

Projektberichte zum Forschungsprojekt

Benutzer- **O**rientierte **S**oftwareentwicklung und **S**chnittstellengestaltung

Herausgegeben von:

P. Spinas, M.Rauterberg, O.Strohm, D.Waeber & E.Ulich

Projektbericht Nr. 7

Ein quantitatives Bewertungssystem zur
software-ergonomischen Beurteilung von
Benutzungsoberflächen interaktiver Software.

Matthias Rauterberg

Zürich

1991

**Vom Bundesministerium für Forschung und Technologie
gefördertes Verbundprojekt**

Förderkennzeichen: 01HK706/0

Projekttitel:

**"Entwicklung und empirische Überprüfung von Kriterien, Methoden und Modellen
zur benutzerorientierten Software-Entwicklung und Dialog-Gestaltung"**

Projektträger:

**DFVLR
Projektträgerschaft
Arbeit und Technik (AuT)
Südstrasse 125
D-5300 BONN 2**

Verbundpartner:

IfAP-ETH, Zürich (CH)



Prof. Dr. Eberhard Ulich (Projektleitung) +41-1-254-7070
Dr. Philipp Spinas (Ansprechpartner) +41-1-254-7077
Dipl.Inf.,Dipl.Psych. Matthias Rauterberg +41-1-254-7082
Dipl.Psych. Oliver Strohm +41-1-254-7084
lic.phil. Daniel Waeber +41-1-254-7077

ADI Software GmbH, Karlsruhe (BRD)



Dr. Karl Schlagenhaut (Projektleitung)
Raimund Mollenhauer (Ansprechpartner) +49-721-57000-66
Jürgen Dobrinski

ISBN 3-9065xx-yy

Inhaltsverzeichnis

Teil-I:.....	4
1 Einleitung	4
2 Allgemeine Definition von interaktiven Aufsetzpunkten	4
2.1 Interaktiver Aufsetzpunkt einer Kommando-Oberfläche	6
2.2 Interaktive Aufsetzpunkte einer Menü-Oberfläche	8
2.3 Interaktive Aufsetzpunkte einer Desktop-Oberfläche	11
3 Klassifikation von Benutzungsoberflächen	13
4 Zusammenfassung	14
5 Literaturverzeichnis.....	14
Teil-II:.....	16
1 Einleitung	16
2 Ein Konzept benutzer-orientierter Softwaregestaltung	16
2.1 Anforderungen an die Gestaltung interaktiver Software.....	19
2.2 Die Gestaltungsmatrix als Orientierungsrahmen.....	20
3 Ausgewählte quantitative Maße	22
3.1 I/O-Interface und repräsentationale Direktheit (RD).....	28
3.2 Dialogkomponente und interaktionale Direktheit (ID)	31
3.3 Anwendungskomponente und funktionale Direktheit (FD)	36
4 Zusammenfassung und Ausblick	40
5 Literaturverzeichnis.....	41

Vorwort

Der vorliegende BOSS-Bericht setzt sich aus zwei aufeinander aufbauenden Teilen zusammen. Der erste Teil befaßt sich mit dem Beschreibungsmittel des "interaktiven Aufsetzpunktes". Dieses Konzept wird definiert und an drei typischen Oberflächen-Beispielen erläutert. Dieses Beschreibungsmittel bildet eine der Grundlagen, welche für die Herleitung von quantitativen Maßen zur Operationalisierung arbeitspsychologischer Kriterien benötigt werden.

Im zweiten Teil dieses Berichtes werden für einige ausgewählte arbeitspsychologische Kriterien zur benutzer-orientierten Dialoggestaltung von ULICH quantitative Meßvorschriften vorgestellt. Zusätzlich zu dem Beschreibungsmittel des interaktiven Aufsetzpunktes, welches primär auf die Beschreibung der Ein/Ausgabe-Schnittstelle ausgerichtet ist, werden weitere Beschreibungsmittel zur Beschreibung der Dialogstruktur vorgestellt. Aufbauend auf diesen Beschreibungsmitteln werden verschiedene Meßvorschriften abgeleitet.

Teil-I:

Interaktive Aufsetzpunkte: ein Konzept zur Beschreibung und Klassifizierung von Benutzungsoberflächen interaktiver Software

1 Einleitung

Um die verschiedenen Typen von Benutzungsoberflächen beschreiben und klassifizieren zu können, bedarf es Beschreibungskonzepte, welche sich auf die verschiedenen Oberflächen gleichermaßen sinnvoll anwenden lassen (Rauterberg, 1989). Diese Beschreibungskonzepte müssen von den spezifischen Aspekten der einzelnen Oberflächentypen abstrahieren, um generell anwendbar zu sein. Im folgenden werden exemplarisch drei verschiedene Oberflächen vorgestellt und das Beschreibungskonzept des "interaktiven Aufsetzpunktes" an ihnen verdeutlicht.

Es wird weiterhin die Unterscheidung in *Wahrnehmungsbereich* und *Aktionsbereich* vorgenommen, um den Unterschieden zwischen einer traditionellen Menü-Oberfläche und einer Desktop-Oberfläche einfangen zu können. Zusätzlich wird die Dimension der "interaktionalen Direktheit" eingeführt, um dann zusammen mit der Dimension "Visualisierungsgrad", welche auf dem Konzept der interaktiven Aufsetzpunkte aufbaut, eine Klassifikation der unterschiedlichen Oberflächen vornehmen zu können.

Benutzungsoberflächen lassen sich gemäß der Beschreibung der Mensch-Computer-Schnittstelle nach dem IFIP-Modells mindestens in die folgenden Komponenten und Schnittstellen aufgliedern: den Benutzer, die Ein/Ausgabeschnittstelle (EAS), die Dialogkomponente (DK) und die Anwendungs-, bzw. Werkzeugkomponente (AK) (IFIP, 1981; Dzida, 1987).

2 Allgemeine Definition von interaktiven Aufsetzpunkten

Ein Benutzer interagiert mit dem System, indem er sich, an bestimmten Merkmalen auf der EAS orientierend, über die Dialog-Komponente mit der Anwendungs-Komponente Aufgaben, bzw. Probleme zu lösen versucht. Betrachtet man nun die AK als das "traditionelle" Werkzeug zur Erledigung der gestellten Aufgaben, so schiebt sich mit dieser neuen Technologie zwischen den Benutzer und das eigentliche Werkzeug ein neues, das "interaktive" Werkzeug in Form der DK (Spinas, 1987).

Als "Operationen" eines Benutzer werden die einzelnen Tastendrucke, bzw. Maus-Klicks, Joystick-Aktionen, etc. verstanden. Aus diesen einzelnen Operationen setzen sich dann die jeweiligen Operatoren (Open) zusammen. So setzt sich die Eingabe des Operators "dir" bei Verwendung der Kommando-Oberfläche MS-DOS aus der Tastenfolge "d", "i", "r" und "CR" (CR = carriage return) zusammen. Operatoren sind all diejenigen interaktiven Teilhandlungen eines Benutzers, welche von dem jeweiligen System akzeptiert und in die intern entsprechenden Funktionen umgesetzt werden. Diese internen Funktionen werden nun "funktionale interaktive Aufsetzpunkte" (FIAPe) genannt. So ist z.B. dem Operator, bzw. Kommando "dirCR" eine systeminterne Funktion, bzw. ein Algorithmus zugeordnet, welcher die aktuellen 'Directory'-Inhalte auf dem Bildschirm auflistet. Diese FIAPe sind dem Benutzer *nur indirekt* über die jeweilige Dialog-Syntax zugänglich. Die Menge der FIAPe umspannt die Semantik des interaktiven Systems. Wenn diesen einzelnen FIAPen eine wahrnehmbare Struktur auf dem Bildschirm zugeordnet ist, so nennen wir diese wahrnehmbaren Strukturen "repräsentationale interaktive Aufsetzpunkte" (RIAPe). Die *Klasse der Repräsentationsformen* setzt sich nun aus den folgenden zwei unterschiedlichen Mengen zusammen:

- *Die Menge aller aktiven Repräsentationsformen:*
Diese Menge umfaßt alle diejenigen interaktiven Aufsetzpunkte (IAPe), welche es dem Benutzer ermöglichen, entweder eine Dialogfunktion oder eine Anwendungsfunktion auszulösen. Hierzu zählen alle Tasten der Tastatur und gegebenenfalls alle maus-sensitiven Bereiche auf dem Bildschirm, bzw. alle zusätzlichen Eingabegeräte wie Digitizer, spezielle Funktionstasten-Tabletts, bei speziellen Tastaturen auch die LCD-Anzeigen auf den Funktionstasten, etc. Alle diese interaktiven Aufsetzpunkte umspannen den **Aktionsbereich** des Benutzers.
- *die Menge aller passiven Repräsentationsformen:*
Diese Menge beinhaltet alle auf dem Ausgabemedium (meistens ein oder zwei Bildschirme) seitens des Benutzers wahrnehmbaren Signalmuster (einschliesslich der auditiven Signale). Diese Repräsentationsformen heißen passiv, weil sie ausschließlich der Rückmeldung über system-interne Zustandsparameter dienen und nicht zum Aktionsbereich des Benutzers gehören. Alle diese passiven Repräsentationsformen sind Bestandteil des **Wahrnehmungsbereiches**.

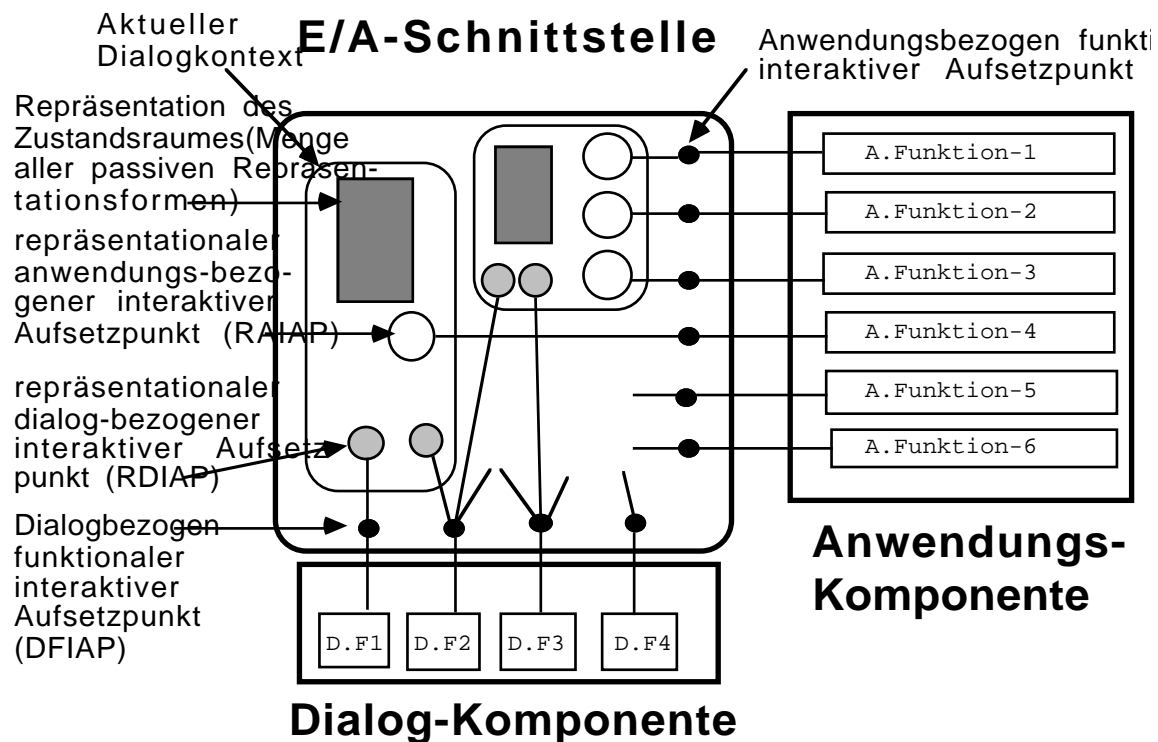


Abb. 1. Schematische Aufteilung der Benutzungsoberfläche in die dialog- und die anwendungsbezogenen interaktiven Aufsetzpunkte. Entsprechend der Aufteilung in Dialog- und Anwendungskomponente werden auch die funktionalen interaktiven Aufsetzpunkte (FIAPe) in dialog- und anwendungsbezogene funktionale interaktive Aufsetzpunkte (DFIAPe, bzw. AFIAPe) unterschieden. Wenn diese beiden Sorten von FIAPen noch auf der Ein/ Ausgabe-schnittstelle semantisch eindeutige, wahrnehmbare Repräsentationsformen haben, dann werden sie repräsentationale dialog-bezogene, bzw. anwendungs-bezogene IAPe genannt (RDIAP, bzw. RAIAP).

Die Menge der aktiven Repräsentationsformen läßt sich weiterhin in zwei Bereiche unterteilen: 1.) die Menge der Operatoren, welche sich auf die Handhabung von Dialogobjekten beziehen (z.B. Fenster; Menü-Items, sofern sie zur reinen Dialogsteuerung dienen) und 2.) diejenige Menge der Operatoren, welche sich auf die Menge der Anwendungsobjekte beziehen (z.B. Textdokumente, Absätze, Zeilen, Zeichen, etc.). Wenn also eine reine Dialogfunktion vorliegt, so nennen wir diesen FIAP dialog-bezogenen funktionalen interaktiven Aufsetzpunkt (DFIAP). Demgegenüber werden alle FIAPe, welche sich auf die Veränderung von Eigenschaften der Anwendungsobjekte mit Hilfe der Anwendungsfunktionen beziehen,

anwendungs-bezogene funktionale interaktive Aufsetzpunkte (AFIAPe) genannt (siehe Abb. 1).

Die EAS läßt sich hinsichtlich der dort vorhandenen Repräsentationsformen in die den DFIAP, bzw. AFIAP zugeordneten repräsentationalen interaktiven Aufsetzpunkten unterscheiden (RDIAP, bzw. RAIAP; siehe Abb. 1). Die Dimension der interaktionalen Direktheit hat nun zum Ziel, möglichst viele der AFIAPe bei möglichst wenigen DFIAPen dem Benutzer in dem aktuellen, d.h. in dem anwendungsobjekt-bezogenen Dialogkontext *direkt* zur Verfügung zu stellen. Die Menge der RDIAPe und der RAIAPe werden unter der Menge aller RIAPe zusammengefaßt.

Formal lassen sich die verschiedenen Mengen an IAPen wie folgt beschreiben:

IAP	$:= \{r : r \in \text{RIAP}\} \cup \{f \mid f \in \text{FIAP}\}$
FIAP	$:= \{d : d \in \text{DFIAP}\} \cup \{a \mid a \in \text{AFIAP}\}$
RIAP	$:= \{r : r \in \text{RDIAP}\} \cup \{r \mid r \in \text{RAIAP}\}$
RDIAP	$:= \{(d,r) \in \{\text{DFIAP}\} \times \{\text{RD}\} : r = \delta(d)\}$
RAIAP	$:= \{(a,r) \in \{\text{AFIAP}\} \times \{\text{RA}\} : r = \alpha(a)\}$
RD	$:= \{\text{alle geeigneten Repräsentationsformen für DIAPe}\}$
RA	$:= \{\text{alle geeigneten Repräsentationsformen für AIAPe}\}$

2.1 Interaktiver Aufsetzpunkt einer Kommando-Oberfläche

Als erstes Beispiel soll die Kommando-Oberfläche von MS-DOS dazu dienen, das Beschreibungskonzept der interaktiven Aufsetzpunkte zu verdeutlichen. Zunächst wählen wir verschiedene dialog- und anwendungs-bezogene Funktionen aus: die Kommandos "dirCR" und "cd <path>CR" sind stellvertretend für alle dialog-bezogenen funktionalen interaktiven Aufsetzpunkte, und die beiden Kommandos "delete <file name>CR" und "rename <old name> <new name>CR" sind stellvertretend für alle anwendungs-bezogenen funktionalen interaktiven Aufsetzpunkte.

Dem Benutzer steht in der Regel bei einer Kommando-Oberfläche nur ein repräsentationaler interaktiver Aufsetzpunkt (RIAP) zur Verfügung (Abb. 2; der Eingabebereich hinter dem "System-Prompt" auf dem Bildschirm). Wenn bestimmte Funktionen auch über spezielle Funktionstasten oder Tastenkombinationen ausgelöst werden können, so sind ihre Repräsentationen auf der Tastatur ebenfalls RIAPe (Abb. 2; die Kreise über den Tasten der Tastatur). Es besteht zwischen dem RIAP auf dem Bildschirm und den AFIAPen, bzw. DFIAPen eine (1:n)-Beziehung. Dies läßt sich abstrakter durch die schematische Darstellung in Abbildung 3 veranschaulichen.

An einem einfachen Beispiel läßt sich die Menge der DFIAPe verdeutlichen: ein Teil der Objekte der AK von MS-DOS sind die auf der Festplatte, bzw. Diskette verwalteten Dateien; diese Anwendungs-Objekte lassen sich über Eigenschaften wie "Name.Extension", "Größe in Byte", "Datum der Erstellung", "Datum der letzten Änderung", "Ort der Speicherung", etc. beschreiben, bzw. auf der EAS repräsentieren und gegebenenfalls verändern; alle diese Eigenschaften sind jedoch nur potentiell wahrnehmbar; erst durch den DFIAP "dirCR" lassen sich diese Eigenschaften in eine passive Repräsentationsform überführen; andere Oberflächen für MS-DOS wie zum Beispiel PCTOOLS, QDOS, GEM oder MS-WINDOWS haben aus diesen *potentiell* wahrnehmbaren Eigenschaften *aktuell* wahrnehmbare Eigenschaften werden lassen, indem jeweils automatisch der aktuelle 'Directory'-Inhalt auf der EAS repräsentiert wird. Wie unbedingt notwendig dies ist, belegen die empirischen Untersuchungen von Greenberg und Witten (1988). Das mit Abstand am häufigsten benutzte Kommando bei der Benutzung von UNIX ist das "ls" Kommando, welches wie das "dir" Kommando bei MS-DOS lediglich den aktuellen 'Directory'-Inhalt auf den Bildschirm ausgibt und somit eindeutig der Orientierung dient (Kraut et al., 1983). Es zeigte sich, daß eine der Hauptschwächen von Kommandooberflächen in der mangelnden Rückmeldung zu sehen ist. Diese Schwäche erweist sich sowohl für Anfänger, als auch für erfahrene Benutzer nachteilig (Kraut et al., 1983; Greenberg & Witten, 1988).

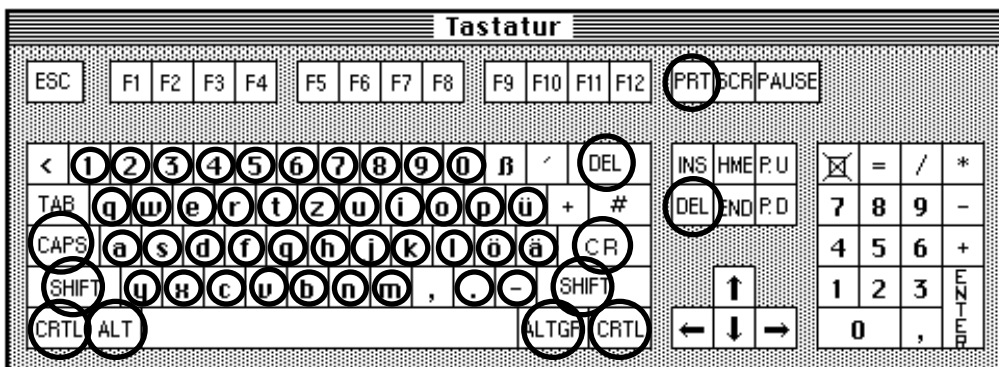
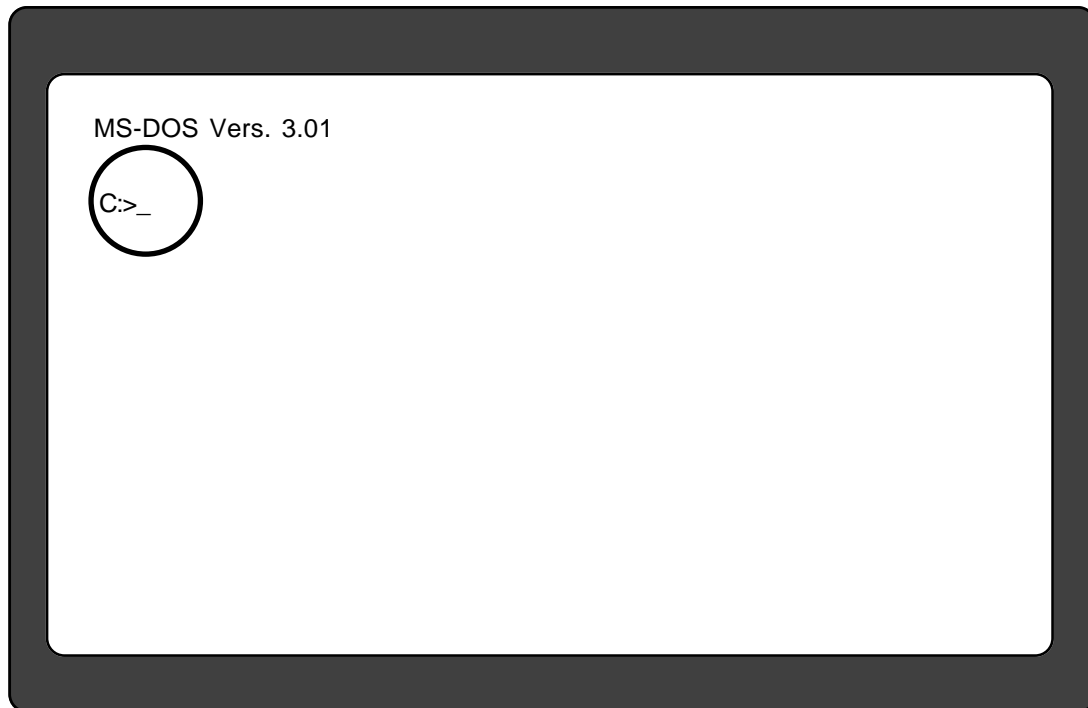
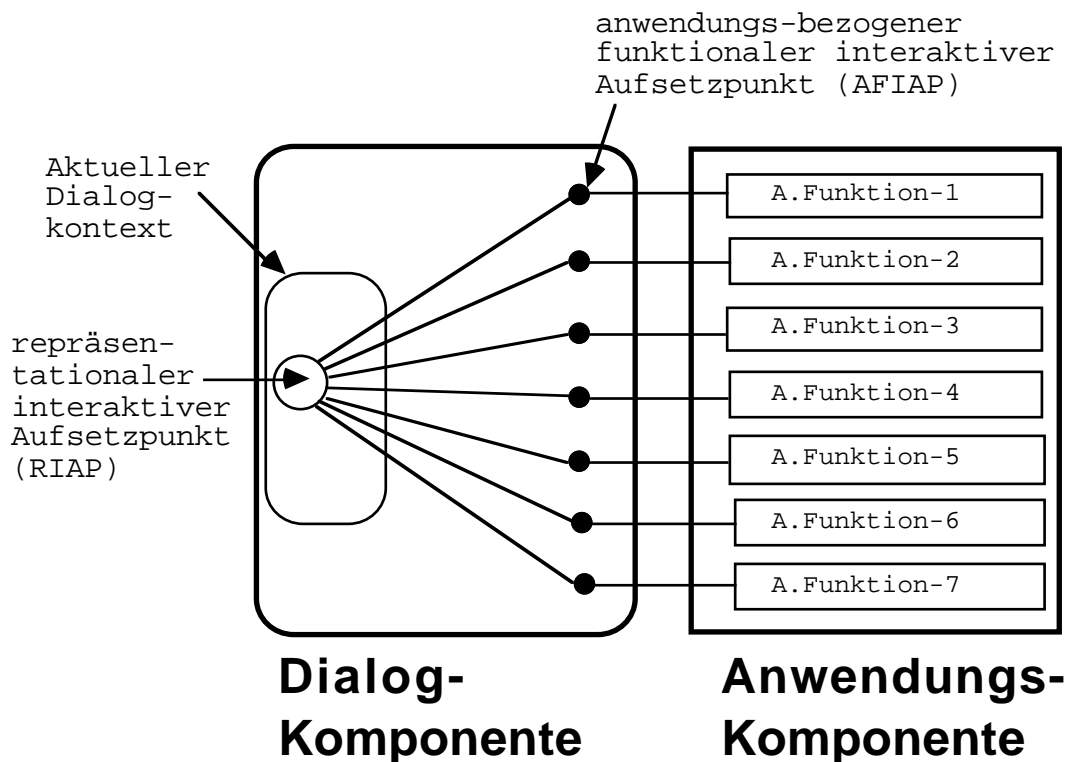


Abb. 2 Jede Kommando-Oberfläche (hier z.B. MS-DOS) zeichnet sich dadurch aus, daß auf dem Bildschirm nur ein repräsentationaler interaktiver Aufsetzpunkt (RIAP) dem Benutzer zur Verfügung steht (der Eingabebereich hinter dem "System-Prompt"). Weitere RIAPe sind eventuell über spezielle Funktionstasten (z.B. "PRT"), bzw. Tastenkombinationen (z.B. "ALT CTRL DEL") gegeben. Der Aktionsbereich des Benutzers ist im wesentlichen die Tastatur. Der Bildschirm dient lediglich als Ausgabemedium.



Kommando-orientierte Benutzung

Abb. 3 Die schematische Darstellung der (1:n)-Beziehung zwischen dem einen repräsentationalen interaktiven Aufsetzpunkt des Bildschirms und den n anwendungs-bezogenen funktionalen interaktiven Aufsetzpunkte (AFIAPe). Aus Gründen der Übersichtlichkeit wurden die dialog-bezogenen funktionalen interaktiven Aufsetzpunkte (DFIAPe) fortgelassen.

2.2 Interaktive Aufsetzpunkte einer Menü-Oberfläche

Sehen wir uns als zweites Beispiel die historisch nächste Entwicklungsstufe von Benutzungsoberflächen an. Eine Menü-Oberfläche zeichnet sich im wesentlichen dadurch aus, daß möglichst für jeden funktionalen interaktiven Aufsetzpunkt (FIAP) auch ein repräsentationaler interaktiver Aufsetzpunkt (RIAP) vorhanden ist und somit dem Benutzer *Feedback* über die im aktuellen Dialogkontext gültigen Operatoren gibt (Ulich et al., 1991).

Da jedoch die Ein/Ausgabeschnittstelle (EAS) nur einen physikalisch begrenzten Darstellungsraum hat, können auf der Bildschirmoberfläche nur ausgewählte FIAPe repräsentiert werden (Ilg & Ziegler, 1987). Dies hat zur Konsequenz, daß die RIAPe auf verschiedene Dialogkontexte aufgeteilt werden müssen. Übertragen wir diese Sicht auf unser schematisches Diagramm, so ergibt sich Abbildung 5.

Ein besonderes Problem bleibt jedoch noch - insbesondere im Vergleich zu direkt-manipulativen Oberflächen - bestehen: welchen repräsentationalen Stellenwert haben die Hinweise auf die aktuelle Semantik der Funktionstasten am unteren Bildschirmrand (siehe Abb. 4) ? Zur Beantwortung dieser Frage ist die oben eingeführte Unterscheidung in den Wahrnehmungsbereich und den Aktionsbereich hilfreich. Wir nennen diese semantischen 'Labels' der Funktionstasten "repräsentationale interaktive Aufsetzpunkte des Wahrnehmungsbereiches" (WRIAPe) und die RIAPe der Funktionstasten auf der Tastatur "repräsentationale interaktive Aufsetzpunkte des Aktionsbereiches" (ARIAPe).

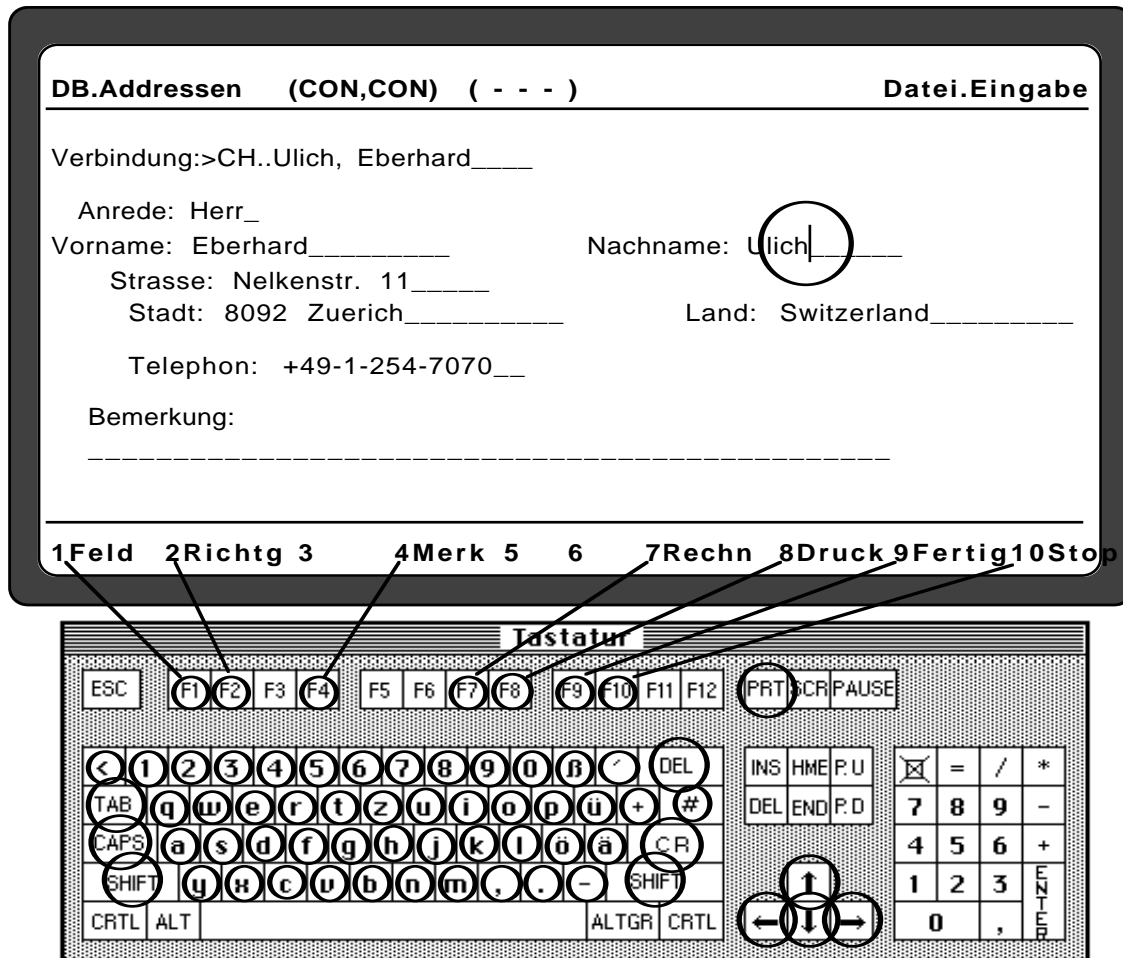
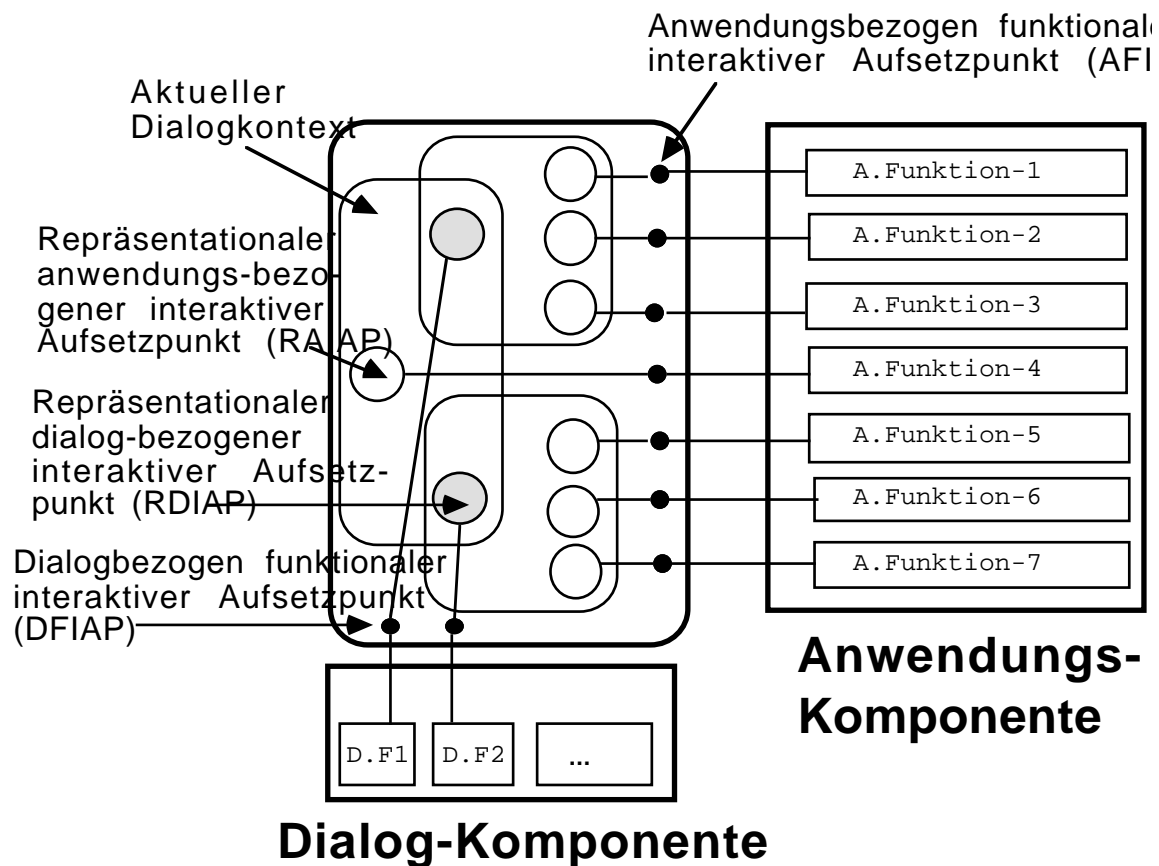


Abb. 4 Die Menü-Oberfläche des Datenbankprogrammes ADIMENS. Man befindet sich in der Routine "Eingabe" des Moduls "Daten". Der Benutzer hat die Möglichkeit, bei dem "Eingabe"-RIAP (senkrechter Strich) weitere ASCII-Zeichen einzugeben, bzw. zu löschen. Die Verlagerung dieses RIAP erfolgt über die Cursor-Tasten. Die weitere Dialogsteuerung kann nur noch über die Funktionstasten erfolgen. Die Semantik der Funktionstasten ist in der untersten Bildschirmzeile gegeben.

Dieser Aspekt ist immer dann von besonderer Bedeutung, wenn der für einen FIAP die Semantik tragende perzeptuelle RIAP räumlich nicht mit dem aktionalen RIAP zusammenfällt. Dies ist z.B. dann der Fall, wenn auf dem Bildschirm die Bedeutung (in irgendeiner Form) für einzelne Tasten (insbesondere Funktionstasten) oder sonstige Interaktionselemente als RIAP gegeben ist. Wir müssen also zwischen der Menge der *RIAPe des Wahrnehmungsbereiches* (WRIAPe) und der Menge der *RIAPe des Aktionsbereiches* (ARIAPe) unterscheiden. Diese Unterscheidung ist deshalb wichtig, weil hier die eigentlich zusammengehörenden Elemente des Wahrnehmungsbereiches und des Aktionsbereiches auseinanderfallen. Diese Distanz ist z.B. durch die physikalische Entfernung Δ zwischen WRIAP und ARIAP bestimmbar (Abb. 6). Um dieses Problem teilweise zu entschärfen, werden bei überwiegend durch Funktionstasten gesteuerte Systeme *Tastatur-Schablonen* mit ausgeliefert.



Menü-orientierte Benutzung

Abb. 5 Die schematische Darstellung einer Menü-Oberfläche. In dem aktuellen Dialogkontext sind zwei dialog-bezogene und ein anwendungs-bezogener interaktiver Aufsetzpunkt repräsentiert. Alle IAPe, welche nur zum Wechsel des Dialogkontextes dienen, sind funktional dialog-bezogen.

Die Zuordnung zwischen WRIAP und ARIAP muß vom Benutzer entweder im (internen) Kurzzeit- oder bei längerer Übung im Umgang mit der jeweiligen Software auch im Langzeitgedächtnis für die handlungleitende Entscheidung verfügbar sein. Eine andere Möglichkeit besteht darin, daß der Benutzer diese Zuordnung im externen visuellen "Gedächtnis" (die EAS) beläßt und die jeweilige Funktionstaste ohne Blickwendung hin zur Tastatur findet und betätigt. Je größer nun die physikalische Distanz zwischen WRIAP und ARIAP ist, desto stärker ist - zumindest für den ungeübten Benutzer - die Gedächtnisbelastung.

Bei direktmanipulativen Oberflächen wird (gegenüber den Menü-Oberflächen) für die maus-sensitiven Bereiche (bzw. die entsprechenden RIAPe) die Distanz Δ Null sein. Dies kann deshalb als sinnvoll angenommen werden, weil bei dem Interaktionselement "Maus" der entsprechende ARIAP nicht die Maus-Taste selbst, sondern der Maus-Cursor auf dem Bildschirm ist. Dies gilt dann auch ebenso für die Cursor-Tasten der herkömmlichen Tastatur, bzw. für alle Arten der Interaktionssteuerung, bei der ein ARIAP des jeweiligen Aktionsbereiches auf dem Bildschirm vorhanden ist. Hierbei wird angenommen, daß die jeweilige Positionierung, bzw. Steuerung des ARIAPes auf dem Bildschirm auf der sensorischen Regulationsebene ausgeführt werden kann, ohne dabei mit höheren Regulationsebenen zu interagieren. Der Nachteil der Cursor-Steuertasten liegt darin begründet, daß sie einen erhöhten interaktiven Aufwand benötigen (Ilg & Ziegler, 1987; Hächler et al., 1990).

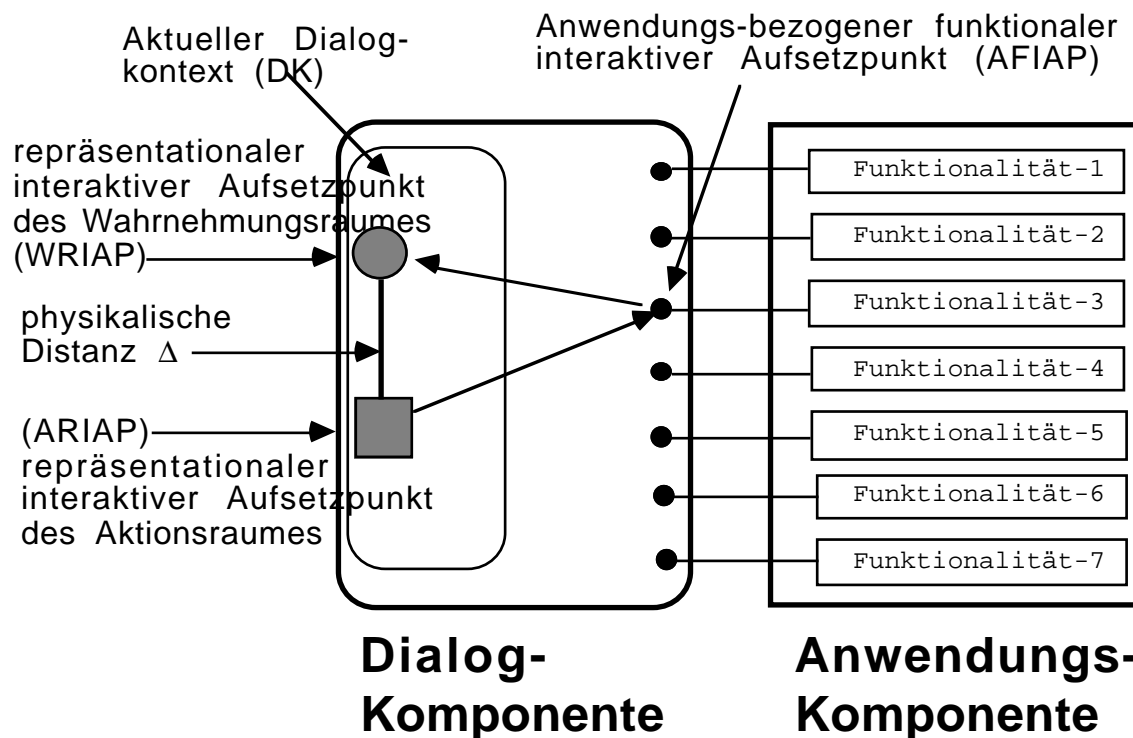


Abb. 6. Die Unterscheidung der Menge der repräsentationalen interaktiven Aufsetzpunkte (RIAPe) in die Menge der Elemente des Wahrnehmungsbereiches (WRIAPe) und die Menge der Elemente des Aktionsbereiches (ARIAPe). Dieser Unterschied ist immer dann wichtig, wenn der Wahrnehmungsbereich und der Aktionsbereich auseinanderfallen.

2.3 Interaktive Aufsetzpunkte einer Desktop-Oberfläche

Kommen wir nun zur modernsten Art von Benutzungsoberflächen: den direkt-manipulativen, bzw. objekt-orientierten Oberflächen (DMO). Eine Übergangsform zwischen den traditionellen Menü-Oberflächen und den DMOs bilden die Desktop-Oberflächen. Eine Desktop-Oberfläche zeichnet sich neben der grafikfähigen Bildschirmausgabe sowohl durch 'Pull-down'-Menüs, als auch durch Piktogramme, Fenster, Dialog-Boxen, Dialog-Knöpfe, etc. aus. Durch diese neuen Dialog-Objekte ergeben sich auch neue Dialog-Operatoren. Das folgende Zitat verdeutlicht am Beispiel eines "Fensters" die Menge der aktiven und passiven Repräsentationsformen dieses Dialogobjektes (siehe auch Abb. 8).

Ein einzelnes Fenster benötigt eine Menge von Zustandsvariablen (*passive Komponente*), die seine verschiedenen Parameter beschreiben (z.B. Position und Größe auf dem Bildschirm, Verweis auf den Fensterinhalt). Zusätzlich gibt es noch eine Menge von Operationen (*aktive Komponente*), die auf ein Fenster angewendet werden können (Fabian, 1986, S. 113).

Bei einer DMO werden die meisten FIAPe dem Benutzer direkt auf einem globalen Dialogzustand in den verschiedensten Repräsentationsformen angeboten. Da bei einem einigermaßen komplexen System dennoch nicht alle FIAPe als RIAPe gleichzeitig auf dem Bildschirm Platz haben (Ilg & Ziegler, 1987), werden sie als Menü-Items z.B. in den Pull-down Menü der Menü-Leiste versteckt (Fabian & Rathke, 1983). Wie man sehen kann (Abb. 7) nimmt die Bedeutung der Tastatur bei der konkreten Dialogsteuerung stark ab. Die Dialogsteuerung erfolgt fast ausschließlich über Interaktionselemente wie Maus, Rollkugel, Finger (bei berührungssensitiven Bildschirmen), bis hin zu dem Datenhandschuh (Foley, 1988). Der Vorteil dieser neuen Interaktionsarten besteht darin, daß Aktionsbereich und Wahrnehmungsbereich weitgehend zusammenfallen (Dumais & Jones, 1985). Ebenso weisen neuere Entwicklungen im Schnittstellenbereich in diese Richtung (Ward & Phillips, 1987; Mel et al. 1988). Eine Reihe von speziellen Interaktionselementen werden nur dazu entwickelt, um dem Benutzer möglichst direkt die benötigte Anwendungsfunktion darzubieten zu können. Dies kann man z.B. bei CAD-Arbeitsplätzen beobachten.

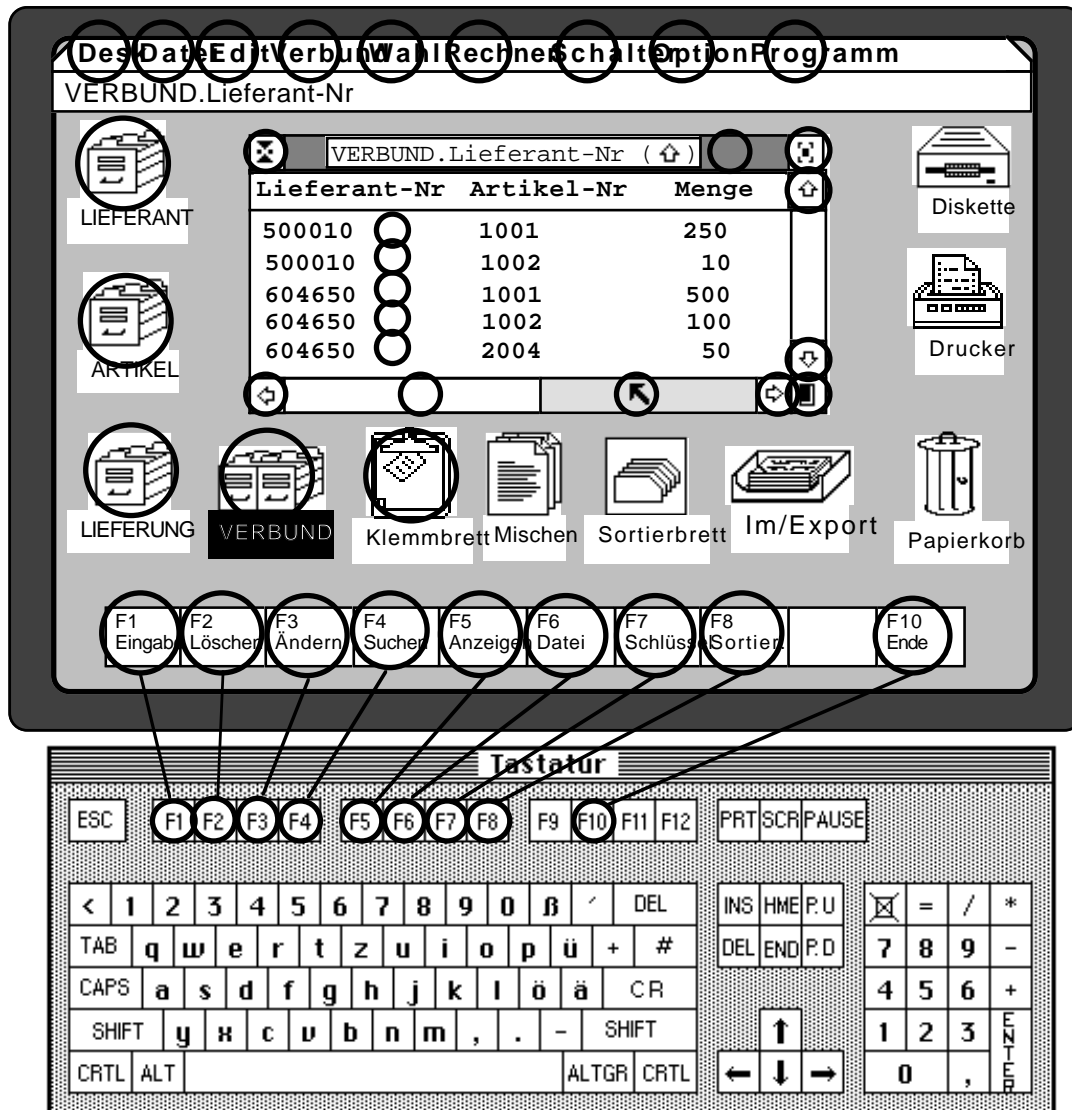


Abb. 7 Alle die eingekreisten Repräsentationsformen (RIAPe) gehören zum Aktionsbereich des Benutzers dieser exemplarisch ausgewählten Desktop-Oberfläche. Bis auf die fünf RIAPe bei den Datensätzen im Fenster und bis auf F6, F7, F8 dienen alle anderen RIAPe der weiteren Dialogsteuerung. Man beachte, daß bei dieser Desktop-Oberfläche die semantischen 'Label' der einzelnen Funktionstasten selbst wiederum (als maus-sensitive Bereiche) RDIAPe darstellen. Der Maus-Cursor als repräsentativer interaktiver Aufsetzpunkt des Aktionsbereiches (ARIAP) befindet in einem der maus-sensitiven Bereiche des vertikalen Rollbalkens im Fenster.

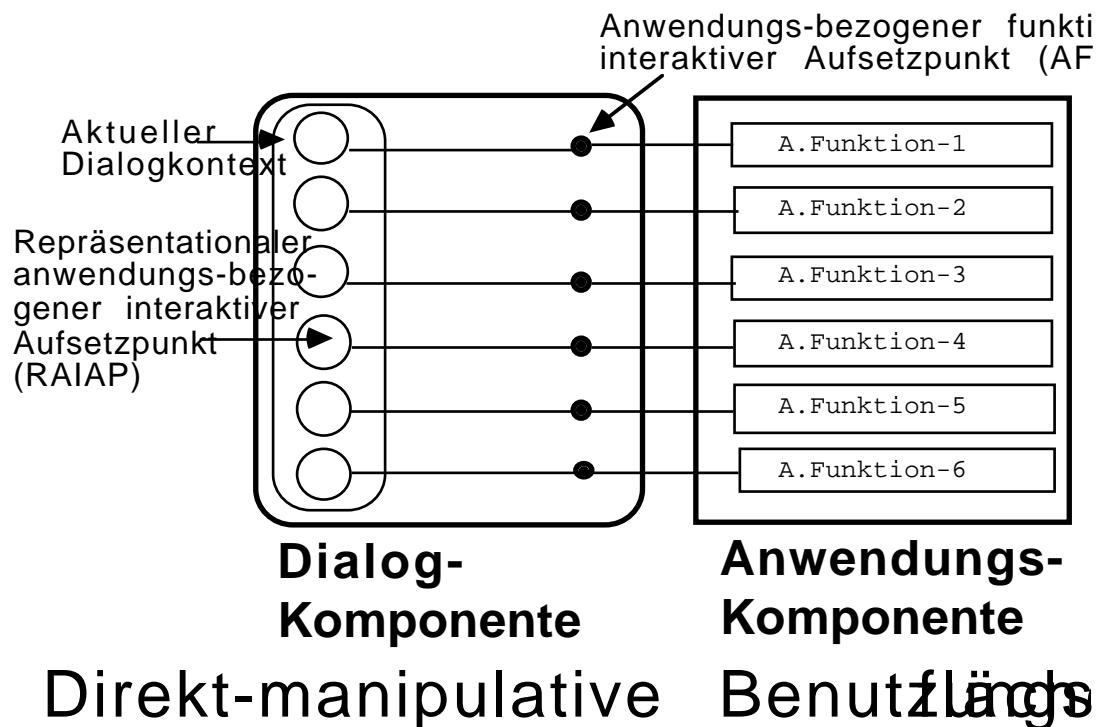


Abb. 8 Die schematische Darstellung einer direktmanipulativen-Oberfläche (DMO). In dem aktuellen Dialogkontext sind nur noch anwendungs-bezogene interaktive Aufsetzpunkte repräsentiert (RAIAPe). Der Benutzer hat aus dem aktuellen Dialogkontext heraus *direkten* Zugriff über die vorhandenen Repräsentationen auf alle Funktionen der Anwendungs-komponente.

Das bisher vorgestellte Beschreibungs-konstrukt des interaktiven Aufsetzpunktes gibt natürlich nur den *rein syntaktischen Charakter* der repräsentationalen Formen wieder und läßt die Problematik der semantischen Bedeutung der einzelnen RIAPe zunächst außer acht (siehe in diesem Zusammenhang die Ergebnisse von STREITZ, LIESER & WOLTERS; 1989). Ein wesentlicher Vorteil des Konzeptes der IAPe liegt darin, daß man den interaktiven Aufwand beschreiben kann. Wieviele verschiedene RIAPe müssen z.B. benutzt werden, um an einen bestimmten FIAP heranzukommen. Dieser Aspekt wird "interaktionale Direktheit" genannt.

Übertragen wir die bisherigen Aspekte auf unsere schematische Darstellung, so gelangen wir zu Abbildung 8. Der interaktive Overhead, welcher durch das Navigieren durch komplexe Menü-Strukturen entsteht, verschwindet. Hier weist auch die Technologie von Hyper-Text-Systemen auf neue Arten der Interaktion *direkt* auf den Anwendungsobjekten hin. Einfachste Dialogoperatoren (Maus-Klick, etc.) erlauben es dem Benutzer, unmittelbar auf dem Anwendungsobjekt zu navigieren und zu operieren. Die Funktionalität der Dialogkomponente wird somit auf ein Minimum reduziert.

3 Klassifikation von Benutzungsoberflächen

Um nun die hier vorgestellten Oberflächen klassifizieren zu können, werden die beiden Dimension "Visualisierungsgrad" und "interaktionale Direktheit" herangezogen. Den Visualisierungsgrad kann man über Verhältnis der repräsentationalen zu den funktionalen interaktiven Aufsetzpunkten bestimmen. Wenn für alle vom interaktiven System angebotenen Funktionen (dialog- und anwendungs-bezogen) eine spezifische Repräsentationsform existiert, so ist der Visualisierungsgrad 100%.

Je weniger überflüssige Dialogschritte notwendig sind, um an eine gerade (bezogen auf den aktuellen Dialogkontext) gesuchte Anwendungs- oder Dialogfunktion heranzukommen, desto interaktiv direkter ist die Oberfläche (Laverson, Norman & Shneiderman, 1987).

Wenden wir diese beiden Dimensionen auf unsere drei Oberflächen-Typen an, so zeigt sich, daß die Kommando-Oberfläche zwar genauso interaktiv direkt (wenn nicht oftmals sogar direkter) ist wie eine Desktop-Oberfläche, aber nur einen sehr niedrigen Visualisierungsgrad hat.

		Visualisierungsgrad	
		niedrig	hoch
Interaktionale Direktheit	niedrig	(Batch-processing)	Menue-Oberfläche (z.B. MsWORD für P
	hoch	Kommando Oberfläche (z.B. MS-DOS, UNIX)	Direkte Manipulation (z.B. Desktop-Oberfläc

Abb. 9 Klassifikationsschema für die verschiedenen Oberflächen mit den beiden Dimensionen "Visualisierungsgrad" und "interaktionale Direktheit". Wie man sehen kann, ist die Kommando-Oberfläche ähnlich interaktiv direkt wie eine Desktop-Oberfläche, hat jedoch einen außerordentlich geringen Visualisierungsgrad.

4 Zusammenfassung

Wir haben das Beschreibungskonzept der interaktiven Aufsetzpunkte vorgestellt und es mit den Komponenten eines interaktiven Systems ("IFIP-Modell") in Beziehung gesetzt. Hierdurch liessen sich Oberflächen bezogene Charakteristiken festlegen. Diese Charakteristiken lassen sich den Klassifizierungsdimensionen "Visualisierungsgrad" und "interaktionale Direktheit" zuordnen. Um die verschiedenen Typen von Benutzungsoberflächen adäquat mit einheitlichen Begriffen beschreiben und damit dann auch vergleichen zu können, wurden verschiedene Beschreibungskonstrukte für interaktive Aufsetzpunkte eingeführt und an konkreten Beispielen erläutert. Mit Hilfe der beiden Beschreibungsdimensionen "Visualisierungsgrad" und "interaktionale Direktheit" lassen sich die drei Oberflächen "Kommando", "Menü" und "direkte Manipulation" in einem Vierfelderschema sinnvoll einordnen.

5 Literaturverzeichnis

- DUMAIS S T, JONES W P (1985) A Comparison of Symbolic and Spatial Filing. In: BORMAN L, CURTIS B (eds.) Human Factors in Computing Systems-II. Elsevier (North-Holland), Amsterdam, pp.127-130
- DZIDA W (1987) On tools and interfaces. In: FRESE M, ULICH E, DZIDA W (eds.) Psychological Issues of Human Computer Interaction in the Work Place. North-Holland, Amsterdam, pp 339-355
- FABIAN F (1986) Fenster- und Menüsysteme in der MCK. In: FISCHER G, GUNZENHÄUSER R (Hrsg.) Methoden und Werkzeuge zur Gestaltung benutzergerechter Computersysteme. Walter de Gruyter, Berlin New York, S 101-119
- FABIAN F, RATHKE C (1983) Menüs: Einsatzmöglichkeiten eines Fenstersystems zur Unterstützung der Mensch-Maschine-Kommunikation. Office Management, 42-44
- FOLEY J D (1988) Neuartige Schnittstellen zwischen Mensch und Computer. Spektrum der Wissenschaft, Sonderheft Nr. 6 "Die nächste Computer-Revolution", 66-69

- GREENBERG S, WITTEN I H (1988) Directing The User Interface: How People Use Command-Based Computer Systems. Preprints of the IFAC/IFIP/IEA/IFORS Conference on Man-Machine Systems: Analysis, Design and Evaluation, vol. II, 299-305
- HÄCHLER G, JULEN M, KELLER R, STOFFEL M (1990) Arbeitspsychologische Analyse moderner Benutzungsoberflächen der MCI. unveröffentlichte Semesterarbeit, Institut für Arbeitspsychologie, ETH-Zürich
- IFIP (1981) Report of the 1st Meeting of the European User Environment Subgroup of IFIP WF 6.5. Gesellschaft für Mathematik und Datenverarbeitung, Bonn
- ILG R, ZIEGLER J (1987) Interaktionstechniken. In: FÄHNRIK K-P (Hrsg.) Software-Ergonomie. Oldenbourg, München Wien (State of the Art, No. 5, S. 106-117)
- KRAUT R E, HANSON S J, FARBER J M (1983) Command Usage and Interface Design. In: JANDA A (ed.) Proceedings of the CHI'83 Conference "Human Factors in computing Systems". Elsevier (North-Holland), Amsterdam New York, pp. 120-124
- LAVERSON A, NORMAN K, SHNEIDERMAN B (1987) An Evaluation of Jump-ahead Techniques in Menu Selection. Behaviour and Information Technology 6(2):97-108
- MEL B W, OMOHUNDRO S M, ROBINSON A D, SKIENA S S, THEARLING K H, YOUNG L T, WOLFRAM S (1988) Tablet: Personal Computer in the Year 2000. Communications of the ACM 31(6):639-646
- RAUTERBERG M (1989) Maus versus Funktionstaste: ein empirischer Vergleich einer desktop- mit einer ascii-orientierten Benutzungsoberfläche. In: MAASS S, OBERQUELLE H (Hrsg.) Software-Ergonomie '89. Teubner, Stuttgart (German Chapter of the ACM Berichte, Bd 32, S 313-323)
- SPINAS P (1987) Arbeitspsychologische Aspekte der Benutzerfreundlichkeit von Bildschirmssystemen. Psycholog. Dissertation, Universität Bern
- STREITZ N, LIESER A, WOLTERS A (1989) The combined effects of metaphor worlds and dialogue modes in human-computer-interaction. In: KLIX F, STREITZ N, WAERN Y, WANDKE H (eds.) Man-Computer Interaction Research MACINTER-II. Elsevier (North-Holland), Amsterdam, pp 75-88
- WARD J R, PHILLIPS M J (1987) Digitizer Technology: Performance Characteristics and the Effects on the User Interface. IEEE Computer Graphics and Applications 4:31-44
- ULICH E, RAUTERBERG M, MOLL T, GREUTMANN T, STROHM O (1991) Task Orientation and User-Oriented Dialog Design. International Journal of Human Computer Interaction (in press)

Teil-II: Ein quantitatives Bewertungs- und Gestaltungsschema für Benutzungsoberflächen von interaktiven Systemen

1 Einleitung

Es wurden in den letzten Jahren eine Reihe von verschiedenen Krieriensammlungen, Gestaltungskonzepten, Richtlinien, etc. aufgestellt, welche dazu gedacht sind, dem Software-Entwickler eine brauchbare und umsetzbare Handlungsanweisung zur benutzergerechten Systemgestaltung an die Hand zu geben.

Für die Einschränkung der Anwendbarkeit von Gestaltungskriterien lassen sich die folgenden drei Gründe ausmachen (für eine ausführlichere Diskussion siehe SMITH, 1986):

1. Es hat sich bisher kein allgemein verbindlicher standardisierungsfähiger Ansatz für die Gestaltung der anwendungsbezogenen Aspekte interaktiver Software herausgebildet.
2. Es hat sich in der Anwendungspraxis der Gestaltungskriterien gezeigt, daß für die Gestaltung einer Vielzahl interaktiver Systeme der Software-Entwickler um das anwendungsspezifische Wissen des Benutzer nicht umhinkommt. Die Anwendbarkeit der Gestaltungskriterien wird daher ergänzt und zum Teil überlagert von den Ansätzen im Bereich des partizipativen Software-Engineerings, bei denen Methoden zum Benutzereinbezug in den Erstellungsprozeß interaktiver EDV-Werkzeuge entwickelt und untersucht werden (SPINAS, WAEBER & STROHM; 1990).
3. Die Gestaltungskriterien sind hinsichtlich ihrer konkreten Umsetzbarkeit scheinbar in dem folgenden Dilemma: wenn das jeweilige Gestaltungskriterium allgemeingültigen Charakter aufweist, so ist es nicht spezifisch genug, um von dem Softwareentwickler auf sein konkretes Gestaltungsproblem angewendet werden zu können; ist das Gestaltungskriterium jedoch andererseits konkret umsetzbar, so ist der Anwendungskontext dieses Kriteriums oft nicht passend oder unbekannt. "However, the more specific a guideline is, the less generally applicable it will be" (MOSIER & SMITH; 1986, S.43).

Aufgrund des letzten Aspektes soll versucht werden, ein möglichst umfassendes, benutzerorientiertes Gestaltungskonzept so zu entwickeln, daß es sich durch eine Ergänzung entsprechend operationalisierter Transformationsregeln auf das jeweilige konkrete Gestaltungsproblem im Software-Entwicklungsprozeß handlungsleitend anwenden läßt.

Im weiteren soll nun eine detaillierte Sicht von Benutzungsoberflächen interaktiver Software vorgestellt werden, anhand derer sich dann die Stoßrichtung der Transformationsregeln auf der Grundlage des benutzerorientierten Gestaltungskonzeptes von ULLICH (1988, 1989, 1991) aufzeigen läßt.

2 Ein Konzept benutzer-orientierter Softwaregestaltung

Die Benutzungsoberfläche läßt sich durch die Beschreibung der Mensch-Computer-Schnittstelle des IFIP-Modells mindestens in die folgenden Komponenten und Schnittstellen aufgliedern: den Benutzer, die Ein/Ausgabeschnittstelle¹ (EAS), die Dialogkomponente² (DK) und die Anwendungs-, bzw. Werkzeugkomponente³ (AK) (IFIP, 1981; DZIDA, 1987).

¹ EAS = Ein/Ausgabeschnittstelle

² DK = Dialogkomponente

³ AK = Anwendungs-, bzw. Werkzeugkomponente

Ein Benutzer interagiert mit dem System, indem er sich an bestimmten Merkmalen auf der Ein/Ausgabeschnittstelle orientierend über die Dialogkomponente mit der Anwendungskomponente seine Aufgaben, bzw. Probleme zu lösen versucht. Betrachtet man nun die Anwendungskomponente als das "traditionelle" Werkzeug zur Erledigung der gestellten Aufgaben, so schiebt sich mit dieser neuen Technologie zwischen den Benutzer und das eigentliche Werkzeug ein neues Werkzeug, das "interaktive" Werkzeug in Form der Dialogkomponente.

Als "Operationen" eines Benutzer werden die einzelnen Tastendrucke, bzw. Maus-Klicks, Joystick-Aktionen, etc. verstanden. Aus diesen einzelnen Operationen setzen sich dann die jeweiligen Operatoren⁴ (Open) zusammen. So setzt sich die Eingabe des Operators "dir" bei Verwendung der Kommando-Oberfläche MS-DOS aus der Tastenfolge "d", "i", "r" und "CR" zusammen. Diese Beschreibungsebene der einzelnen Operationen stellt den höchsten Auflösungsgrad einer externen Beobachtung dar und läßt sich software-technisch durch "Logfile"-Aufzeichnung realisieren.

Operatoren (Open) sind all diejenigen interaktiven Teilhandlungen eines Benutzers, welche von dem jeweiligen System akzeptiert und z.T. in die intern entsprechenden Funktionen umgesetzt werden. Diese Operatoren werden auf Dialog-Objekte⁵ (DObe) angewendet.

Die *Klasse der Dialogobjekte* setzt sich nun aus den folgenden zwei unterschiedlichen Mengen zusammen:

- *Die Menge aller aktiven Repräsentationsformen:*

Diese Menge umfaßt alle diejenigen interaktiven Aufsetzpunkte⁶ (IAPe), welche es dem Benutzer ermöglichen, entweder eine Dialogfunktion oder eine Anwendungsfunktion auszulösen. Hierzu zählen alle Tasten der Tastatur und gegebenenfalls alle maus-sensitiven Bereiche auf dem Bildschirm, bzw. alle zusätzlichen Eingabegeräte wie Digitizer, spezielle Funktionstasten-Tablets, etc.. Alle diese interaktiven Aufsetzpunkte umspannen den **Aktionsbereich** des Benutzers.

- *die Menge aller passiven Repräsentationsformen:*

Diese Menge beinhaltet alle auf dem Ausgabemedium (meistens ein oder zwei Bildschirme; bei speziellen Tastaturen auch die LCD-Anzeigen auf den Funktionstasten) seitens des Benutzers wahrnehmbaren Signalmuster (einschliesslich der auditiven Signale). Diese Repräsentationsformen heißen passiv, weil sie ausschließlich der Rückmeldung über system-interne Zustandsparameter dienen und nicht zum Aktionsbereich des Benutzers gehören. Alle diese passiven Repräsentationsformen sind Bestandteil des **Wahrnehmungsbereich**.

Der Wahrnehmungsbereich des Benutzers umfaßt natürlich zusätzlich zu der Menge aller passiven Repräsentationsformen auch noch - sofern vorhanden - die Menge aller Signalmuster, welche die interaktiven Aufsetzpunkte repräsentieren. Hierzu zählen in der Regel alle Tasten der Tastatur (also die hardwaremäßig installierten Eingabegeräte), sowie die auf dem Bildschirm zusätzlich repräsentierten Symbole der maus-sensitiven Bereiche.

Die Menge der aktiven Repräsentationsformen läßt sich zusätzlich noch in zwei Bereiche unterteilen: die Menge der Operatoren, welche sich auf die Handhabung von Dialogobjekten (DOben) beziehen (DOpe⁷) und diejenige Menge der Operatoren, welche sich auf die Menge der Anwendungsobjekte⁸ (AObe) beziehen (AOpe⁹). Das folgende Zitat verdeutlicht am Beispiel eines "Fensters" die Menge der aktiven und passiven Repräsentationsformen dieses Dialogobjektes.

Ein einzelnes Fenster benötigt eine Menge von Zustandsvariablen (*passive Komponente*), die seine verschiedenen Parameter beschreiben (z.B. Position und Größe auf dem Bildschirm, Verweis auf den Fensterinhalt). Zusätzlich gibt es noch eine Menge von Operationen (*aktive Komponente*), die auf ein Fenster angewendet werden können. (FABIAN, 1986, S. 113)

Da in der Regel der Zustandsraum der Dialogkomponente, bzw. Anwendungskomponente sowie der zugehörigen Objekte sehr groß ist, können nur kontextspezifisch ausgewählte

⁴ Op = Operator (z.B. "dirCR")

⁵ DOOb = Dialog-Objekt (z.B. Fenster)

⁶ IAP = interaktiver Aufsetz-Punkt

⁷ DOp = dialog-objekt bezogener Operator

⁸ AOOb = Anwendungs-Objekt (z.B. Textdokument)

⁹ AOOp = anwendungs-objekt bezogener Operator

Eigenschaften aufgrund der physikalischen Begrenztheit einer jeden Ein/Ausgabeschnittstelle automatisch, d.h. *aktuell* auf der Ein/Ausgabeschnittstelle repräsentiert werden. Will der Benutzer nun Auskunft über weitere Zustands- oder Objekteigenschaften erhalten, so muß er sich diese Informationen durch die Eingabe entsprechender Operatoren beschaffen. Diese Operatoren werden nun *Feedback-Operatoren*¹⁰ (*FOpen*) genannt. Alle diejenigen Zustands- oder Objekteigenschaften, welche sich der Benutzer durch einen Feedback-Operator auf der Ein/Ausgabeschnittstelle repräsentieren lassen kann, bilden den *potentiell wahrnehmbare Zustandsraum*.

Diejenigen Zustandseigenschaften, welche weder aktuell noch potentiell wahrnehmbar repräsentierbar sind, sind für den Benutzer verborgen. Diese Menge der verborgenen Eigenschaften läßt sich allenfalls durch das Systemverhalten von dem Benutzer mittelbar erschliessen. Die Menge der Feedback-Operatoren läßt sich für die Dialogkomponente und die Anwendungskomponente getrennt definieren. Dieser Aspekt der Wahrnehmbarkeit wird unter dem Gestaltungskriterium des *Feedbacks* diskutiert und schon seit längerem gefordert (NORMAN, 1983, S.257; SPINAS, 1987, S.51).

An dem einfachen Beispiel aus der Benutzungsumgebung des Betriebssystems MS-DOS läßt sich das Konzept der Feedback-Operatoren verdeutlichen: ein Teil der Objekte der Anwendungskomponente von MS-DOS sind die auf der Festplatte, bzw. Diskette verwalteten Dateien; diese Anwendungs-Objekte lassen sich über Eigenschaften wie "Name.Extension", "Größe in Byte", "Datum der Erstellung", "Datum der letzten Änderung", "Ort der Speicherung" beschreiben, bzw. auf der Ein/Ausgabeschnittstelle repräsentieren; alle diese Eigenschaften sind jedoch nur potentiell wahrnehmbar; erst durch den Feedback-Operator "dirCR" lassen sich diese Eigenschaften sichtbar machen; andere Oberflächen für MS-DOS wie zum Beispiel PCTOOLS, QDOS, GEM oder MS-WINDOWS haben aus diesen potentiell wahrnehmbaren Eigenschaften aktuell wahrnehmbare Eigenschaften werden lassen, indem jeweils automatisch der aktuelle Directory-Inhalt auf der Ein/Ausgabeschnittstelle repräsentiert wird. Zusätzlich werden bei diesen Oberflächen-Tools auch Feedback-Operatoren dem Benutzer zur Verfügung gestellt, welche die Struktur des Directory-Subdirectory-Baumes auf der Ein/Ausgabeschnittstelle darstellt.

Im folgenden werden nur diejenigen Kriterien des benutzer-orientierten Gestaltungskonzeptes von ULICH (Abb. 1) im einzelnen vorgestellt, welche im weiteren eine besondere Rolle spielen werden. Die Beschreibung der einzelnen Kriterien ist aus SPINAS, TROY und ULICH (1983), SPINAS (1987, 1990), ULICH (1985; 1986; 1988), bzw. BAITSCH et al. (1989) entnommen.

Transparenz: Unter der Transparenz ist im Sinne des Kontrollkonzeptes die "Durchschaubarkeit" zu verstehen; sie ist eine grundlegende Voraussetzung dafür, daß der Benutzer sich eine adäquate mentale Vorstellung von der Funktionsweise des Systems aufbauen kann, welche ihm die zur erfolgreichen Aufgabebewältigung notwendigen Orientierungsgrundlagen liefert; damit verbunden ist die Möglichkeit zur "Vorhersehbarkeit" bzw. "Antizipation" der Folgen von Eingaben in das System (siehe auch NIEVERGELT, 1982).

Feedback: Feedback muß in Ablauf- und Ergebnis-Feedback unterschieden werden; in beiden Fällen geht es jedoch um die Art und Weise der Rückmeldungen seitens des Systems an den Benutzer, welche dem Benutzer gegenüber Auskunft über Erfolg oder Mißerfolg seiner Handlungen, bzw. Teilhandlungen geben. Diese Rückmeldungen können somit zur Korrektur bzw. Differenzierung der mentalen Vorstellungen über Struktur und Funktionsweise des interaktiven Systems seitens des Benutzers beitragen.

Der Bereich der "*Kontrolle*" läßt sich in verschiedene Aspekte untergliedern (siehe Abb. 1). Ein im weiteren wesentlicher Aspekt ist das Kriterium der "Flexibilität".

Flexibilität: Flexibilität meint die im System vorgesehenen Freiheitsgrade und damit die dem Benutzer prinzipiell zugestandene Autonomie zur Vorgehensweise bei seiner Aufgabebearbeitung (z.B. alternative Wege durch die Dialogstruktur, unterschiedliche Reihenfolge der einzelnen Bearbeitungsschritte zur Lösung einer Aufgabe). Jede Art von Freiheitsgrad, welche das System dem Benutzer passiv zur Verfügung stellt, fällt unter das Kriterium der Flexibilität.

Individualisierbarkeit: Im Unterschied zur Flexibilität bedeutet die Individualisierbarkeit die Möglichkeit zur aktiven Anpassung von Eigenschaften des Systems an die persönlichen, d.h. individuellen Vorstellungen und Präferenzen des Benutzers. Dies kann durch den Benutzer selbst oder durch andere in Absprache mit dem Benutzer erfolgen. Jede Art einer Veränderbarkeit eines Systems ist hier angesprochen.

10

FOp = Feedback-Operator

Die Kriterien "Unterstützung" (MOLL, 1989) und "Individualisierbarkeit" (ACKERMANN, 1987; GREUTMANN & ACKERMANN, 1987; ULICH, 1987; 1991; ULICH et al., 1990) werden an den angegebenen Stellen ausführlich diskutiert.

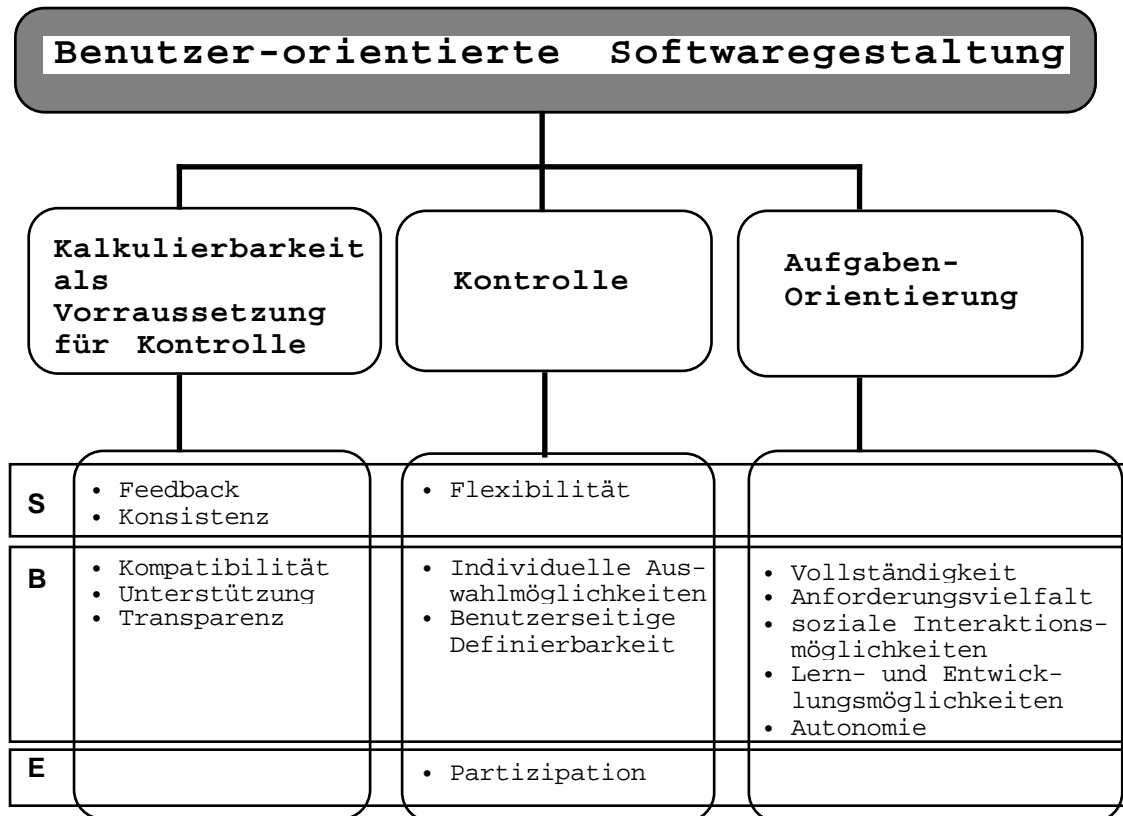


Abb. 1. Das benutzer-orientierte Gestaltungskonzept von ULICH (1991). Die den drei Bereichen "Kalkulierbarkeit...", "Kontrolle" und "Aufgabenorientierung" zugeordneten Kriterien sind hinsichtlich ihres Wirkungsbereiches vertikal weiter aufgegliedert: S bedeutet "auf Systemeigenschaften selbst bezogen"; B bedeutet "auf den Benutzungs-, Handhabungs-, bzw. Aufgabebearbeitungsprozess bezogen" und E bedeutet " auf den Erstellungsprozess bezogen".

2.1 Anforderungen an die Gestaltung interaktiver Software

Um das benutzer-orientierte Gestaltungskonzept von ULICH auf die drei Bestandteile einer Benutzungsoberfläche abbilden zu können, werden entsprechend den drei Komponenten (EAS, DK und AK) drei Gestaltungsdimensionen mit entsprechend operationalisierten Meßvorschriften abgeleitet.

Zunächst muß das Verhältnis der verschiedenen Begrifflichkeiten (Kriterium, Standard, Norm, Richtlinie, Regel, Maß, Dimension) geklärt werden (DUDEN, 1963).

Ein Kriterium ist ein unterscheidendes Merkmal und dient zur Kennzeichnung bestimmter inhaltlicher Aspekte; ein Kriterium kann als Prüfstein für eine anspruchsvolle Beurteilung verwendet werden.

Eine Richtlinie weist auf in ihr enthaltene Aspekte hin und dient der Hinlenkung der Aufmerksamkeit auf diese Inhalte. Eine Richtlinie ist weniger verbindlich als ein Standard.

Ein Standard dient primär der Vereinheitlichung und kann als Normalmaß, bzw. als Richtlinie zur Erstellung einer Normalausführung, bzw. eines Standardmodelles verwendet werden.

Eine Norm ist eine Richtlinie, eine Regel, bzw. ein Maßstab, welche primär der einheitlichen Gestaltung eines Produktes auf dem Hintergrund einer sozialen Übereinkunft ("Normung") dient. Während ein Standard einen lediglich empfehlenden Charakter hat, so gilt eine Norm als verbindlich.

Eine Regel ist eine sprachlich gegebene Vorschrift und dient als inhaltliche Konkretisierung einer Norm, bzw. eines Standards. Eine Regel kann als qualitativer "Maßstab" angesehen werden.

Ein Maß ist eine formal gegebene Vorschrift und bildet die quantitative Entsprechung zur Regel.

Eine Dimension ist Ausdehnung, bzw. Ausmaß und dient der Abmessung eines bestimmten Bereiches. Im folgenden wird unter Dimension die Zusammenfassung von verschiedenen Regeln und Maßen nach vorgegebenen Kriterien verstanden.

Die "Kriterien" werden als die inhaltlichen Schwerpunktsetzungen eines Gestaltungskonzeptes verstanden. Die Begriffe "Richtlinie", "Standard" und "Norm" zielen auf die Form und die soziale Verbindlichkeit dieser Kriterien ab. Die beiden Begriffe "Regel" und "Maß" werden als operationalisierte Vorschriften für Kriterien verwendet. Eine "Dimension" ist nun ein Sammelbecken für verschiedene, inhaltlich zusammengehörige "Regeln" und "Maße".

2.2 Die Gestaltungsmatrix als Orientierungsrahmen

Es werden die einzelnen Kriterien des ULIChschen-Gestaltungskonzeptes (Abb. 1) den einzelnen Komponenten der Benutzungsoberfläche eines interaktiven Softwaresystems zugeordnet. Hierzu soll uns die Gestaltungsmatrix als Orientierungsrahmen dienen (Abb. 2). Den Zeilen dieser Gestaltungsmatrix entsprechen die einzelnen Kriterien und den Spalten entsprechen die drei Komponenten der Benutzungsoberfläche (EAS, DK und AK).

Durch eine vorläufige Gewichtung lassen sich drei unterschiedliche Gestaltungsschwerpunkte ausmachen. Der Bereich "Kalkulierbarkeit als Voraussetzung für Kontrolle" läßt sich am ehesten der Gestaltung des Wahrnehmungsbereiches der Ein/Ausgabeschnittstelle zuordnen. Der Bereich der "Kontrolle" selbst hat dagegen seinen Gestaltungsschwerpunkt bei dem Entwurf der Dialogkomponente und der Bereich der "Aufgabenorientierung" bei der Gestaltung der Anwendungskomponente.

Für jeden dieser drei Gestaltungsschwerpunkte gilt es, möglichst qualitative Regeln und quantitative Meßverfahren zur Operationalisierung der einzelnen Kriterien zu finden. Wie man in Abb. 2 erkennen kann, ist dies bisher schon für einige Kriterien vorgesehen. Die Erläuterung dieser einzelnen Maße erfolgt weiter unten.

Nachdem wir nun im wesentlichen drei Gestaltungsschwerpunkte gemäß den drei Komponenten einer Benutzungsoberfläche und die zugehörigen Gestaltungskriterien herausgearbeitet haben, sind wir in der Lage, die den einzelnen Gestaltungs-Kriterien entsprechenden Gestaltungs-Dimensionen benennen zu können. Diese drei Gestaltungsdimensionen werden als die repräsentationale, die interaktionale und die funktionale Direktheit bezeichnet. Der Begriff "Direktheit" ergibt sich daraus, daß das interaktive System dem Benutzer möglichst *unmittelbar* die ihn interessierenden Anwendungs-Funktionen zur Aufgabenbearbeitung zur Verfügung stellen sollte. Es soll vermieden werden, daß "die Beherrschung des Werkzeuges, hier des Dialogsystems, zu einer eigenständigen Aufgabe wird, obwohl das System ein Instrument zur Unterstützung der originären Aufgaben des Benutzers sein sollte" (DZIDA, 1982, S. 5).

Bei der Gestaltungsdimension der "repräsentationalen Direktheit" geht es um die inhaltliche Ausfüllung der Abbildungsfunktionen, welche die relevanten Eigenschaften des Dialogzustandes mit den Dialog-Objekten und des Zustandes der Anwendungskomponente mit den Anwendungs-Objekten beinhaltet. Insgesamt geht es bei dieser Dimension um die repräsentational adäquate *Gestaltung des Wahrnehmungsbereiches*.

Der Schwerpunkt der Dimension der "interaktionalen Direktheit" liegt demgegenüber auf der *Gestaltung des Aktionsbereiches*. Der Aktionsbereich setzt sich jedoch grundsätzlich aus zwei Bereichen zusammen: dem Bereich der Dialog-Objekte und die auf die Dialog-Objekte anwendbaren dialog-objekt bezogenen Operatoren, sowie dem Bereich der Anwendungs-

Objekte und die auf die Anwendungs-Objekte anwendbaren anwendungs-objekt bezogenen Operatoren.

Kalkulierbarkeit als Voraussetzung für Kontrolle	Feedback	A,R,WAR-ReDi		
	Konsistenz	Symmetrie-Gesetz	Task-Action-Grammar	
	Transparenz	Methaphern-Analyse		
	Kompatibilität			
	Unterstützung			
Kontrolle	Flexibilität		HG,DiAI,AAnF,RAnF	RAuF
	Individuelle Auswahlmöglichkeiten			
	Benutzerseitige Definierbarkeit			
	Partizipation			
Aufgabenorientierung	Vollständigkeit			
	Anforderungsvielfalt			
	soziale Interaktionsmöglichkeiten			
	Lern- und Entwicklungsmöglichkeiten			
	Autonomie			
		I/O-Interface	Dialog-Komponente	Anwendungs-Komponente

Legende:

sehr wichtig
 wichtig
 relevant
 nicht relevant

Abb. 2. Abbildung der benutzer-orientierten Gestaltungskriterien auf die einzelnen Komponenten der Benutzungsoberfläche; je nach Anwendungsschwerpunkt sind die einzelnen Felder der Matrix unterschiedlich schraffiert; bisher entwickelte Maße sind in die entsprechenden Felder eingetragen.

Die Dimension der "interaktionalen Direktheit" erstreckt sich somit primär auf den Bereich der Dialog-Objekte. Die Dimension der "funktionalen Direktheit" dagegen deckt den Bereich der Anwendungs-Objekte mit den zugehörigen anwendungs-objekt bezogenen Operatoren ab. Alle Funktionen, die den Zustand von Anwendungs-Objekten verändern, heißen *anwendungs-bezogene Funktionen*. Alle übrigen Funktionen eines interaktiven Systems sind dann *dialog-bezogene Funktionen*.

3 Ausgewählte quantitative Maße

Als erstes muss die Unterscheidung zwischen der Dialog- und der Anwendungsfunktionalität etwas eingehender besprochen werden. Diese beiden Funktionsbereiche lassen sich

mit dialog-bezogenen funktionalen interaktiven Aufsetzpunkten¹¹ (DFIAP) und mit anwendungs-bezogenen funktionalen interaktiven Aufsetzpunkten¹² (AFIAP) beschreiben (siehe auch Abb. 6 weiter unten).

Die EASE läßt sich hinsichtlich der dort vorhandenen Repräsentationsformen in die den dialog-bezogener funktionaler interaktiver Aufsetzpunkt, bzw. anwendungs-bezogener funktionaler interaktiver Aufsetzpunkt zugeordneten repräsentationalen interaktiven Aufsetzpunkten unterscheiden (RDIAP¹³, bzw. RAIAP¹⁴). Die Menge der repräsentationalen IAP eines DFIAPe und der repräsentationalen anwendungs-bezogenen interaktiven Aufsetzpunkte werden unter der Menge aller repräsentationalen interaktiven Aufsetzpunkte (RIAP¹⁵) zusammengefaßt.

Formal lassen sich die verschiedenen Mengen an interaktiven Aufsetzpunkten wie folgt beschreiben:

$$\begin{aligned}
 \text{IAP} &:= \{r : r \in \text{RIAP}\} \cup \{f \mid f \in \text{FIAP}\} \\
 \text{FIAP} &:= \{d : d \in \text{DFIAP}\} \cup \{a \mid a \in \text{AFIAP}\} \\
 \text{RIAP} &:= \{r : r \in \text{RDIAP}\} \cup \{r \mid r \in \text{RAIAP}\} \\
 \text{RDIAP} &:= \{(d,r) \in \{\text{DFIAP}\} \times \{\text{RD}\} : r = \delta(d)\} \\
 \text{RAIAP} &:= \{(a,r) \in \{\text{AFIAP}\} \times \{\text{RA}\} : r = \alpha(a)\} \\
 \text{RD} &:= \{\text{alle geeigneten Repräsentationsformen für DIAPe}\} \\
 \text{RA} &:= \{\text{alle geeigneten Repräsentationsformen für AIAPe}\}
 \end{aligned}$$

Die Gestaltungsdimension der interaktionalen Direktheit hat zum Ziel, möglichst viele funktionale interaktive Aufsetzpunkte (FIAPe¹⁶) dem Benutzer auf dem aktuellen, d.h. auf dem anwendungsobjekt-bezogenen Dialogkontext direkt zur Verfügung zu stellen (siehe die Verlagerung eines AFIAP in Abb. 6). Um nun diesen Aspekt der interaktionalen Direktheit operational messbar zu machen, werden einige Beschreibungskonstrukte eingeführt. Ziel dieses Vorgehens ist es zunächst, die Länge des interaktionalen Zugriffes auf die einzelne Funktion (DFIAP, bzw. AFIAP) quantitativ angeben zu können. Um sich z.B. in einem Dialog-Baum "bewegen" zu können, stehen dem Benutzer im aktuellen Dialog-Kontext - direkt - entsprechende dialog-bezogene funktionale interaktive Aufsetzpunkte zur Verfügung.

Zunächst muß die Basis-Aktion - der *Dialogschritt* (DS) - definiert werden. Um diese Definition plausibel machen zu können, wird ein einfacher Dialog-Baum eingeführt und die Rolle des Dialog-Schrittes daran exemplarisch erläutert (Abb. 3). Wir werden also zunächst einen Dialog-Baum, einen Dialog-Operator und einen Dialog-Kontext definieren.

Die Länge der einzelnen Dialogpfade von dem obersten Dialogkontext, dem Start-Dialogkontext¹⁷ (SDKo) bis hin zu der Ebene der anwendungs-bezogenen funktionalen interaktiven Aufsetzpunkten ist für unser Beispiel für alle anwendungs-bezogene funktionale interaktive Aufsetzpunkte als Pfadlängen angegeben (Abb. 3). Sehen wir uns als erstes die Definition eines hierarchischen *Dialog-Baumes* an. Mit der Beschränkung auf einen hierarchischen Dialog-Baum gehen wir von der Annahme aus, daß der Benutzer zunächst nach dem Aufstarten eines bestimmten interaktiven Programmes sich in einem Start-Dialogkontext befindet und an der Auslösung, bzw. Aktivierung einer bestimmten Anwendungs-Funktion interessiert ist (z.B. Erzeugung eines neuen Anwendungsobjektes, etwa ein Text-Dokument). Wir gehen bei der folgenden Betrachtung davon aus, daß der Benutzer immer von dem Start-Dialogkontext ausgeht. Der Benutzer kann nur von der anwendungs-bezogenen funktionalen interaktiven Aufsetzpunkt-Ebene aus zu diesem ausgezeichneten Start-Dialogkontext gelangen. Wir werden später diesen einschränkenden Aspekt durch die Definition eines Dialog-Netzes besser berücksichtigen können. Um sich überhaupt in einem Dialog-Baum durch die Hierarchie der Dialog-Kontexte navigieren zu können, stehen dem

11 DFIAP = dialog-bezogener funktionaler interaktiver Aufsetzpunkt
 12 AFIAP = anwendungs-bezogener funktionaler interaktiver Aufsetzpunkt
 13 RDIAP = repräsentationaler IAP eines DFIAP
 14 RAIAP = repräsentationaler IAP eines AFIAP
 15 RIAP = repräsentationaler interaktiver Aufsetzpunkt {RDIAP,RAIAP}
 16 FIAP = funktionaler interaktiver Aufsetzpunkt
 17 SDKo = Start-, bzw. Wurzel-Dialogkontext

Benutzer jeweils entsprechende dialog-bezogene funktionale interaktive Aufsetzpunkte zur Verfügung (Abb. 3).

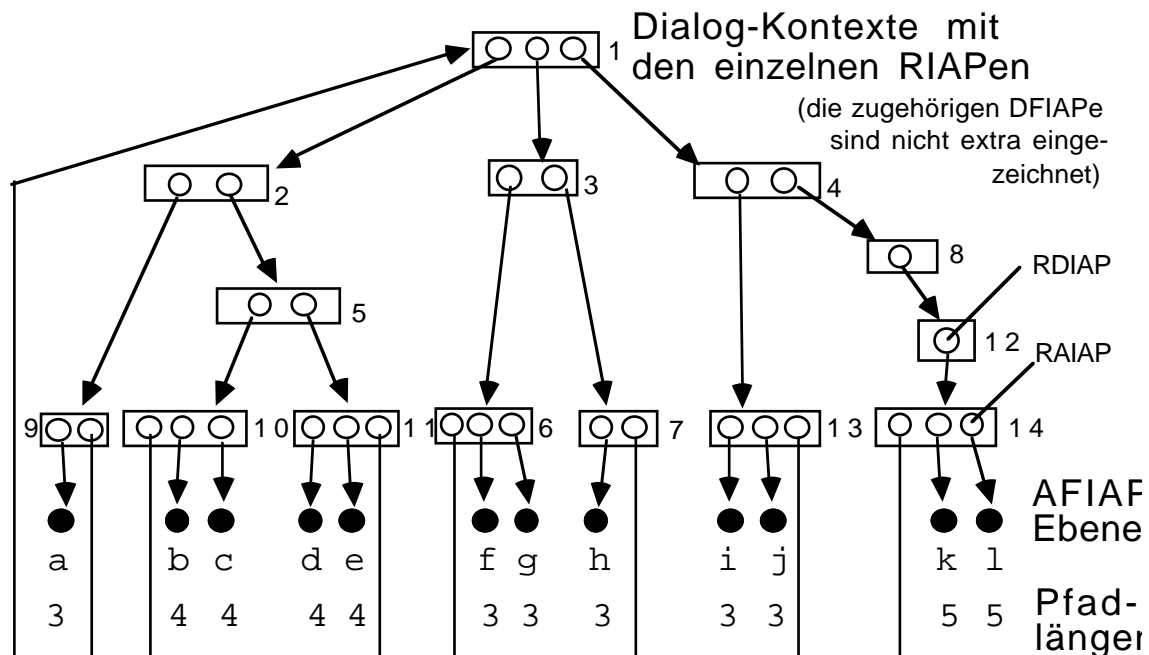
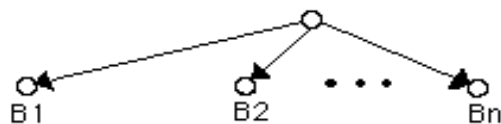


Abb 3. Schematische Darstellung eines beispielhaft gegebenen, hierarchischen Dialogstruktur-Baumes. Der Benutzer navigiert sich durch die einzelnen Dialogkontexte hindurch zu demjenigen Dialogkontext mit der gewünschten Anwendungsfunktionalität (anwendungsbezogene funktionale interaktive Aufsetzpunkt-Ebene). Nur von der anwendungsbezogenen funktionalen interaktiven Aufsetzpunkt-Ebene aus kann der Benutzer zum Wurzel-Dialogkontext zurück. Die Anzahl der Dialogschritte vom Start-Dialogkontext pro anwendungsbezogenen funktionalen interaktiven Aufsetzpunkt ist als Pfadlänge angegeben. Dieser Dialogbaum hat 14 Dialogkontexte, 32 repräsentative interaktive Aufsetzpunkte bestehend aus 12 repräsentativen anwendungsbezogenen interaktiven Aufsetzpunkten und 20 repräsentativen dialogbezogenen interaktiven Aufsetzpunkten, zu denen jeweils 12 anwendungsbezogene funktionale interaktive Aufsetzpunkte und 20 dialogbezogene funktionale interaktive Aufsetzpunkte gehören.

(Definition eines DIALOG-BAUMES)

- (1) der Knoten \odot ist ein Baum.
- (2) Wenn B_1, B_2, \dots, B_n für $n \geq 1$ Bäume sind, dann ist auch



ein Baum.

Ein Dialog-Baum ist ein gerichteter Graph (ohne geschlossene Wege; hierbei werden die Rückkehrmöglichkeiten zum Wurzelknoten vorerst aussen vor gelassen), in dem zwischen je zwei Knoten genau ein Pfad existiert. Es handelt sich um einen eingeschränkt zyklensfreien, zusammenhängenden gerichteten Graphen, welcher aus endlich vielen Knoten und Kanten besteht.

Ein Knoten in einem Dialog-Baum heißt Dialog-Kontext (DKo); eine gerichtete Kante steht für eine entsprechende Dialog-Operation (DOp).

Der Wechsel zwischen den Dialog-Kontexten (z.B. in einem Dialog-Baum) erfolgt mittels eines dialog-objekt bezogenen Operators. Ein dialog-objekt bezogener Operator läßt sich wie folgt definieren.

(Definition eines DIALOG-OPERATORS)

ein Dialog-Operator (DOp) ist die Auslösung einer dialog- oder anwendungsbezogenen Funktionalität über den entsprechenden funktionalen interaktiven Aufsetzpunkt (FIAP):

$$\text{DOp}_n = A(\text{FIAP}_m \rightarrow F_m)$$

mit ((n=m) oder (n=nil)) und ($m \geq 1$)

A ist eine systeminterne Abbildung des FIAP_m auf die zugeordnete Funktion, bzw. den Algorithmus F_m ; gemeint ist die benutzerinitiierte Auslösung, bzw. Aktivierung der zum FIAP_m gehörenden Funktion, bzw. Algorithmus F_m .

n=nil bedeutet, daß die Funktionalität F_m selbsttätig durch das System ausgelöst wird und nicht durch eine benutzerinitiierte Aktion ausgelöst werden kann; dieser Dialog-Operator heißt Dialog-Operator-Dummy.

Als nächstes müssen wir noch eine bessere Definition für den Dialog-Kontext finden, als sie in der Definition für den Dialog-Baum angeklungen ist. Ein Dialog-Kontext setzt sich aus sehr verschiedenen Elementen zusammen. Zum einen gehören sicherlich die verschiedenen interaktiven Aufsetzpunkte dazu; dies sind dann insbesondere die repräsentationalen und ihre funktional zugehörigen interaktiven Aufsetzpunkte. Dann gehören noch die repräsentationalen Zustandseigenschaften der Dialog- und der Anwendungs-Komponente dazu.

Wenn wir diese wichtigen Unterscheidungen in die Definition eines Dialog-Kontextes einbauen, so kommen wir zu der folgenden Definition:

(Definition eines DIALOG-KONTEXTes)

ein Dialog-Kontext ist ein Zwölf-Tupel der folgenden Form:

$\text{DK}_o =$

$(\text{RDIAP}_n, \text{DFIAP}_m, \delta, \text{RDIAP}_n, \text{DFIAP}_m, \alpha, \text{RDKZ}_d, \text{DKZ}, \zeta\delta, \text{RAKZ}_a, \text{AKZ}, \zeta\alpha)$

wenn gilt:

RDIAP_n ist die Menge mit n-Elementen aus der Menge aller repräsentationalen dialog-bezogenen interaktiven Aufsetzpunkte.

DFIAP_m ist die Menge mit m-Elementen aus der Menge aller dialog-funktionalen interaktiven Aufsetzpunkte.

δ ist die kontext-spezifische Abbildungsfunktion der m DFIAP_e auf die n repräsentationalen dialog-bezogenen interaktiven Aufsetzpunkte.

RAIAP_o ist die Menge mit o-Elementen aus der Menge aller repräsentationalen anwendungs-bezogenen interaktiven Aufsetzpunkte.

AFIAP_p ist die Menge mit p-Elementen aus der Menge aller anwendungs-funktionalen interaktiven Aufsetzpunkte.

α ist die kontext-spezifische Abbildungsfunktion der p AFIAP_e auf die o repräsentationalen anwendungs-bezogenen interaktiven Aufsetzpunkte.

RDKZ_d ist die Menge der repräsentierten Zustände (d an der Zahl) aus der Menge aller Zustände der Dialog-Komponente auf der E/A-Schnittstelle.

DKZ ist die Menge aller Zustände der Dialog-Komponente.

$\zeta\delta$ ist die kontext-spezifische Abbildungsfunktion der d Zustände der Dialog-Komponente in ihre repräsentationale Form RDKZ .

RAKZ_a ist die Menge der repräsentierten Zustände (a an der Zahl) aus der Menge aller Zustände der Anwendungs-Komponente auf der E/A-Schnittstelle.

AKZ ist die Menge aller Zustände der Anwendungs-Komponente.

$\zeta\alpha$ ist die kontext-spezifische Abbildungsfunktion der a Zustände der Anwendungs-Komponente in ihre repräsentationale Form RAKZ .

Das wichtigste sind jedoch sicherlich die jeweils zu den einzelnen Bereichen gehörenden Abbildungsvorschriften, bzw. Abbildungsfunktionen. Hinter diesen Abbildungsfunktionen (δ , α , $\zeta\delta$, $\zeta\alpha$) verbergen sich im Detail die noch ausstehenden Forschungsergebnisse, wie sie für die Gestaltungsdimension der repräsentationalen Direktheit vorzusehen sind (siehe auch CORDES, 1988). Diese Abbildungsfunktionen werden leider in einigen formalen Betrachtungen einer Dialogsprache ausgegrenzt (z.B. ZACHMANN, 1989; S.57), wodurch die Kriterien Transparenz und Feedback nicht adäquat berücksichtigt werden können.

Entsprechend ist die Abbildungsfunktion der funktionalen interaktiven Aufsetzpunkte auf die entsprechenden repräsentationalen interaktiven Aufsetzpunkte aufgeteilt worden. Wir benutzen dazu die Menge aller dialog-funktionalen interaktiven Aufsetzpunkte (DFIAPe) und die Menge aller anwendungs-funktionalen interaktiven Aufsetzpunkte (AFIAPe). Diesen beiden Mengen werden über die Abbildungsfunktionen entsprechende Repräsentationen auf der Ein/ Ausgabeschnittstelle zugeordnet: die Menge der repräsentationalen dialog-bezogenen interaktiven Aufsetzpunkte und der repräsentationalen anwendungs-bezogenen interaktiven Aufsetzpunkte (siehe auch Abb. 6 auf S.17).

Der Wechsel zwischen den Dialog-Kontexten (z.B. in einem Dialog-Baum) soll *Dialog-Schritt* heißen und läßt sich wie folgt definieren.

(Definition eines DIALOG-SCHRITTES)

ein Dialog-Schritt ist die Folge von einem Dialog-Operator (DOP) und einem Dialog-Kontext (DKo):

$$DS_n = (DOP_n, DKo_m)$$

mit ($n \geq 1$) und (($m=n$) oder ($m=n+1$))

Ein Dialog-Schritt ist somit der Übergang von einem gegebenen DKO durch Auslösung einer Funktion F (über einen entsprechenden funktionalen interaktiven Aufsetzpunkt) zu einem folgenden DKO. Wenn $m=n$ ist, handelt es sich um einen zustands-, bzw. kontext-bewahrenden Dialogschritt. In diesem Fall reden wir auch von einer elementaren Dialog-Schlinge.

(Definition eines DIALOG-PFADES)

ein Dialog-Pfad ist die Folge von aufeinanderfolgenden Dialog-Schritten:

$$Pfd [DS_1 \dots DS_n] = [(DOP_1, DKo_2), (DOP_2, DKo_3), \dots, (DOP_n, DKo_m)]$$

mit ($n \geq 1$) und (($m=n$) oder ($m=n+1$))

Zur Verbesserung der Übersicht wird nun die folgende, abkürzende Schreibweise für Dialog-Pfade eingeführt:

$$Pfd [DS_1 \dots DS_n] = [DOP_1, DOP_2, DOP_3, \dots, DOP_n]$$

mit ($n \geq 1$)

Ein Dialog-Schritt¹⁸ (DS) ist also der Wechsel eines Dialog-Kontextes durch Eingabe, bzw. durch Auslösung einer Dialog- oder Anwendungsfunktion zu einem nächsten Dialog-Kontext. Hiermit ist der Zustandsübergang im Zustandsraum der Dialog- und Anwendungskomponente (DKZ, bzw. AKZ) gemeint. Es gibt zustandsbewahrende und zustandsverändernde Dialogschritte. Ein zustandsbewahrender Dialogschritt heißt Dialogschlinge. Fehleingaben eines Benutzer sind häufig zustandsbewahrende Dialogschritte (z.B. Drücken einer sinn-"losen" Taste), sofern das interaktive System nicht extra mit einer Fehlermeldung oder sonstigen Hilfe-Ausgabe reagiert. Definieren wir nun die Abfolge von Dialogschritten als ein zusammenhängendes Ganzes, so kommen wir zu der angegebenen Definition des Dialog-Pfades.

Mit dieser hier vorgelegten Definition eines Dialogpfades wird die Struktur eines Logfiles abstrakt beschrieben. Man kann also direkt aus den meisten Logfile-Aufzeichnungen die

¹⁸ DS = Dialog-Schritt

Länge einzelner Dialog-Pfade ablesen. Offen bleibt dabei zunächst noch das Problem, wie und nach welchen Kriterien man den Anfangs- und den Endpunkt eines Dialog-Pfades bestimmt. Wie dieses Problem gelöst werden kann, bleibt vorerst offen.

Sehen wir uns nun die Länge eines Dialog-Pfades etwas näher an und definieren sie wie folgt:

(Definition der LÄNGE eines DIALOG-PFADES)

Die Länge eines Dialog-Pfades ist die Anzahl der Dialog-Schritte zwischen dem Start- und dem End-Dialogkontext des Pfades:

$$\text{lng}(\text{Pfd}[\text{DS}_1 \dots \text{DS}_n]) = n.$$

Wenn wir die eingeschränkte Gültigkeit des Dialogbaumes verlassen wollen, so müssen wir zu der Beschreibung eines entsprechenden Dialog-Netzes übergehen. Hierzu haben wir das Beispiel des Dialog-Baumes um einige Kanten erweitert und kommen zu unserem einfachen Beispiel-Netz (Abb. 4). In der Abbildung 4 sind zu jedem funktionalen interaktiven Aufsetzpunkt (a-l) die entsprechenden Pfadlängen für die interaktive Distanz vom Start-Kontext bis hin zur anwendungs-bezogenen funktionalen interaktiven Aufsetzpunkt-Ebene für die verschiedenen Dialog-Pfade angegeben. Man beachte, daß im DKO(1) nur *ein* repräsentationaler dialog-bezogener interaktiver Aufsetzpunkt ist.

(Definition eines DIALOG-NETZes)
 ein Dialog-Netz ist ein gerichteter Graph

$DN = (DKo, DOp, DS)$
 wenn gilt:
 DKO ist eine endliche Menge an Dialog-Kontexten.
 DOP ist eine endliche Menge an Dialog-Operatoren.
 DS ist eine endliche Menge von Dialog-Schritten, welche die Zuordnungsfunktion zwischen den Elementen aus der Menge der Dialog-Operatoren zu den Elementen aus der Menge der Dialog-Kontexte darstellt.

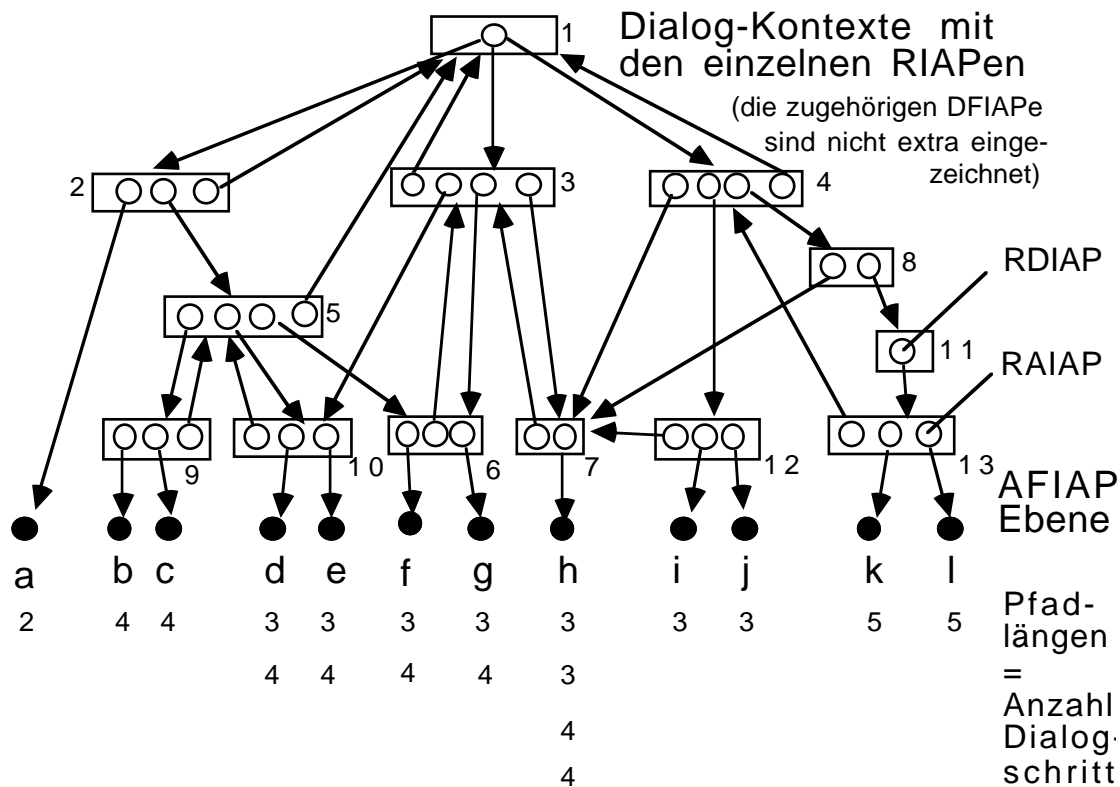


Abb. 4. Schematische Darstellung einer beispielhaft gegebenen, netzartigen Dialogstruktur. Der Benutzer navigiert sich durch die einzelnen Dialogkontexte hindurch zu der jeweils gewünschten Anwendungsfunktionalität, wobei ihm jeweils unterschiedlich viele alternative Dialogpfade zur Verfügung stehen. Die alternativen Dialogpfade sind - vom Start-Dialogkontext (1) aus betrachtet - z.T. unterschiedlich lang. Zu jedem der 38 Pfeile im Diagramm gehört ein dialog-bezogener funktionaler interaktiver Aufsetzpunkt, bzw. anwendungs-bezogener funktionaler interaktiver Aufsetzpunkt. Dieses Dialog-Netz enthält 36 repräsentationale interaktive Aufsetzpunkte; es sind den 26 dialog-bezogenen funktionalen interaktiven Aufsetzpunkten jedoch nur 24 repräsentationale dialog-bezogene interaktive Aufsetzpunkte zugeordnet; den 12 anwendungs-bezogenen funktionalen interaktiven Aufsetzpunkten genau 12 repräsentationale anwendungs-bezogene interaktive Aufsetzpunkte.

Dialog-Flexibilität drückt sich in der Vielfalt der dem Benutzer angebotenen *Dialog-Funktionalität* aus. Diese Vielfalt in der Dialog-Funktionalität - insbesondere in der Dialog-Struktur (z.B. Maus-Steuerung und Funktionstasten) - führt zu netz-artigen Dialogstrukturen und beinhaltet die *Abfolge der Dialog-Funktionen*. Da die meisten modernen Oberflächen immer stärker in die Richtung auf netzartige Dialogstrukturen hin entwickelt werden (siehe z.B. der Bereich "HyperText", etc.), so haben wir uns zunächst die angegebene Definition für ein Dialog-Netz zu verschaffen.

3.1 I/O-Interface und repräsentationale Direktheit (RD)

Um als grobe Abschätzung das Kriterium "Feedback" quantitativ bestimmen zu können, läßt sich das Verhältnis der Anzahl an repräsentationalen interaktiven Aufsetzpunkten zu der Anzahl an funktionalen interaktiven Aufsetzpunkten als Mittelwert über alle Dialogkontexte hinweg errechnen (Formel 1). Wir gehen dabei von der folgenden Überlegung aus: je mehr repräsentationale interaktive Aufsetzpunkte vorhanden sind, desto größer ist auch das Ausmaß an repräsentationaler Direktheit.

In diesem Sinne wird zunächst die RD *ausschließlich* auf die Repräsentation der dialog-objekt bezogenen Operatoren und anwendungs-objekt bezogenen Operatoren beschränkt! Der gesamte Bereich der Dialogkomponente- und Anwendungskomponente-Zustände bleibt vorerst unberücksichtigt. Darüber hinaus ist bei den beiden folgenden Maßen (AReDi und RReDi) auch nicht die Trennung zwischen Bildschirm und Tastatur erfaßt; so ist z.B. ein mausensensitiver Bereich repräsentational direkter als die Erklärung der Bedeutung einer Funktionstastenbelegung auf dem Bildschirm, weil für diesen letzteren Fall der Benutzer immer noch die Semantik der jeweiligen Taste kognitiv kurzfristig für die Dauer der Zuwendungsreaktion hin zur Tastatur zwischenspeichern muß.

Es wird zunächst nicht zwischen der Dialog- und der Anwendungsfunktionalität unterschieden, obwohl sich das Maß (AReDi) auch jeweils getrennt für diese beiden Funktionsbereiche angeben läßt.

(Formel 1)

$$AReDi = \frac{1}{D} \sum_{d=1}^D \#RIAP_d * |\text{sqrt}(\#RIAP_d / \#FIAP_d)|$$

AReDi absolutes Maß für die *repräsentationale Direktheit*;

#RIAP_d Anzahl an repräsentationalen interaktiven Aufsetzpunkten bezüglich Dialogkontext d;

#FIAP_d Anzahl an funktionalen interaktiven Aufsetzpunkten bezüglich Dialogkontext d;

D Anzahl an Dialog-Kontexten der Oberfläche insgesamt.

[sqrt (..) := Square_root_of (..)]

Für das Dialogbaum-Beispiel ist AReDi = 2.3; d.h., daß pro Dialogkontext dem Benutzer im Mittel 2.3 repräsentationale interaktive Aufsetzpunkte zur Auswahl zur Verfügung stehen. Für das Dialognetz-Beispiel ist AReDi = 2.8. Das so errechnete Maß AReDi ist also in seiner *absoluten* Größe noch abhängig von der Gesamtzahl an funktionalen interaktiven Aufsetzpunkten im interaktiven System und kann daher nicht zum Vergleich zwischen verschiedenen Systemen mit unterschiedlicher Funktionalität herangezogen werden. Um diesen Nachteil auszugleichen, wurde das folgende, *relative Maß* entwickelt (Formel 2: RReDi).

(Formel 2)

$$RReDi = \frac{1}{D} \sum_{d=1}^D \#RIAP_d / \#FIAP_d * 100\%$$

RReDi relatives Maß für die *repräsentationale Direktheit*;

#RIAP_d Anzahl an repräsentationalen interaktiven Aufsetzpunkten bezüglich Dialogkontext d;

#FIAP_d Anzahl an funktionalen interaktiven Aufsetzpunkten bezüglich Dialogkontext d;

D Anzahl an Dialog-Kontexten der Oberfläche.

Für das Dialogbaum-Beispiel ist RReDi = 100%; d.h., daß alle funktionalen interaktiven Aufsetzpunkte dem Benutzer als repräsentationale interaktive Aufsetzpunkte zur Auswahl zur Verfügung stehen. Für das Dialognetz-Beispiel ist jedoch wegen der fehlenden repräsentationale dialog-bezogene interaktive Aufsetzpunkte aus DKo(1) WRRReDi = 95%. Im Unterschied zum Maß AReDi kann RReDi für Systeme mit unterschiedlich vielen funktionalen interaktiven Aufsetzpunkten verwendet werden.

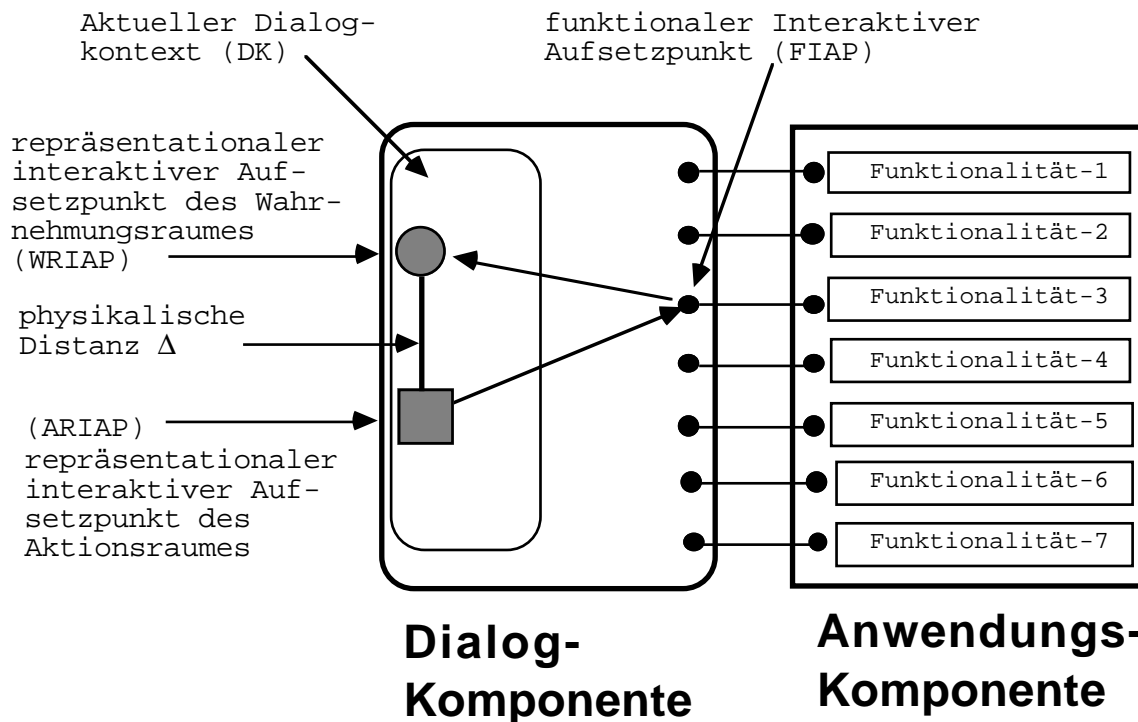


Abb. 5. Die Unterscheidung der Menge der repräsentationalen interaktiven Aufsetzpunkte (RIAPe) in die Menge der Elemente des Wahrnehmungsraumes (WRIAPe) und die Menge der Elemente des Aktionsraumes (ARIAPe)

Es bleibt jedoch noch der Aspekt unberücksichtigt, daß der für einen funktionalen interaktiven Aufsetzpunkt die Semantik tragende, perzeptuelle repräsentationale interaktive Aufsetzpunkt räumlich nicht mit dem aktionalen repräsentationalen interaktiven Aufsetzpunkt zusammenfällt. Dies ist z.B. immer dann der Fall, wenn auf dem Bildschirm die Bedeutung (in irgendeiner Form) für einzelne Tasten (insbesondere Funktionstasten) als repräsentationaler interaktiver Aufsetzpunkt gegeben ist. Wir müssen also zwischen der Menge der *Elemente des Wahrnehmungsraumes*¹⁹ (WRIAPe) und der Menge der *Elemente des Aktionsbereiches*²⁰ (ARIAPe) unterscheiden (Abb. 5). Diese Unterscheidung ist deshalb wichtig, weil hier die eigentlich zusammengehörenden Elemente des Wahrnehmungsbereiches und des Aktionsraumes auseinanderfallen. Diese Distanz ist durch die physikalische Entfernung Δ zwischen repräsentationalen interaktiven Aufsetzpunkt des Wahrnehmungsraumes und repräsentationalen interaktiven Aufsetzpunkt des Aktionsraumes bestimmbar (z.B. Angaben in cm).

Gegeben sei das folgende Beispiel: auf dem Bildschirm wird zu jedem aktuellen Dialogkontext auf der untersten Bildschirmzeile die Belegung der Funktionstasten angezeigt. Die Menge der repräsentationalen interaktiven Aufsetzpunkt des Aktionsraumes ist z.B. die Menge der Funktionstasten auf der Tastatur selbst, zumeist mit Benennungen wie F1.., bzw. PF1.. beschriftet. Die Menge der repräsentationalen interaktiven Aufsetzpunkte des Wahrnehmungsraumes ist die Menge der Bezeichnungen auf der untersten Bildschirmzeile: PF1=HELP oder ähnlich. Diese Zuordnung muß vom Benutzer entweder im (internen)

¹⁹ WRIAP = repräsentationaler interaktiver Aufsetzpunkt des Wahrnehmungsraumes

²⁰ ARIAP = repräsentationaler interaktiver Aufsetzpunkt des Aktionsraumes

Kurzzeit- oder bei längerer Übung im Umgang mit der jeweiligen Software auch im Langzeitgedächtnis für die handlungleitende Entscheidung verfügbar sein.

Eine andere Möglichkeit besteht darin, daß der Benutzer diese Zuordnung im externen visuellen "Gedächtnis" (die Ein/Ausgabeschnittstelle) beläßt und die jeweilige Funktionstaste ohne Blickwendung hin zur Tastatur findet und betätigt. Je größer nun die physikalische Distanz zwischen repräsentationalem interaktivem Aufsetzpunkt des Wahrnehmungsraumes und repräsentationalem interaktivem Aufsetzpunkt des Aktionsraumes ist, desto stärker ist für den ungeübten Benutzer die - kurzfristige - Gedächtnisbelastung.

(Formel 3)

$$\text{WARReDi} = 1/D \sum_{d=1}^D \frac{1/I_d \sum_{i=1}^{I_d} \#RIAP_{id}}{(\#FIAP_d * \Delta_i/K)} * 100\%$$

WARReDi	gewichtetes, relatives Maß für die <i>repräsentationale Direktheit</i> ;
#RIAP _d	Anzahl an repräsentationalen interaktiven Aufsetzpunkten bezüglich Dialogkontext d;
#FIAP _d	Anzahl an funktionalen interaktiven Aufsetzpunkten bezüglich Dialogkontext d;
D	Anzahl an Dialog-Kontexten der Oberfläche;
WRIAP _i	der i-te repräsentationale interaktive Aufsetzpunkt des Wahrnehmungsraumes;
ARIAP _i	der i-te repräsentationale interaktive Aufsetzpunkt des Aktionsraumes;
Δ _i	= WRIAP _i - ARIAP _i _d der Absolutbetrag der physikalisch messbaren Distanz des i-ten repräsentationalen interaktiven Aufsetzpunktes bezüglich Dialogkontext d; Angabe in cm;
K	Korrekturfaktor für die Distanz; z.B K=10cm; falls Δ _i null ergeben sollte, so trägt K dafür Sorge, daß (Δ _i /K) = (1) wird.
I _d	Anzahl der vorhandenen Distanzen bezüglich des Dialogkontextes d;

Erweitern wir nun das Maß für RReDi (Formel 2) um diese physikalische Distanz, so kommen wir zur Formel 3. Die Distanz eines gegebenen repräsentationalen interaktiven Aufsetzpunktes wird in physikalischen Maßeinheiten gemessen (z.B. in cm). Um nun dieser Distanz Δ ein angemessenes Gewicht innerhalb des Gesamtmaßes zu geben, wird der Korrekturfaktor K eingeführt. Dieser Korrekturfaktor K dient zusätzlich noch dazu, die physikalische Maßeinheit herauszukürzen und hat daher selbst die jeweilige Maßeinheit der Distanz Δ.

Bei direktmanipulativen Oberflächen wird für die maus-sensitiven Bereiche (bzw. die entsprechenden repräsentationalen interaktiven Aufsetzpunkte) diese Distanz Null sein. Dies kann deshalb als sinnvoll angenommen werden, weil bei dem Interaktionslement "Maus" der entsprechende repräsentationale interaktive Aufsetzpunkt des Aktionsraumes nicht die Maustaste selbst, sondern der Maus-Cursor auf dem Bildschirm ist. Dies gilt dann auch ebenso für die Cursor-Tasten der herkömmlichen Tastatur, bzw. für alle Arten der Interaktionssteuerung, bei der ein wahrnehmbarer repräsentationaler interaktiver Aufsetzpunkt des jeweiligen Aktionsraumes auf dem Bildschirm vorhanden ist. Hierbei wird angenommen, daß die jeweilige Positionierung, bzw. Steuerung des repräsentationalen interaktiven Aufsetzpunkt des Aktionsraumes auf dem Bildschirm auf der sensu-motorischen Regulationsebene ausgeführt werden kann, ohne dabei mit den höheren Regulationsebenen zu interagieren.

Bei der Anwendung des Maßes WARReDi (Formel 3) wird der Unterschied einer traditionellen Menü-Oberfläche mit den üblichen Auswahl-Menüs (Zahlen oder Buchstaben-Codierungen für die entsprechenden Menü-Optionen) zu einer maus-gesteuerten direktmanipulativen Oberfläche deutlich. Bei der direktmanipulativen Oberfläche werden die Distanzen Δ = |WRIAP-ARIAP| überwiegend Null ergeben.

Diese bisher vorgestellten Maße geben natürlich nur einen *rein quantitativen Aspekt* der repräsentationalen Direktheit wieder und lassen die Problematik der semantischen Bedeutung

der einzelnen repräsentationalen interaktiven Aufsetzpunkte völlig außer acht (siehe die beachtenswerten Ergebnisse von STREITZ, LIESER & WOLTERS; 1989).

3.2 Dialogkomponente und interaktionale Direktheit (ID)

Um das Kriterium der "Konsistenz" bezüglich der Dialogkomponente näher erfassen zu können, wurden eine Reihe von Grammatik-Modellen zur formalen Beschreibung der interaktiven Strukturen entwickelt (einen Überblick gibt RAUTERBERG, 1989b). Einer der konsequentesten Ansätze ist die "Task-Action-Grammar".

Die Aufgaben-Handlungs-Grammatik (AHG) (engl. "Task-Action-Grammar", TAG) wird in ihren Grundzügen von TAUBER & SCHIELE (1986, S.248-252) beschrieben.

Mit dem AHG-Modell von PAYNE & GREEN (1986) wird ein Ansatz vorgestellt, der das Wissen des Benutzers um die Umsetzung seiner Aufgaben bezogenen Ziele in die vom System zur Verfügung gestellten Aktionen thematisiert. Insbesondere sollen hier die semantischen Aspekte bei der Bearbeitung von Aufgaben berücksichtigt werden.

Als ein Nachteil der AHG stellte sich die stark eingeschränkte Möglichkeiten zur Modellierung von Unterschieden in Dialog- und Oberflächenstrukturen heraus. Dieser Nachteil konnte empirisch festgestellt werden, als in der Untersuchung von TAVOLATO & WEIDMANN (1989) eine Erweiterung der AHG vorgenommen werden mußte, um die Unterschiede zwischen einer menü-orientierten und einer direkt-manipulativen Oberflächenstruktur einfangen zu können.

Es werden nun für die Gestaltungsdimension der "interaktionalen Direktheit" einige quantitative Maße entwickelt und diskutiert. Als erstes möchte ich jedoch die Unterscheidung zwischen der Dialog- und der Anwendungsfunktionalität etwas eingehender besprechen. Diese beiden Funktionsbereiche werden mit dialog-bezogenen funktionalen interaktiven Aufsetzpunkten und mit anwendungs-bezogenen funktionalen interaktiven Aufsetzpunkten bezeichnet (Abb. 6).

Die E/A-Schnittstelle läßt sich hinsichtlich der dort vorhandenen Repräsentationsformen in die den dialog-bezogenen funktionalen interaktiven Aufsetzpunkt, bzw. anwendungs-bezogenen funktionalen interaktiven Aufsetzpunkt zugeordneten repräsentationalen interaktiven Aufsetzpunkten unterscheiden (RDFIAP, bzw. RAFIAP). Die Gestaltungsdimension der interaktionalen Direktheit hat zum Ziel, möglichst viele der anwendungs-bezogenen funktionalen interaktiven Aufsetzpunkte bei möglichst wenigen dialog-bezogenen funktionalen interaktiven Aufsetzpunkten dem Benutzer in dem aktuellen, d.h. in dem anwendungsobjekt-bezogenen Dialogkontext möglichst direkt zur Verfügung zu stellen (siehe das Beispiel der Verlagerung eines anwendungs-bezogenen funktionalen interaktiven Aufsetzpunkt in Abb. 6).

Um nun diesen Aspekt der interaktionalen Direktheit operational messbar zu machen, wurde die *Länge eines Dialogpfades* eingeführt. Ziel dieses Vorgehens ist es, die Länge des interaktionalen Zugriffes auf die einzelne Funktion (DFIAP, bzw. AFIAP) quantitativ angeben zu können. Um sich z.B. in einem Dialog-Baum "bewegen" zu können, stehen dem Benutzer im aktuellen Dialog-Kontext - direkt - entsprechende dialog-bezogene funktionale interaktive Aufsetzpunkte zur Verfügung, um z.B. von Dialogkontext zu Dialogkontext zu gelangen.

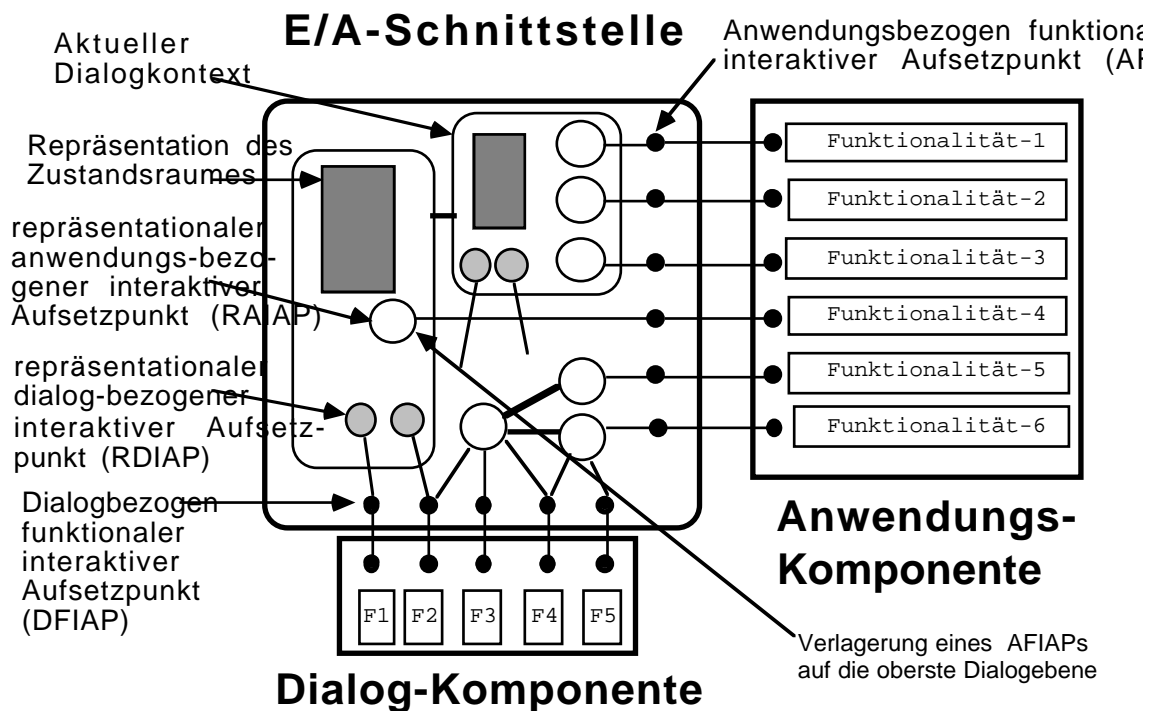


Abb. 6. Schematische Aufteilung der Benutzungsoberfläche in die dialog- und die anwendungsbezogenen interaktiven Aufsetzpunkte. Direktheit im Sinne der interaktionalen Direktheit heißt, diese Funktionen dem Benutzer auf dem obersten Dialogkontext adäquat repräsentational zur Verfügung zu stellen.

Um diese Nachteile zu überwinden, werden für einige ausgewählte Aspekte der GeMit der Länge eines Dialogpfades besteht nun die Möglichkeit eines vorläufigen, allgemeinen Maßes für den *Hierarchisierungsgrad* (siehe SPINAS, 1987, S.117) bei der interaktionalen Direktheit für einen hierarchischen Dialog-Baum angeben zu können. Berechnen wir die *mittlere Länge aller Dialogpfade* einer Oberfläche, so haben wir die gesuchte Kenngröße HG für diesen Aspekt der interaktionalen Direktheit (Formel 4).

(Formel 4)

$$HG = \frac{1}{N} \sum_{n=1}^N \ln g(\text{Pfd}_n)$$

HG relatives Maß für die *Hierarchisierung* der interaktionalen Direktheit;

$\ln g(\text{Pfd}_n)$ Länge des Dialogpfades n vom [S]DKo hin zur AFIAP-Ebene;

N Anzahl an Dialog-Pfaden zwischen [S]DKo und jedem AFIAP.

Wenden wir dieses Maß HG auf den hierarchischen Dialogbaum (Abb. 3) an, so ergibt sich eine mittlere Pfadlänge von 3.7 für die verschiedenen Dialogpfade vom Start-Dialogkontext bis hin zu den 12 Anwendungsfunktionen. Wenden wir dagegen dieses Maß auf unser beispielhaft eingeführtes Dialognetz (Abb. 4) an, so ergibt sich eine mittlere Pfadlänge von 3.6 für die insgesamt 19 verschiedenen Dialogpfade (siehe auch Tab. 1). Wie man sehen kann, zeichnet sich das Dialognetz-Beispiel durch einen geringeren mittleren Hierarchisierungsgrad aus.

Um jedoch den offensichtlicheren Unterschied zwischen der baum- und der netz-artigen Struktur quantitativ bestimmen zu können, wird ein Maß für das Kriterium der *Dialog-Flexibilität* eingeführt. Diese Dialog-Flexibilität einer Oberfläche läßt sich als die Anzahl der alternativen Dialogpfade hin zum selbem funktionalen interaktiven Aufsetzpunkt operationalisieren (SPINAS, 1987, S. 121).

Da die meisten Benutzer sich individuell unterscheiden und diese Unterschiede einen wesentlichen Einfluß auf die jeweiligen Problemlösungen haben, ist es sinnvoll, diese Unterschiedlichkeit im Systementwurf adäquat zu berücksichtigen. Wir betrachten hier zunächst nur die Möglichkeit, im Rahmen der Dialog-Funktionalität den inter- und intra-individuellen Unterschieden entgegenkommen zu können; je mehr Wege einem Benutzer hin zu einem funktionalen interaktiven Aufsetzpunkt, z.B. anwendungs-bezogenen funktionalen interaktiven Aufsetzpunkt zur Verfügung stehen, desto flexibler können die Benutzer auf diesen anwendungs-bezogenen funktionalen interaktiven Aufsetzpunkt zugreifen (Abb. 7).

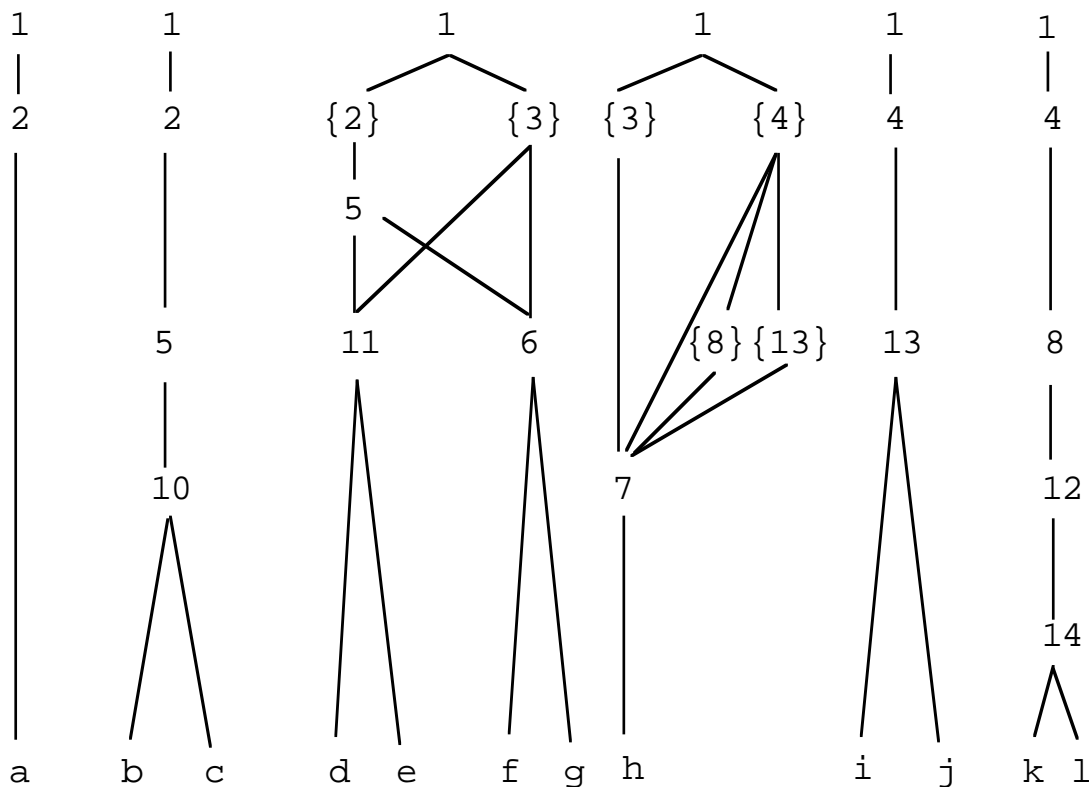


Abb. 7. Übersicht über alle Dialogpfade vom Start-Dialogkontext bis hin zu jeder Anwendungsfunktion (a-l) aus unserem Dialog-Netz-Beispiel (Abb. 4); die *alternativen Dialogpfade* sind durch geschweifte Klammern gekennzeichnet.

Sehen wir uns alle *alternativen Dialog-Pfade* für unser Dialog-Netz am Beispiel der anwendungs-bezogenen funktionalen interaktiven Aufsetzpunkte "d" und "h" an.

<p>Menge aller alternativen Dialog-Pfade für den AFIAP "d":</p> <p>MaPfd(d) = {[1,2,5,11], [1,3,11]}</p> <p>Die Kardinalität κ von MaPfd gibt die Anzahl # der Dialog-Alternativen für den AFIAP "d" wieder: $\kappa \text{ MaPfd(d)} = 2$</p>
<p>Menge aller alternativen Dialog-Pfade für den AFIAP "h":</p> <p>MaPfd_h = {[1,3,7], [1,4,7], [1,4,8,7], [1,4,13,7]}</p> <p>Die Kardinalität κ von MaPfd gibt die Anzahl # der Dialog-Alternativen für den AFIAP "h" wieder; $\kappa \text{ MaPfd(h)} = 4$</p>

Die Kardinalität der Menge aller alternativen Dialog-Pfade für jeden funktionalen interaktiven Aufsetzpunkt bildet die Grundlage für ein Maß der Flexibilität der interaktionalen Direktheit:

die *Dialog-Flexibilität*, abgekürzt DiAl. Errechnen wir nun den Mittelwert der Kardinalitäten aller funktionalen interaktiven Aufsetzpunkte, so erhalten wir die mittlere Anzahl an Dialogalternativen (Formel 5).

(Formel 5)

$$\text{DiAl} = 1/(\#\text{FIAP}) \sum_{f \in \text{FIAP}} \#\text{MaPfd}(f)$$

DiAl relatives Maß für die *Dialogalternativen* der interaktionalen Direktheit ("Flexibilität");

#MaPfd(f) Anzahl an alternativer Dialogpfade für die Funktion f;

FIAP Menge mit allen Funktionen der DK und AK.

Sehen wir uns nun in der Tabelle 1 die jeweilige Anzahl an alternativen Dialogpfaden für die 12 funktionalen interaktiven Aufsetzpunkte unseres Dialognetz-Beispiels (Abb. 4) an. Die alternativen Dialogpfade sind in der Abbildung 7 veranschaulicht. DiAl für unser Netz-Beispiel anhand der Tabelle 1 errechnet sich zu $\text{DiAl} = 1.58$. Im Unterschied dazu ergibt sich die Dialog-Flexibilität des hierarchischen Dialog-Baumes zu $\text{DiAl} = 1.00$; denn es gibt dort keine Dialogalternativen!

Tabelle 1 Anzahl der alternativen Dialog-Pfade eingeschränkt auf die Anwendungsfunktionalität des Netz-Beispiels aus Abbildung 4 (siehe dazu auch Abb. 7).

AFIAP	MaPfd = Menge der alternativen Dialog-Pfade	#MaPfd
a	[1,2]	1
b	[1,2,5,10]	1
c	[1,2,5,10]	1
d	[1,2,5,11], [1,3,11]	2
e	[1,2,5,11], [1,3,11]	2
f	[1,2,5,6], [1,3,6]	2
g	[1,2,5,6], [1,3,6]	2
h	[1,3,7], [1,4,7], [1,4,8,7], [1,4,13,7]	4
i	[1,4,13]	1
j	[1,4,13]	1
k	[1,4,8,12,14]	1
l	[1,4,8,12,14]	1

Wenden wir uns der eigentlichen, *anwendungs-bezogenen Flexibilität* zu. Das hierzu entwickelte Maß für das Kriterium der Flexibilität bezieht sich auf die Handhabung der Anwendungsfunktionen. Hiermit ist gemeint, daß der Benutzer in jedem Dialog-Kontext die Möglichkeit haben sollte, auf eine möglichst große Vielfalt der Anwendungsfunktionen zuzugreifen zu können. Dies würde dem Benutzer nicht nur sehr unterschiedliche Aufgabenstellungen zu bearbeiten erlauben, sondern es auch möglich machen, ein und dieselbe Aufgabe auf unterschiedlichen Lösungswegen abzarbeiten (Abfolge der Anwendungsfunktionen).

Das Maß für die anwendungsbezogene Flexibilität beruht auf der Überlegung, daß die Anzahl der im aktuellen Dialogkontext direkt vorhandenen Anwendungsfunktionen einen brauchbarer Indikator darstellt. Wir nennen diese anwendungsbezogene Flexibilität AAnF. Das quantitative Maß für die *anwendungsbezogene Flexibilität "AAnF"* läßt sich wie folgt definieren (Formel 6):

(Formel 6)

$$AA_nF = 1/D \sum_{d=1}^D \#AFIAP_d$$

AA_nF absolutes Maß für die *anwendungsbezogene Flexibilität*;
 #AFIAP_d Anzahl an anwendungs-bezogenen funktionalen interaktiven Aufsetzpunkten im Dialogkontext d;
 D Anzahl aller vorhanden Dialogkontexte DKo.

Es ist jedoch mit diesem Maß noch keine Aussage über die aufgaben-bezogene Zusammenführung von Anwendungsfunktionen im aktuellen Dialogkontext getroffen worden. Diese Einschränkung hat jedoch nur bei Dialogkontexten mit relativ wenigen Anwendungsfunktionen schwerwiegende Beeinträchtigungen des Benutzers zur Folge. Je größer das Ausmaß an anwendungsbezogener Flexibilität für die Oberfläche in dem hier vorgestellten Sinne ist, desto kleiner ist die Wahrscheinlichkeit, daß eine vom Benutzer gerade benötigte Anwendungsfunktion nicht erreichbar ist.

Im Extrem-Fall wäre zu jedem Zeitpunkt alles vom Start-Dialog-Kontext aus machbar: ein "modeless-state" wäre erreicht. Da dies jedoch in dieser letzten Konsequenz nur bei einem recht eingeschränkten Anwendungsumfang möglich ist, ist eine entsprechende Analyse der pro Dialogkontext sinnvollen Zusammenstellung von Anwendungsfunktionalität weiterhin notwendig.

Wenden wir AA_nF auf unsere beiden Beispiele (Dialog-Baum und -Netz) an, so ergibt sich AA_nF = 0.86 ("baum", mit D = 14) und AA_nF = 0.92 ("netz", mit D = 13) bei insgesamt 12 anwendungs-bezogenen funktionalen interaktiven Aufsetzpunkten für beide Systeme. Es werden also bei beiden Systemen dem Benutzer im Durchschnitt weniger als ein anwendungs-bezogenen funktionalen interaktiven Aufsetzpunkt pro Dialogkontext angeboten.

Da dieses Maß AA_nF jedoch den gesamten interaktiven Aufwand für die dialog-bezogenen funktionalen interaktiven Aufsetzpunkte außer Acht läßt, soll dieser Aspekt im nächsten Maß seinen Niederschlag finden. Wir definieren uns das folgende Maß RAnF (Formel 7):

(Formel 7)

$$RAnF = 1/D \sum_{d=1}^D \#AFIAP_d / (\#DFIAP_d + \#AFIAP_d) * 100\%$$

RAnF relatives Maß für die *anwendungsbezogene Flexibilität*;
 #AFIAP_d Anzahl an anwendungs-bezogenen funktionalen interaktiven Aufsetzpunkten im Dialogkontext d;
 #DFIAP_d Anzahl an dialog-bezogenen funktionalen interaktiven Aufsetzpunkten im Dialogkontext d;
 D Anzahl aller vorhandenen Dialog-Kontexten DKo.

Für die baum-artige Dialogstruktur ist RAnF = 31%, und für die netz-artige Struktur ist RAnF = 32%. Dies bedeutet, daß das "Netz"-Beispiel ein Prozent mehr anwendungsbezogener Funktionalität dem Benutzer im Mittel pro Dialogkontext zur Verfügung stellt. Ein interaktives Programm ist immer dann besonders interaktiv direkt hinsichtlich seiner anwendungsbezogenen Flexibilität, wenn das Verhältnis zwischen Anzahl Anwendungsfunktionen und Anzahl Dialogfunktionen zugunsten der Anwendungsfunktionen ausfällt.

Sehen wir uns unsere beiden Dialog-Beispiele zum Schluß noch einmal in der Übersicht an (Tab. 2).

Tabelle 2. Übersicht über die Ergebnisse der Maße für die Dimension der repräsentationalen und der interaktionalen Direktheit angewendet auf die beiden Beispiele.

Dialog-Struktur	AReDi	RReDi	HG	DiAI	AAAnF	RAnF
baum-artig	2.3	100%	3.7	1.00	0.86	31%
netz-artig	2.8	95%	3.6	1.58	0.92	32%

Um unsere beiden - in vielerlei Hinsicht sehr ähnlichen - Beispiele bezüglich ihrer repräsentationalen und interaktionalen Direktheit unterscheiden zu können, sind die Maße RReDi, HG und RAnF für die repräsentationale, bzw. interaktionale Direktheit am besten geeignet. Dies sind auch in der Tat diejenigen Unterschiede, welche zwischen beiden Oberflächen-Strukturen am deutlichsten zur Geltung kommen.

3.3 Anwendungskomponente und funktionale Direktheit (FD)

In diesem Abschnitt geht es um die Gestaltung der Anwendungskomponente hinsichtlich der aufgabenbezogenen Funktionalität. Eine der wichtigsten und eine notwendige Voraussetzung für aufgaben-orientierte Softwaregestaltung ist die Vollständigkeit der Funktionalität. Dieser Bewertungsaspekt hat daher auch seinen Niederschlag in den verschiedenen Bewertungs- und Gestaltungskonzepten gefunden:

- Funktionserfüllung (BECKER, HABERFELLNER & LIEBETRAU, 1990);
- Vollständigkeit der Funktionen (ZEHNDER, 1986);
- Aufgabenangemessenheit (VDI 5005 Entwurf, 1988).

Es ist unabdingbar, daß die Anwendungsfunktionalität auch tatsächlich für die gegebene Aufgabenbearbeitung vollständig vorhanden ist. Da dies nicht immer gewährleistet sein muß, wird das Maß der *funktionalen Vollständigkeit* eingeführt.

Um die funktionale Vollständig quantitativ angeben zu können, muß einerseits zwischen dem funktionalen Leistungsumfang laut Pflichtenheft (als Ergebnis der Mensch-Computer-Funktionsteilung) und andererseits der tatsächlich implementierten Funktionalität unterschieden werden. Diese Aufteilung ist jedoch nicht immer strikt gegeben, weil sie in das Spannungsfeld des kommunikativen Problembereiches der Anwender-Entwickler-Kommunikation fällt. Hier können Verfahren zum Benutzereinbezug in den Entwicklungsprozeß selbst, unterstützt durch Prototyping, etc., Abhilfe schaffen (SPINAS, WAEBER & STROHM, 1990).

Wir gehen davon aus, daß die benötigte Funktionalität im Pflichtenheft, bzw. in der Dokumentation vollständig beschrieben vorliegt. Ist dies der Fall, so läßt sich die *funktionale Vollständigkeit* einfach als prozentuales Verhältnis zwischen vorgesehener und implementierter Funktionalität berechnen (Formel 8); vorausgesetzt natürlich, daß die Angemessenheit der vorhandenen Anwendungsfunktionalität in der Phase der Mensch-Computer-Funktionsteilung entsprechend den handlungstheoretisch begründeten Kriterien zur "Aufgabenorientierung" (Abb.2) berücksichtigt worden ist.

Das Maß FuVo kann auch Werte über 100% annehmen, und zwar immer dann, wenn einzelne Funktionen laut Pflichtenheft mit einem höheren Auflösungsgrad ($\#AFIAP > \#Pf(AFIAP)$) implementiert worden sind. Dieser eventuell nachteilige Aspekt schlägt jedoch dann bei den Maßen zur interaktionalen Direktheit zu Buche. Natürlich sollte dieses Maß FuVo stets mindestens zu 100% erfüllt sein; leider hat es sich in der Praxis gezeigt, daß dies nicht immer der Fall ist (MELZER, 1989; S. 117).

(Formel 8)

$$FuVo = \#AFIAP / \#Pf(AFIAP) * 100\%$$

FuVo	absolutes Maß für die <i>funktionale Vollständigkeit</i> ;
#AFIAP	Anzahl an interaktiven Aufsetzpunkten bezüglich der Anwendungs-Funktionen, wie sie insgesamt in der zu bewertenden Implementation vorliegen;
#Pf(AFIAP)	Leistungsumfang der Anwendungs-Funktionen, wie sie in der Mensch-Computer-Funktionsteilung im Pflichtenheft vor der Implementierung festgelegt worden sind.

Um zu einem quantitativen Maß für das Kriterium "Flexibilität" - bezogen auf die Anwendungsfunktionen - zu gelangen, haben wir oben bei den Maßen AAnF und RAnF die Anzahl an *direkt zugänglichen Anwendungsfunktionen* pro Dialog-Kontext betrachtet.

Das Kriterium der "Flexibilität" läßt sich nun weiterhin auf die Gestaltung der Anwendungskomponente in soweit umsetzen, als es sich um den Aspekt der "*Permutation*" der einzelnen anwendungs-objekt bezogenen Operatoren handelt. Worum geht es hierbei?

Der Benutzer hat zur Erledigung seiner jeweiligen Aufgaben das Problem, für ein ausgewähltes Anwendungsobjekt ausgehend von dessen Ausgangszustand im Bezug auf einen gesetzten Zielzustand die adäquate Abfolge der zur Verfügung stehenden anwendungs-objekt bezogenen Operatoren auszuwählen. Wenn der Benutzer diese Abfolge der einzelnen anwendungs-objekt bezogenen Operatoren nun in einer permutierten Reihenfolge vornehmen kann, er also unterschiedliche Wege zur Zielerreichung beschreiten kann, dann zeichnet sich das interaktive System durch einen hohen Grad an *aufgabenbezogener Flexibilität* aus. Verdeutlichen wir diesen Aspekt der Permutation von anwendungs-objekt bezogenen Operatoren an einem einfachen Beispiel.

In einem Textverarbeitungssystem sei dem Benutzer die Anwendungsfunktion "Erstellen eines Inhaltsverzeichnisses" gegeben. In den meisten Textverarbeitungssystemen (z.B. MS-WORD) macht diese Anwendungsfunktion jedoch tatsächlich erst am Ende der Erstellung eines längeren Textes einen Sinn. Dies liegt darin begründet, daß die relevanten Eigenschaften (Seitenzahlen, Art und Anzahl der Kapitelüberschriften) erst am Ende vollständig vorliegen, sodaß das System nur noch diese Angaben aus dem Text herausucht und zu dem Inhaltsverzeichnis zusammenträgt. Wird diese Anwendungsfunktion zu einem früheren Zeitpunkt ausgelöst, so läuft der Benutzer Gefahr, daß durch seine weiteren Bearbeitungsschritte in dem Text, das Inhaltsverzeichnis rasch veraltet und er es vor dem Ausdruck erneut aktualisieren muß.

Ein anderes Textverarbeitungssystem, welches jedoch die aktuellen Angaben (Seitenzahl, Kapitelnummer, etc.) automatisch selbstständig aktualisiert, nachdem irgend wann einmal der Benutzer seinen Wunsch nach einem Inhaltsverzeichnis dem System mitgeteilt hat, wäre somit flexibler, bezogen auf die konkrete Aufgabenbearbeitung. Die grobe Bearbeitungsstruktur "Erstellen", "Bearbeiten", "Korrektur", "Inhaltsverzeichnis" und "Ausdrucken" könnte nun auch wie folgt aussehen: "Erstellen", "Bearbeiten", "Inhaltsverzeichnis", "Korrektur" und "Ausdrucken". Die beiden Bearbeitungsschritte "Korrektur" und "Inhaltsverzeichnis" wären permutabel. Eine derartige Vertauschung von "Korrektur" und "Inhaltsverzeichnis" soll *Transposition* heißen.

Ein Textverarbeitungssystem, welches dem Benutzer gegenüber den Zeitpunkt für das Anlegen des Inhaltsverzeichnisses offen läßt, aber dann eine entsprechende automatische Aktualisierung *nicht* durchführt und auch vor dem Ausdrucken nach einer weiteren Textkorrektur keinen Hinweis auf das veraltete Inhaltsverzeichnis ausgibt, wäre somit "pseudo-flexibel" und damit eher verwirrend.

Zur quantitativen Messung des Aspektes der "*Permutation*" gehen wir davon aus, das eine vollständige Liste aller benötigten Anwendungsfunktionen vorliegt. Greifen wir wieder unser Beispiel des Dialog-Netzes (Abb. 4) auf. Die Menge der Buchstaben {a, b, ..., l} symbolisiert die vorhandenen Anwendungsfunktionen, im folgenden durch ihren anwendungs-bezogenen funktionalen interaktiven Aufsetzpunkt gekennzeichnet. "Permutation" im kombinatorischen Sinne heißt, gegeben sind n verschiedene Anwendungsfunktionen; wie

viele Möglichkeiten gibt es, diese Anwendungsfunktionen in verschiedener Weise anzuordnen? Diese Möglichkeit wird als "Permutation von n Elementen" bezeichnet.

Wären alle anwendungs-bezogenen funktionalen interaktiven Aufsetzpunkte vollständig permutabel, so ließe sich dieses Ausmaß an Flexibilität einfach durch die Fakultät: (#AFIAP)! angeben. Für unser einfaches Beispiel ergäbe sich: $12! = 12 \cdot 11 \cdot 10 \cdot \dots \cdot 2 \cdot 1 = 479.001.600$, eine zugegeben sehr hohe Zahl an Permutationen. Dieses Ausmaß an Flexibilität wird jedoch durch die *kausalen Abhängigkeiten* zwischen den einzelnen, anwendungsbezogenen Funktionen stark eingeschränkt. Es lassen sich somit nur einzelne Klassen von Anwendungsfunktionen ausmachen, innerhalb derer die einzelnen Funktionen variabel sind. Wie lassen sich jedoch diese Klassen finden?

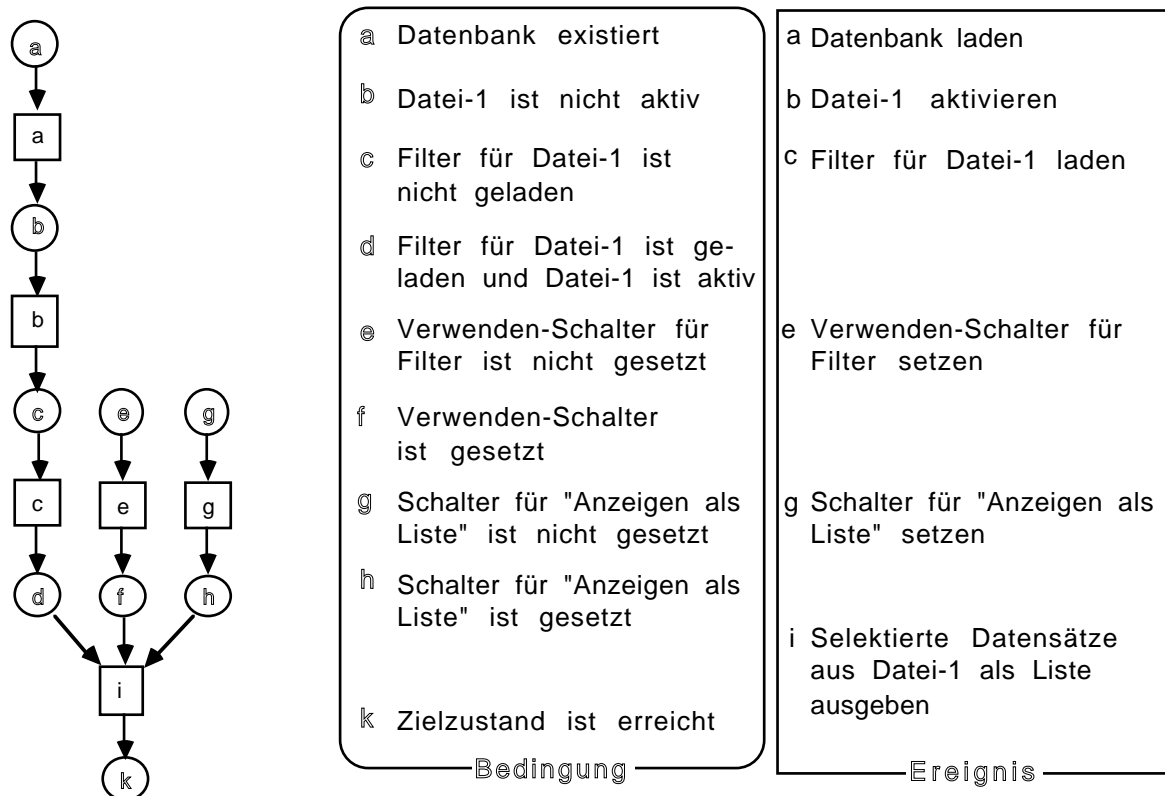


Abb. 9. Kausaler Abhängigkeitsgraph als Bedingungs-Ereignis-Netz für Teile der Anwendungsfunktionen eines Datenbankprogrammes. Die hier angegebenen kausalen Abhängigkeiten sind im Text beispielhaft erläutert. Die Kreise bedeuten Bedingungen und die Quadrate bedeuten Operationen. Alle parallelen Operationen, bzw. Aktionsgruppen ($[a,b,c] \parallel [e] \parallel [g]$) sind voll permutabel.

Um die kausalen Abhängigkeiten zwischen den einzelnen Anwendungsfunktionen veranschaulichen zu können, führen wir den *kausalen Abhängigkeitsgraphen* ein (Abb. 9). Diese kausalen Abhängigkeiten setzen sich aus zwei Aspekten zusammen: die systemtechnischen Abhängigkeiten und die objektive Kausalstruktur der Aufgabe. Der Abhängigkeitsgraph für die objektive Kausalstruktur der Aufgabe läßt sich z.B. durch eine arbeitspsychologische Aufgabenanalyse sinnvoll erstellen (TRIEBE, 1981; S. 136f.). Über diese aufgaben-immanenten kausalen Abhängigkeiten hinaus ergeben sich auch noch kausale Abhängigkeiten durch die konkrete Implementation der einzelnen Funktionen im realen Software-System. Ein kausaler Abhängigkeitsgraph (system- oder aufgaben-bezogen) wird als ein kreisfreies Petri-Netz mit unverzweigten S-Elementen ("Kreise") definiert.

Erläutern wir kurz an einem konkreten Beispiel die Brauchbarkeit der Bedingungs-Ereignis-Netze zur Darstellung des kausalen Abhängigkeitsgraphen für ein Datenbank-Management-System (DBMS). Die konkrete Implementation dieses DBMS gestattet es nun dem Benutzer drei verschiedene Handlungsstränge parallel und damit permutabel auszuführen (Abb. 9). Der Benutzer kann zwischen diesen drei Handlungssträngen beliebig hin- und her-

springen. Das konkrete Teil-Ziel eines Benutzers in diesem Beispiel besteht darin, eine bestimmte Menge an Datensätzen in einer Liste auf dem Bildschirm auszugeben. Die Selektion der einzelnen Datensätze erfolgt über einen schon definierten "Filter". Erst wenn alle Vor-Bedingungen erfüllt sind, kann die Ausgabe-Operation erfolgen. Da die Vor-Bedingung "d" für die Aktion "i" eine komplexe Bedingung ist, d.h. es sind mehrere vorbereitende Schritte zur Herstellung dieser Bedingung notwendig, müssen alle diese vorbereitenden Operationen (a, b, c) zu der Gruppe [a, b, c] zusammengefaßt werden. Erst diese Gruppe ist tatsächlich mit den anderen beiden Operationen [e] und [g] parallel und damit permutabel.

Durch diese kausalen Abhängigkeiten wird der permutable Aktionsbereich des Benutzers eingeschränkt. Wie stark diese Einschränkung nun tatsächlich ist, läßt sich quantitativ mit dem folgenden Maß RAuF bestimmen (Formel 9). Dazu wird die Anzahl der Gruppen mit einer festgelegten Kausalstruktur anhand einer Systemanalyse der vorliegenden, bzw. geplanten Software bestimmt. Zusätzlich wird anhand einer Aufgabenanalyse die Anzahl Gruppen mit einer aufgabenangemessenen Kausalstruktur bestimmt.

Wäre z.B. die Anzahl der Gruppen mit einer entsprechenden Kausalstruktur gleich eins ($\#AFIAP_s = 1$), so liegt eine ganz bestimmte Abfolge aller Anwendungsfunktionen vor, von welcher der Benutzer nicht abweichen darf. Gehen wir zunächst davon aus, daß die arbeitspsychologische Aufgabenanalyse einen voll permutablen Aktionsraum ergäben hätte, dann liefert RAuF für unser Dialog-Beispiel (Abb. 4) $RAuF = 1! / 12! * 100\% = 2 * 10^{-7} \%$. Wäre die Anzahl der Gruppen mit Kausalstrukturen für eine gegebene Systemimplementation aller Anwendungsfunktionen gleich der Anzahl Gruppen mit Kausalstrukturen einer Aufgabenanalyse ($\#AFIAP_s = \#AFIAP_a$), dann ist $RAuF = 100\%$. Ist RAuF größer als 100%, so sprechen wir von einem *pseudo-flexiblen* System.

(Formel 9)

$$RAuF = (\#G(AFIAP_s)! / \#G(AFIAP_a)!) * 100\%$$

RAuF	relatives Maß für die <i>aufgaben-bezogene Flexibilität</i> ;
$\#G(AFIAP_s)$	Anzahl an voll permutablen Gruppen von anwendungs-bezogenen funktionalen interaktiven Aufsetzpunkten, welche sich in einer kausalen Abhängigkeit zueinander befinden. Diese kausalen Abhängigkeiten werden durch die <i>Systemanalyse</i> eines gegebenen Systems, bzw. Systementwurfes bestimmt.
$\#G(AFIAP_a)$	Anzahl an voll permutablen Gruppen von anwendungs-bezogenen funktionalen interaktiven Aufsetzpunkten, welche sich in einer kausalen Abhängigkeit zueinander befinden. Diese kausalen Abhängigkeiten werden auf der Grundlage einer <i>Aufgabenanalyse</i> bestimmt.

Berechnen wir RAuF für den kausalen Abhängigkeitsgraphen aus unserem DBMS-Beispiel und gehen wir ebenfalls davon aus, daß eine Aufgabenanalyse einen voll permutablen Aktionsbereich für vier Operationen (b', c, e, g; mit $b'=ab$) ergäben hätte, während dagegen das System nur drei permutablen Operationen zuläßt (c', e, g; mit $c'=abc$), so ist $RAuF = 3! / 4! * 100\% = 25\%$. Es wären also 75% der aufgabenangemessenen Flexibilität durch den konkreten Systementwurf verloren gegangen. Dieses Maß RAuF bringt am deutlichsten zum Ausdruck, welche Einschränkungen Benutzer bei Systemen mit geringer Permutabilität hinnehmen müssen. Zusätzlich ist dieses Maß auch noch sehr sensibel gegenüber den tatsächlichen Einschränkungen; betrachten wir z.B. RAuF mit 65 voll-permutablen Systemgruppen gegenüber 66 voll-permutablen Aufgabengruppen, so bleiben nur noch 1.5% aufgabenbezogene Flexibilität übrig! Dies kann mit ein Grund dafür sein, warum Benutzer sehr empfindlich auf die Beschränkung ihrer individuellen Freiheitsgrade reagieren.

Eine letzte Anmerkung zu der hier vorgestellten Operationalisierung des Kriteriums der "Flexibilität" und zu der weiteren Verwendbarkeit der einmal aufgestellten kausalen Abhängigkeitsgraphen besteht darin, daß sich diese kausalen Strukturen sehr hilfreich für die *Gestaltung der Systemausgaben* heranziehen lassen können. Jeder Versuch eines Benutzers, eine System-Funktion auslösen zu wollen, für die noch nicht alle notwendigen Bedingungen

hergestellt worden sind, muß nicht mehr wie bisher frustrierend im Sande verlaufen oder auf mühseliges Trail & Error-Verhalten hinauslaufen, sondern kann durch entsprechende Systemmeldungen zielgerichtet unterstützt werden. Hiermit zeigen sich also die Bezüge zu dem Kriterium der "*Unterstützung*" (MOLL, 1989).

4 Zusammenfassung und Ausblick

Was haben wir nun erreicht? Wir sind von dem Konzept der benutzer-orientierten Softwaregestaltung ausgegangen und haben es mit den Komponenten eines interaktiven Systems ("IFIP-Modell") in Beziehung gesetzt ("Gestaltungsmatrix"). Hierdurch liessen sich systembezogene Gestaltungsschwerpunkte festlegen. Diese Gestaltungsschwerpunkte werden "repräsentationale", "interaktionale" und "funktionale" Gestaltungsdimensionen genannt. Um die verschiedenen Typen von Benutzungsoberflächen adäquat mit einheitlichen Begriffen beschreiben und damit dann auch vergleichen zu können, wurden verschiedene Begriffe eingeführt und überwiegend klar definiert.

Der wichtigste Begriff ist sicherlich der des "interaktiven Aufsetzpunktes", auch mit IAP abgekürzt. Dieser Begriff soll die verschiedenen Arten von Interaktionstechniken und -arten gleichermaßen unter sich vereinen, um somit auch typ-übergreifende Vergleiche zwischen Oberflächen anstellen zu können. Die Motivation hierzu ist darin begründet, experimentelle Befunde zwischen verschiedenen Oberflächentypen erklärend beschreiben zu können. Für jede der drei Gestaltungsdimensionen nun wurden verschiedene quantitative Maße entwickelt und an zwei fiktiven, einfachen Beispielen erläutert und durchgerechnet.

Für die Dimension der *repräsentationalen Direktheit* wurde das Kriterium "Feedback" in den Maßen A-, R- und WAR-ReDi operationalisiert. Die besondere Schwierigkeit dieser Gestaltungsdimension liegt darin begründet, daß die Semantik des Wahrgenommenen primär durch den individuellen Wissensbestand bedingt ist, und sich nicht generell angeben läßt. Dies ist auch ein zwingender Grund mit dafür, Benutzer in den Erstellungsprozeß von interaktiver Software möglichst frühzeitig mit einzubeziehen.

Für die Dimension der *interaktionalen Direktheit* wurde das Kriterium "Flexibilität" in den Maßen HG, DiAl, AAnF und RAnF operationalisiert. Diese vier Maße sind unterschiedlich gut geeignet, die verschiedenen Aspekte der interaktiven Flexibilität messen zu können. Am brauchbarsten sind sicherlich HG für den mittleren Hierarchisierungsgrad der Dialogstruktur und RAnF für die durchschnittliche Anzahl an Anwendungsfunktionen pro Dialogkontext.

Für die Dimension der *funktionalen Direktheit* wurde ebenfalls das Kriterium "Flexibilität" in dem Maß RAuF und das Kriterium "Vollständigkeit" in dem Maß FuVo operationalisiert. Flexible Systeme in einem aufgaben-bezogenen Sinne zeichnen sich dadurch aus, daß sie den permutatorischen Raum aller kausal möglichen Abfolgen von Bearbeitungsschritten innerhalb einer Aufgabe zu erhalten versuchen. Unter dem Aspekt der Vollständigkeit ist hier zunächst nur die Vollständigkeit des vor der Implementation eines interaktiven Systems festgelegten Leistungsumfanges in dem nachher erstellten System gemeint. Die Vollständigkeit in einem handlungstheoretischen Sinne bleibt weiterhin Gestaltungsziel von detaillierten Aufgabenanalysen.

Die Maße sind so system-nah ausgerichtet, daß sich auch konkrete Gestaltungsmaßnahmen aus ihnen ableiten lassen. Somit ist es möglich geworden, nicht nur retro-spektiv, sondern auch pro-spektiv gestaltend tätig zu werden. Zusätzlich sind jedoch noch folgende Dinge zu tun: für alle Bereiche, in denen die Semantik von ausschlaggebender Bedeutung ist, sind Gestaltungsregeln mit konkreten Beispielen zu erstellen. Erst wenn sich Grenzwerte für die einzelnen quantitativen Maße empirisch finden lassen, kann ein derart quantitativ ausgerichtetes Bewertungs- und Gestaltungssystem seine volle Kraft entfalten.

5 Literaturverzeichnis

- BAITSCH C, KATZ C, SPINAS P, ULICH E (1989) Computerunterstützte Büroarbeit: ein Leitfaden für Organisation und Gestaltung. Verein der Fachverlage, Zürich
- BECKER M, HABERFELLNER R, LIEBETRAU G (1990) EDV-Wissen für Anwender. Verlag Industrielle Organisation, Zürich
- CORDES R (1988) Zur Konzeption benutzerspezifischer Arbeits- und Retrievalumgebungen in modernen Informationssystemen. Naturwissenschaftliche Dissertation, Technische Universität Braunschweig
- DUDEN (1963) Herkunftswörterbuch: die Etymologie der Sprache. Bibliographisches Institut, Mannheim Wien Zürich
- DZIDA W (1982) Dialogfähige Werkzeuge und arbeitsgerechte Dialogformen. In: TAUBER M, SCHAUER H (Hrsg.) Informatik und Psychologie. Oldenbourg, München Wien
- DZIDA W (1987) On tools and interfaces. In: FRESE M, ULICH E, DZIDA W (eds.) Psychological Issues of Human Computer Interaction in the Work Place. North-Holland, Amsterdam, pp 339-355
- FABIAN F (1986) Fenster- und Menüsysteme in der MCK. In: FISCHER G, GUNZENHÄUSER R (Hrsg.) Methoden und Werkzeuge zur Gestaltung benutzergerechter Computersysteme. Walter de Gruyter, Berlin New York, S 101-119
- GREUTMANN T, ACKERMANN D (1987) Individual differences in human-computer-interaction: how can we measure if the dialoggrammar fits the user`s needs? In: BULLINGER H-J, SHACKEL B (eds.) Human-Computer-Interaction INTERACT`87. Elsevier-North Holland, Amsterdam , pp 145-149
- IFIP (1981) Report of the 1st Meeting of the European User Environment Subgroup of IFIP WF 6.5. Gesellschaft für Mathematik und Datenverarbeitung, Bonn
- MELZER W (1989) Psychologische Grundlagen der Gestaltung der Mensch-Rechner-Interaktion. Psycholog. Dissertation, Technische Universität Dresden
- MOLL T (1989) Unterstützung von Softwarebenutzern: Aufgaben versus systemorientierte Lern- und Arbeitshilfen. Psycholog. Dissertation, Universität Bern
- MOSIER J N, SMITH S L (1986) Application of guidelines for designing user interface software. Behaviour and Information Technology 5(1):39-46
- NIEVERGELT J (1982) Errors in dialog design and how to avoid them. Proceedings International Zurich Seminar on Digital Communication, IEEE:199-205
- NORMAN D A (1983) Design rules based on analyses of human errors. Communications of the ACM 26(4):254-258
- PAYNE S J, GREEN T R G (1986) Task-Action Grammars: A Model of the Mental Representation of Task Languages. Human-Computer Interaction 2:93-133
- RAUTERBERG M (1989) Formale Modelle zur Beschreibung der Mensch-Computer Interaktion. unveröffentlichter Bericht, Fachbereich Informatik, Universität Oldenburg.
- SMITH S L (1986) Standards versus guidelines for designing user interface software. Behaviour and Information Technology 5(1):47-61
- SPINAS P (1987) Arbeitspsychologische Aspekte der Benutzerfreundlichkeit von Bildschirmsystemen. Psycholog. Dissertation, Universität Bern
- SPINAS P (1990) Benutzerfreundlichkeit von Dialogsystemen und Benutzerbeteiligung bei der Softwareentwicklung. In: FREI F, UDRIS I (Hrsg.) Bild der Arbeit. Hans Huber, Bern, S 158-171
- SPINAS P, WAEBER D, STROHM O (1990) Kriterien benutzerorientierter Dialoggestaltung und partizipative Softwareentwicklung: eine Literaturlaufarbeitung. In: SPINAS P, RAUTERBERG M, WAEBER D, STROHM O, ULICH E (Hrsg.) Projektberichte zum Forschungsprojekt Benutzer-orientierte Softwareentwicklung und Schnittstellengestaltung. Institut für Arbeitspsychologie, ETH Zürich
- SPINAS P, TROY N, ULICH E (1983) Leitfaden zur Einführung und Gestaltung von Arbeit mit Bildschirmsystemen. Industrielle Organisation, Zürich
- STREITZ N, LIESER A, WOLTERS A (1989) The combined effects of metaphor worlds and dialogue modes in human-computer-interaction. In: KLIX F, STREITZ N, WAERN Y, WANDKE H (eds.) Man-Computer Interaction Research MACINTER-II. Elsevier (North-Holland), Amsterdam, pp 75-88
- TAUBER M J, SCHIELE F (1986) Formale Spezifikation von Benutzerschnittstellen am Vergleich verschiedener Editoren. SEH`86. (Software Ergonomie Herbstschule, Berlin, 13.-16. Oktober 1986, pp 241-253)
- TAVOLATO P, WEIDMANN K-H (1989) Theoretische und empirische Untersuchungen der Benutzeroberfläche von zwei CAD-Systemen. In: MAASS S, OBERQUELLE H (Hrsg.) Software-Ergonomie `89. Teubner, Stuttgart (German Chapter of the ACM Berichte, Bd 32, S 447-456)
- TRIEBE J (1981) Aspekte beruflichen Handelns und Lernens. Psycholog. Dissertation, Universität Bern
- ULICH E (1985) Einige Anmerkungen zur Software-Psychologie. Sysdata 10:53-58
- ULICH E (1986) Aspekte der Benutzerfreundlichkeit. In: REMMELE W, SOMMER M (Hrsg.) Arbeitsplätze morgen. Teubner, Stuttgart (Berichte des German Chapter of the ACM, Bd 27, S 102-121)
- ULICH E (1987) Zur Frage der Individualisierbarkeit von Arbeitstätigkeiten unter besonderer Berücksichtigung der Mensch-Computer-Interaktion. Zeitschrift für Arbeits- und Organisationspsychologie 31:86-93

- ULICH E (1988) Arbeits- und organisationspsychologische Aspekte. In: BALZERT H, HOPPE H U, OPPERMAN R, PESCHKE H, ROHR G, STREITZ N (Hrsg.) Einführung in die Software-Ergonomie. Walter de Gruyter, Berlin New York, S 49-66
- ULICH E (1989) Arbeitspsychologische Konzepte der Aufgabengestaltung. in: MAASS S, OBERQUELLE H (Hrsg.) Software Ergonomie `89. Teubner, Stuttgart (German Chapter of the ACM Berichte, Bd 32, S 51-65)
- ULICH E (1991, im Druck) Arbeitspsychologie. Poeschel, Stuttgart
- ULICH E, RAUTERBERG G W M, MOLL T, GREUTMANN T, STROHM O (1991, in preparation) Task orientation and user-oriented dialog design. International Journal of Human-Computer-Interaction
- VDI-Entwurf 5005 (1988) Bürokommunikation - Software-Ergonomie in der Bürokommunikation. Verein Deutscher Ingenieure, VDI-Gesellschaft, Graf-Recke-Straße 84, D-4000 Düsseldorf 1
- ZACHMANN T (1989) DSL - eine Sprache zur Spezifikation und Implentation von Dialogen. Hannes Gertner, München
- ZEHNDER C A (1986) Informatik - Projektentwicklung. Verlag der Fachvereine, Zürich