academic, Dordrecht
- Piamonte DPT, Abeysekera JDA, Ohlsson K (1999) Testing Videophone Graphical Symbols in Southeast Asia. In: Bullinger H-J, Ziegler J (Hrsg) Human-computer interaction: ergonomics and user interfaces, (Bd 1). Proceedings 8th International Conference on Human-Computer Interaction (HCI International '99, Munich, Germany, August 22–26)
- Preece J (1994) Human-Computer Interaction. Addison Wesley, Harlow
- Piera C (2002) Leonardo da Vinci y la cuadratura humana. <http://webs.adam.es/rillorens/picuad/leonardo.htm>, Madrid
- Preim B (1999) Entwicklung interaktiver Systeme: Grundlagen, Fallbeispiele und innovative Anwendungsfelder, 1. Aufl. Springer, Berlin
- Pschyrembel W (1990) Klinisches Wörterbuch. deGruyter, Berlin
- Rachui T (1998) Programmieren unter grafischen Benutzungsoberflächen, Universität Rostock. www-md.e-technik.uni-rostock.de/ma/wb68/vorlesungsfolien/sld075.htm
- Razi N (2002) Nutzer als Redakteure – Zur Problematik der Motivation bei Online-Arbeitsgemeinschaften. Studienarbeit ZMMS TU, Berlin
- Rasmussen J (1983) Skills, rules, signals, signs and symbols and other distinctions in human performance. IEEE-Transactions SMC-13(3):257–266
- Rasmussen J (1986) Information processing and human-machine interaction. North-Holland, New York

- Rasmussen J (2000) Human factors in a dynamic information society – where are we heading? *Ergonomics* 43(7):869–879
- Rauterberg M (1994) Nutzerorientierte Software-Entwicklung – Konzepte, Methoden und Vorgehen zur Nutzerbeteiligung. Mensch, Technik, Organisation; Bd 3. Teubner, Stuttgart
- Reuther A (2003) useML – Systematische Entwicklung von Maschinenbediensystemen mit XML. Fortschritt-Bericht pak, Bd 8. TU Kaiserslautern, Kaiserslautern
- Röse K (2001) Internationalization and localization of products: influences on the user interfaceengineering process. In: HCI International 2001. 9th International Conference on Human-Computer Interaction, Symposium on Human Interface (Japan) 2002, 4th International Conferenceon Engineering Psychology and Cognitive Ergonomics, 1th International Conference onUniversal Access in Human-Computer Interaction, Poster Session: Abridged
- Röse K (2002) Methodik zur Gestaltung interkultureller Mensch-Maschine-Systeme in der Produktionstechnik. Fortschritt-Bericht pak, Bd 5. TU Kaiserslautern, Kaiserslautern
- Röse K (2003a) Interkulturelle Aspekte der Ergonomie bei der Gestaltung interaktiver Produkte im produktionstechnischen Umfeld. In: Landau K (Hrsg) Good Practice – Arbeitsgestaltung und Ergonomie. Sonderausgabe der Zeitschrift für Arbeitswissenschaft anlässlich des 50jährigen Bestehens der Gesellschaft für Arbeitswissenschaften. Ergonomia Verlag oHG, Stuttgart
- Röse K (2003b) Task-Analyse. In: Heinsen S, Vogt P (Hrsg) Usability praktisch umsetzen: Handbuch für Software, Web, Mobile Devices und andere interaktive Produkte. Hanser, München
- Röse K, Liu L (2002) The understanding of english terms in mainland china: an empirical study. In: Coronado J, Day D, Hall B Designing for global markets 4, Fourth International Workshop on Internationalisation of Products and Systems, 11th to 13th July. Austin, Texas
- Röse K, Ziegeler D (2003) Entwurfsmethoden und –prinzipien. Vorlesungsskript zur Vorlesung Grundlagen der Mensch-Maschine-Interaktion, Kapitel 5, FB MV, TU Kaiserslautern
- Röse K, Liu L, Zühlke D (2002) Analysis and structuring of the interaction areas for a chinese userinterface. In: Luczak H, Cakir A, Cakir A (2002) WWDU, Work With Display Units, WorldWide Work. Proceedings of the 6th International Scientific Conference on Work with DisplayUnits, May 22–25, Berchtesgaden
- Röse K, Liu L, Zühlke D (2003) Design issues in mainland china and western europe: similaritiesand differences in the area of human-machine-interaction design. In: Smith MJ, Salvendy G(Hrsg) Systems, social and internationalization design aspects of human-computer interaction. Volume 2 of the Proceedings of HCI International 2001, 9th International Conference on Human-Computer Interaction, Symposium on Human Interface (Japan) 2002, 4th InternationalConference on Engineering Psychology and Cognitive Ergonomics, 1th International Conferenceon Universal Access in Human-Computer Interaction, 5–10 August 2001, New Orleans, Louisiana, USA
- Rothmund T (2000) Nutzerseitige Anforderungsanalyse an eine Informations- und Kooperationsplattform für Wissenschaftler und Praktiker auf dem Gebiet der Mensch-Maschine-Interaktion. http://www.mmi-interaktiv.de/ausgaben/07_01/rothmund.pdf. Zugriffen: 08. Sept. 2002
- Rubin J (1994) Handbook of usability testing: how to plan, design and conduct effective tests. Wiley, New York
- Rühmann H, Schmidtke H (1992) Körperkräfte des Menschen – Perzentilierung isometrischer Maximalkräfte sowie Ausdauer und Beanspruchung bei konzentrischer und exzentrischer Muskelarbeit. Bundesamt für Wehrtechnik und Beschaffung, München
- Schäfer R, Müller W, Marín A et al (2007) Using smart cards for secure and device independent user interfaces. Proceedings of the International Conference on Mobile Technology, Applications and Systems (Mobility 2007)
- Schenk J, Rigoll G (2010) Mensch-Maschine-Kommunikation Grundlagen von sprach- und bildbasierten Benutzerschnittstellen. Springer, Berlin
- Shneiderman B, Plaisant C (2009) Designing the user-interface: strategies for effective human-computer-interaction. Addison Wesley, Massachusetts
- Snyder HL (1988) Image quality. In: Helander M (Hrsg) Handbook of Human-computer-interaction. Elsevier Science Publishers, Amsterdam, S 437–474

- Souchon N, Vanderdockt J (2003) A review of XML-compliant user interface de-scription languages. Proceedings of the 10th International Workshop on Interactive Systems: Design, Specification and Verification, S 377–391
- Streitz NA (1988) Fragestellungen und Forschungsstrategien der Software-Ergonomie. In: Balzert H (Hrsg) Einführung in die Software-Ergonomie. deGruyter, Berlin
- Thiels N (2008) Personalisierung von technischen Maschinenbenutzungs-oberflächen auf Grundlage einer erweiterten Analysephase. Fortschritt-Bericht pak, Bd 18. TU Kaiserslautern, Kaiserslautern
- Thimbleby H (1990) User interface design. acm Press, New York
- VDI/VDE 3850 – Nutzergerechte Gestaltung von Bediensystemen für Maschinen. Beuth, Berlin
- VDI 4500 – Technische Dokumentation: Nutzerinformation. Beuth, Berlin
- Völz H (1999) Das Mensch-Technik-System – Physiologische, physikalische und technische Grundlagen. Expert, Renningen-Malmsheim
- Wahl M (2000) Systematische Entwicklung nutzergerechter Maschinenbedienoberflächen: Ein Software-Werkzeug mit integrierten Ergonomie- und Gestaltungsregeln. Fortschritt-Bericht pak, Bd 3. TU Kaiserslautern, Kaiserslautern
- Wahl M (2000) Systematische Entwicklung nutzergerechter Maschinenbedienoberflächen: Ein Software-Werkzeug mit integrierten Ergonomie- und Gestaltungsregeln. Fortschritt-Berichte pak, Bd 3. TU Kaiserslautern, Kaiserslautern
- Wahrlich H (1991) Wortlose Sprache – Verständnis und Missverständnis im Kulturkontakt. In: Thomas A (Hrsg) Kulturstandards in der internationalen Begegnung. ssip bulletin No. 61. Verlag für Entwicklungspolitik Saarbrücken GmbH, Saarbrücken
- Wandke H (2003) Methoden der Usability Evaluation. Vorlesungsunterlagen Humboldt Universität Berlin. <http://arb1.psychologie.hu-berlin.de/ingpsy/>
- Wandmacher J (1993) Software-Ergonomie. Walter de Gruyter, Berlin
- Wandmacher J (1994) Software-Ergonomie. deGruyter, Berlin
- Weidenmann B (1999) Die Psychologie des Nichtverstehens. In: tekom-Schriften zur technischen Kommunikation, Bd 1: Verständlichkeit und Nutzungsfreundlichkeit von technischen Dokumentationen. Schmidt-Römhild, Lübeck
- Wessel I (1998) GUI-Design: Richtlinien zur Gestaltung ergonomischer Windows-Applikationen. Hanser, München
- Zapf DH, Brodbeck FC, Prümper J (1989) Handlungsorientierte Fehlertaxonomie in der Mensch-Computer-Interaktion. Zeitschrift für Arbeits- und Organisationspsychologie, 33(4). Hogrefe Verlag, Göttingen
- Zeidler A, Zellner R (1994) Software-Ergonomie – Techniken der Dialoggestaltung. Oldenbourg, München
- Zühlke D (2005) Der intelligente Versager: Das Mensch-Technik-Dilemma. Primus Verlag, Darmstadt
- Zühlke D (2010) Gestaltung ergonomischer Mensch-Maschine-Schnittstellen für komplexe technische Systeme. ZMMI-Seminar, Kaiserslautern, 08.–09.12.2010. ZMMI, Kaiserslautern
- Zühlke D, Romberg M, Meil P (1998) Anforderungen außereuropäischer Märkte an die Gestaltung der Maschinenbedienung (INTOPS). VDI-Fortschritt-Berichte 2/485, VDI, Düsseldorf
- Zühlke D, Romberg M, Röse K (1998) Global Demands of Non-European Markets for the Design of User-Interfaces. (7th IFAC/IFIP/IFORS/IEA Symposium, Analysis, Design and Evaluation of Man-Machine Systems Kyoto, Japan 16–18.09.1998). Kyoto: IFAC, Hokuto Print: Japan
- Zwiesler R (1998) Virtuelle Realität und die Rolle von Haptik. Universität Regensburg, Regensburg

Sachverzeichnis

A

- Abkürzung, 100
- Adaption, 8
- Akkommodation, 9
- Aktivitätsdiagramm, 155
- Aktor, 6, 18
- Alarm, 81
- Allgemeine Informationen, 81
- Analyse, 39, 163
- Ändern, 55
- Anwendungsfalldiagramm, 153
- Anwendungsfenster, 85
- Arbeitsbereich, 87, 199
- Auditives System, 26
- Aufgabenangemessenheit, 79, 184
- Aufmerksamkeit, 18, 27
- Aufmerksamkeitssteuerung, 27, 81
- Auslösen, 54
- Auswählen, 55
- Auswahlfeld, 206

B

- Baumdarstellung, 77
- Baumstruktur, 75
- Bedienfehlerinformation, 81
- Bedienpriorität, 218
- Bediensystem
 - primäres, 61
 - sekundäres, 61
 - tertiäres, 61
- Bediensystemgestaltung, 46, 184
- Beleuchtungsstärke, 9
- Benutzungsmodell, 51, 56, 62, 75, 76
- Benutzungsobjekt, 53, 54
 - elementares, 53
- Beobachtung, teilnehmende, 47
- Betriebsverfassungsgesetz, 49
- Bewegungsfehler, 29
 - bewegungsinduzierte Tiefenkriterien, 22

- Bildschirmlayout, 86, 104
- Bildwiederholfrequenz, 10
- Bildzeichen, 96
- Bildzeichenfamilien, 99

C

- C++, 144, 150
- Checkbox, 206
- China, 131
- Chunks, 19
- CIM, 28, 158
- CRT, 222
- Customizing, 123

D

- Darstellungsattribut, 82
- DELPHI, 144
- Denkfehler, 29
- Design, interkulturelles, 124, 130
- Design-Guide, 103
- DEVICE, 224
- Dialog
 - Formular, 83
 - Frage-Antwort, 83
 - Funktionstasten, 83
 - Kommando, 83
 - Menü, 83
 - natürlichsprachlicher, 83
- Dialogebene, 84
- Dialogfenster, 85
- Dialogform, 83
- Differenzial, semantisches, 115
- DIN EN ISO 9241, 178
- Direkte Manipulation, 83, 201
- Dreh-Druck-Steller, 205

E

- Ease-of-Use, 115
- Eingabefokus, 85

Eingeben, 55
 Entwicklermodell, 25
 Entwicklungsumgebung, 105, 149
 Entwicklungswerkzeug, modellbasiertes, 106
 Ergonomie, 92, 108
 Erkennensfehler, 29
 Erkundungsmethode, 108
 Erwartungskonformität, 185
 Evaluation, 37
 analysierende, 108
 bewertende, 107
 empirische, 108
 formale, 108
 heuristische, 108
 iterative, 123
 vergleichende, 107
 Excel, 64
 Exterozeption, 14

F

Farbenblindheit, 8
 Farbkodierung, 136, 138
 absolute, 93
 relative, 93
 Farbkontrast, 93
 Farbsehschwäche, 92
 Feedback, 26, 30, 95, 222
 Handlungs-, 26
 Status-, 26
 Ziel-, 26
 Fehlermeldebereich, 196
 Fehlertaxonomie, 29
 Fehlertoleranz, 185
 Feingestaltungsphase, 91
 Figur-Grund-Beziehung, 193
 FITTS LAW, 224
 Fließbild, 183
 Flimmern, 10
 Fokusgruppe, 113
 Fokussteuerung, 85
 Fragebogen, 47, 114
 Freies Interview, 47

G

Gehörsinn, 8
 Geruchssinn, 8
 Geschlossenheit, 191
 Gesichtssinn, 8
 Gestaltgesetz, 190
 Gewerkschaft, 44
 Gewohnheitsfehler, 29
 Gleichartigkeit, 194
 Gleichgewichtssinn, 8

Globalisierung, 126
 GOMS-Modell, 109
 Grafiktablett, 211
 Groblayout, 67
 Gruppierung, 90, 190

H

Hamming-Distanz, 99
 Handeln, 6
 Handlungsmuster, 20
 Helligkeitsstufe, 93
 Hilfeinformation, 81
 Hinweisbereich, 87
 Hinweisinformation, 81
 HMD, 223
 Hörschwelle, 13
 Human-Centered-Design, 5
 HÜMNOS, 1, 200

I

Icon, 20, 98, 137
 Ideogramm, 98
 Indien, 131
 Indonesien, 131
 Informationsklasse, 17, 80
 Informieren, 54
 Interaktion
 globale, 60
 lokale, 60
 regionale, 60
 Interkulturelles Design, 124, 130
 Interlacing, 10
 Internationalisierung, 126
 Interview, 114
 freies, 47
 strukturiertes, 46
 INTOPPS, 124, 131, 132
 INTOUCH, 147

J

JAVA, 144, 150
 Joy-of-use, 114
 Joystick, 211

K

Klassendiagramm, 153
 Kognition, 15, 22
 Kommandosprache, 195
 Kompatibilität, 90, 188
 Konsistenz, 90, 188
 Kontrollmethode, 108
 Korea, 131
 Kurzzeitgedächtnis, 7, 16

L

- Langzeitgedächtnis, 17
- LCD, 222
- Lean-Production, 28
- Lernförderlichkeit, 187
- Leserichtung, 137
- Leuchtdichte, 223
- Leuchtdichtheitkontrast, 93
- Likert-Skala, 115
- Linkshänder, 198
- LogFile, 115
- Lokalisation, 124

M

- Mapping, 17, 20
- Maskendialog, 196
- Maus, 210
- Menü, 78, 83
- Menüstruktur, 87
- Merk-/Vergessensfehler, 29
- Merkmalsbildung, 17
- Methoden-Mix, 46
- Mitteilungsfenster, 85
- Modell, mentales, 25, 48
- Moderator, 38
- Motorik, 6
- Mousebutton, 211
- Mousepad, 210
- Mousstick, 211

N

- Nähe, 191
- Nahpunkt, 9
- Navigation
 - absolute, 40
 - relative, 40
- Navigationsbereich, 87, 198
- Navigationsstruktur, 75
- Navigationstaste, 203
- Netzhaut, 9
- Netzwerkstruktur, 75
- Normen, 91
- Nutzerevaluation, 106
- Nutzergruppen, 41, 44, 61, 89
- Nutzermodell, 25

O

- Observation, 115
- Odilienberg, 28
- OPC, 69
- Ortsfrequenz, 11

P

- pak-PDI, 225
- Piktogramm, 20, 97
- Pilot-Test, 116
- Pointing Device Index, 225
- Positivdarstellung, 8, 12
- POWERPOINT, 150
- Prägnanz, 191
- Proportionalschrift, 96
- Propriozeption, 15
- Prototyping, 55, 107
- Prozesseleittechnik, 183

Q

- Qualität, hedonistische, 91
- Querdisparation, 22

R

- Radiobutton, 55, 206
- Rapid-Prototyping, 149
- Rasterschrift, 96
- Realisierung, 37
- Redakteur, 36
- Register, 207
- Regulationsebene, 22
- Retina, 9
- Rezeptor, 9
- Richtlinie, 91

S

- Schalldruck, 12
- Schalldruckpegel, 12
- Schallfrequenz, 12
- Schalter, 203
- Schaltergruppe, 203
- Schalterleiste, 78
- Schmerzgrenze, 13
- Schraffur, 96
- Sehfeld, 11
- Sehschärfe, 9
- Selbstbeschreibungsfähigkeit, 185
- Sensorik, 6
- Shutter-Glasses, 22
- Sinnbedeutung, 193
- Slider, 89
- Softkey, 72
- Stäbchen, 9
- Statusbereich, 87, 197
- Statusinformation, 81
- Steuerbarkeit, 186
- Steuerungsplattform, 68
- Stressoren, kulturelle, 128

Strukturgestaltung, 37, 49, 50, 62
 Strukturiertes Interview, 46
 Strukturlegetechnik, 47
 Style-Guide, 103
 System
 auditivs, 11
 haptisches, 14
 Systemdynamik, 30
 Systemintransparenz, 30
 Systemkomplexität, 30

T

Task-Szenario, 117
 Tastaturfokus, 85
 Tastenbedienung, 67
 Taster, 201
 Tastsinn, 8, 15
 TCP/IP, 69
 Team-Management, 33
 Testmethode, 108
 Thinking-Aloud-Methode, 113
 Three-Mile-Island, 28
 Tiefenkriterien
 monokulare, 21
 okulomotorische, 21
 Touchscreen, 72, 183, 212
 Trackball, 210
 Tschernobyl, 29

U

Überforderung, 33
 Ultrakurzzeitgedächtnis, 16
 UML, 151
 Unterforderung, 34
 Unterlassensfehler, 29
 Unterstreichung, 96
 Urteilsfehler, 29

USA, 131
 Usability, 35
 Testing, 108
 useML, 65
 USEWARE, 2
 Useware Engineering, 2

V

VDI
 3699, 183
 3850, 181
 Verhaltensebene
 regelbasierte, 24
 sensomotorische, 22
 wissensbasierte, 22
 Versagen, menschliches, 28
 VISUAL-BASIC, 144
 Visualisierungspriorität, 218

W

Wahrnehmung, 6
 räumliche, 21
 Warnung, 81
 Wasserfallmodell, 36, 106
 Widget, 144
 WIMP, 71
 WinCC, 147
 Wissensfehler, 29

X

XML, 106, 143

Z

Zäpfchen, 9
 Zustandssteuerung, 85
 Zuverlässigkeit, menschliche, 30