



Whitepaper zur Literaturverwaltung

Liabulo

„Software-Ergonomie“
von Maxim Bauer

erstellt im Rahmen der Projektgruppe
„Virtuelle Organisation - Open Source“

unter der Leitung von
Prof. Uwe Schneidewind und Hendrik Eggers (Lehrbeauftragter)

Inhaltsverzeichnis

1	Einführung	1
2	Formen der Mensch-Computer-Interaktion	3
2.1	Gestaltung der Ein-/Ausgaben	3
2.1.1	Ausgabeelemente	3
2.1.2	Eingabeelemente	5
2.1.3	Techniken der MCI	5
2.1.4	Masken und Formulare	8
3	Individualisierung von Benutzungsschnittstellen	10
4	Aufgabenorientierte Systemgestaltung	12
4.1	Perspektiven und Methoden der Aufgabenanalyse	12
4.2	Beschreibungsmerkmale von Aufgaben	13
4.3	Analyse kognitiver Aufgaben	14
5	Fehlermanagement und Hilfesysteme	15
5.1	Grundlagen und Problemfelder	15
5.1.1	Der menschliche Handlungsprozess	15
5.1.2	Fehlhandlungen	16
5.2	Fehlervermeidung	17
5.3	Fehlermanagement	18
5.4	Hilfesysteme	19

Kapitel 1

Einführung

Die Software-Ergonomie beschäftigt sich seit ihren Anfängen mit der menschengerechten und Aufgaben angemessenen Gestaltung der *Mensch-Computer-Interaktion (MCI)*. Informationsdarstellung auf Bildschirmen und ihre Manipulation durch Eingabegeräte in verschiedenen Interaktionsformen sind zwar ein Gegenstand der Software-Ergonomie, aber auch die dahinter liegende Funktionalität. Benutzergerechte Gestaltung der Software berücksichtigt im umfassenden Sinn vier Aspekte [Obe91]:

- **Menschengerechte Gestaltung:** die Benutzer als Menschen haben ihre Stärken, Schwächen und Bedürfnisse, sowie auch unterschiedliche Entwicklung, und müssen deshalb berücksichtigt werden. D.h. es werden Kenntnisse über menschliche Wahrnehmung, Denken und Problemlösung sowie über Lernen, Kommunikation und Kooperation benötigt, damit die Software menschengerecht gestaltet werden kann.
- **Aufgaben angemessene Gestaltung:** da die Computer für bestimmte Aufgaben eingesetzt werden, sollten diese Aufgaben und das Verhalten des Benutzer bei der Arbeit genauestens bekannt sein, um menschengerechte Software zu entwickeln [Vol93].
- **Technikbewusste Gestaltung:** Die technisch verfügbaren Möglichkeiten sollten so gut es geht ausgenutzt werden.
- **Organisationsgerechte Gestaltung:** Durch die wachsende Verbreitung und Vernetzung der Computer, betrieblich wie privat, ist die Mensch-Computer-Interaktion nicht mehr auf die Einzelbenutzer und ihre Aufgaben beschränkt, sondern schließt auch die organisatorische Einbindung mit ein.

Bedeutung der Software-Ergonomie

Die Software-Ergonomie im weitesten Sinne bedeutet, dass dem Benutzer auf seinem Computer ein oder mehrere Anwendungssoftware-Systeme zur Verfügung stehen, die eine ergonomische, d.h. auf seine Person bezogene, menschengerechte Lösung seiner Aufgaben erlauben.

Im engeren Sinne bedeutet der Begriff der Software-Ergonomie, dass die Dialogstruktur und die Ein-/Ausgaben eines Software-Systems nach ergonomischen Richtlinien gestaltet sind. In beiden Fällen ist Wissen über den jeweiligen Endbenutzer, über seine zu lösenden Aufgaben und das organisatorische Umfeld erforderlich, um die Kriterien zu erfüllen.

Kapitel 2

Formen der Mensch-Computer-Interaktion

Die Formen der MCI wurden stets einem Wandel unterworfen. Dieser Wandel zeigt sich auf unterschiedlichsten Ebenen. Auf der Ebene der Ein-/Ausgabegeräte sind neue Formen möglich geworden wie z.B. Maus oder Stift. Die traditionellen Interaktionsformen wie Kommandodialoge und Menüeingaben wurden langsam durch die „direkte Manipulation“ ergänzt. Graphische Benutzungsschnittstellen (**GUIs**= Graphical User Interfaces) gewannen mittlerweile stark an Bedeutung und sind sehr verbreitet. Anwendungen, die auf graphischen Modellen beruhen wie z.B. 3D-Darstellungen von Informationen, Video- und Audio-Daten und die Einbeziehung von Hand- und Körperbewegungen zur Navigation in künstlichen Welten eröffnen neue Möglichkeiten [Bal88].

2.1 Gestaltung der Ein-/Ausgaben

Die Ein-/Ausgabeelemente werden in der Regel in drei verschiedene Gruppen unterteilt: Die *visuellen*, *akustischen* und *haptischen* Elemente. Wobei den visuellen Elementen die entscheidende Rolle zugeordnet wird und die akustische sowie haptische Elemente eher eine untergeordnete Rolle spielen und werden hier nicht weiter behandelt.

2.1.1 Ausgabeelemente

Visuelle Elemente

Alle visuellen Ausgaben auf einem Bildschirm sind 2-dimensionaler Natur, da sie aus Rasterpunkten zusammengesetzt sind. Wichtige Einheiten sind ent-

weder *Symbole* (Schriftzeichen eines Alphabets, Bildsymbole = Icons) oder *bildliche Darstellungen* (Rahmen, Bilder, Linien, etc.) von Modellkomponenten. Einige der bildlichen Darstellungen können mit Blick auf die Bearbeitung mit speziellen Bedeutungen versehen sein, z.B. können Sie **Buttons** darstellen, die gedrückt werden können.

Bei der Gestaltung bzw. Auswahl von Schriften sollten Texte in Versalien (nur Grossbuchstaben) vermieden werden, genauso wie Schriften ohne Unterlängen, da sie das flüssige Wahrnehmen von Wortsymbolen verhindern. Bekannte Schriften mit Serifen (kleine Häkchen) unterstützen dagegen das sichere Lesen. Schriftgrösse, Farbe, Stil und Helligkeit sind weitere Merkmale, deren Auswahl eine wichtige Rolle bei der Entwicklung der Oberflächen spielt. Das 2-dimensionale Layout von Texten (z.B. Absätze, Blöcke, Spalten) ist ein weiteres Gestaltungsfeld, bei dem die menschliche Gestaltwahrnehmung berücksichtigt werden muss. Die 2D-Anordnung von Punkten, Linien und Flächen ist geeignet, absolute und relative Quantitäten, Ordnungen, Sorten und Beziehungen auszudrücken. Durch Nutzung der anderen graphischen Variablen wie z.B. *Grösse, Helligkeit, Muster, Richtung, Farbe* und *Form* kann eine Verstärkung erreicht werden. Die relative räumliche Lage der graphischen Elemente drückt in der Regel eine inhaltliche Beziehung zwischen den dargestellten Komponenten aus [Mar92].

Die am häufigsten eingesetzte Fenster sind Visualisierungsinstrumente, mit denen unter Benutzerkontrolle relevante Ausschnitte des Systemzustandes permanent angezeigt und interaktiv bearbeitet werden können. Das sind in der Regel Kombinationen von graphischen Ausgabeelementen wie z.B. eine Titelanzeige, eine Reihe von Buttons für das Schließen, Vergrössern/Verkleinern, einen Rollbalken für die Wahl des Abschnitts und Ähnlichen. Die konsistente visuelle Darstellung ähnlicher Information in ähnlichen Formen an erwarteten Positionen unterstützt die Herausbildung von Wahrnehmungsgewohnheiten und damit die schnelle und sichere Orientierung an der Benutzungsschnittstelle.

Akustische Ausgabe

Bei der akustischen Ausgabe wird zwischen Sprach- und Klang- bzw. Geräusch-Ausgabe unterschieden. Die Sprachausgabe erlaubt es beliebige Texte hörbar auszugeben und wird in erster Linie dazu verwendet, um die besondere Aufmerksamkeit zu erzeugen. Oder auch um die mit den Augen anderweitig beschäftigten eine Wahrnehmungsalternative zu bieten.

Die Ausgabe von Klängen oder Geräuschen kann Symbolcharakter haben. Hörbare Symbole werden als eine Verstärkung der visuellen Ausgabe verwendet. Der Nachteil der akustischen Ausgaben liegt in der Beeinträchtigung der anderen Benutzer in selben Raum, was sich eventuell auf deren Arbeitsleistung auswirkt. Deshalb sollten die visuellen Ausgaben abschaltbar sein, ohne dass die Interaktion mit dem Benutzer gestört wird.

2.1.2 Eingabeelemente

Als Eingabeelemente sind die Tastaturen und Zeigeinstrumente die traditionellen Geräte und sind am häufigsten verbreitet. Z.B. eine Maus als Zeigeinstrument hat wie jedes Eingabegerät ihre Handhabungsvor- und nachteile. Sie erlaubt eine entspannte Hand- und Armhaltung, hat eine relativ hohe Positioniergeschwindigkeit und die Bedienung ist leicht erlernbar. Dagegen erfordert die Nutzung einer Maus Platz auf einer waagerechten Fläche, sie wirkt nur indirekt über den Zeiger und ist weniger zum Zeichnen oder Malen geeignet.

Tasten sind die Eingabeelemente mit deren Hilfe bestimmte Zeichen eines festen Alphabets eingegeben werden können. Die Zeichen sind nicht notwendigerweise Buchstaben, es können auch andere Bildsymbole sein. Abstrahiert können die Tastaturen als Quellen von Eingabesymbolen und Symbolketten aus einem festem Alphabet betrachtet werden, die der Beschreibung von Angaben dienen. Über die Eingabe von Symbolen hinaus, können mit Hilfe der Zeigeinstrumente auch graphische Eingaben erzeugt werden.

2.1.3 Techniken der MCI

Grundsätzlich werden mehrere Standardformen der MCI unterschieden. Darunter sind die Kommandodialoge, Frage-/Antwort-Dialog, Menü-Techniken und direkte Manipulation gemeint. Um festzustellen, welche Techniken zur Erzeugung benötigter Angaben besonders geeignet bzw. ungeeignet sind, werden die einzelnen Techniken genauer betrachtet.

Kommandodialogen

Unter Kommandodialogen wird die Form der Kommandoingabe bzw. -ausgabe in Textform, und die dafür verwendete Kommandosprache verstanden. Eine Kommandosprache ist eine formale Sprache zur Formulierung von Anweisungen an den Computer. Für jede Anweisung werden die benötigten Angaben, wie Operation, Argumente, Bezugsobjekt und etc., in Textform

angegeben. Kommandosprachen können mit Elementen von Programmiersprachen angereichert werden, um komplexe Kommandos zu bilden [Shn92].

Vorteile der Kommandosprachen:

- Keine speziellen Eingabegeräte erforderlich
- Kommandosprachen sind universell hinsichtlich der Ausdrucksmächtigkeit
- Erlauben leichte Aufzeichnung von Vergangenheit (Protokoll) und Zukunft (Pläne) der Interaktion
- Erlauben effiziente Formulierung von Anweisungen (Expertenwissen vorausgesetzt)

Nachteile der Kommandosprachen:

- Die Sprache muss erlernt werden
- Schnelle und fehlerfreie Eingabe von Zeichenketten ist erforderlich (flüssige Interaktion), dadurch höherer Lernaufwand
- Der Wirkungsbereich von Kommandos kann den gesamten Zustand des interaktives Systems beeinflussen – die Wirkungen eines Kommandos sind schwer abzuschätzen
- Für Anfänger und gelegentliche Benutzer ungeeignet (hoher Lern- und Erinnerungsaufwand)

Menüs

Menüs sind wahrnehmbare Darstellungen alternativ möglicher Angaben, aus denen der Benutzer eine Auswahl treffen kann. Die Gestaltungsform eines Menüs kann von *einfachen Textform* bis zu einer *Menü-Leiste mit Untermenü als Pulldown-Menü* variieren. Bei den einzelnen Angaben eines Menüs handelt es sich immer um eine beschränkte Anzahl von Elementen, die in Textform oder bildlich dargestellt werden können. Bei hierarchischen werden häufig mindestens zwei Ebenen dargestellt, um den Kontext der aktuellen Auswahl erkennen zu können. Für die graphische Darstellung von Menüs mit Einfach- oder Mehrfachauswahl werden zunehmend **Buttons** verwendet, die direkte Manipulation erlauben und mit Menü-Angaben beschriftet werden [Shn92].

Die Platzierung von Menüs kann sowohl an einer festen Stelle des Bildschirms, bzw. an festen Stellen von Fenstern oder in der Nähe von aktuell bearbeiteten Objekten erfolgen. Eine einheitliche Platzierungsstrategie für einen kompletten Arbeitsplatz ist empfehlenswert, weil dies die Ausprägung von Wahrnehmungs- und Handhabungsgewohnheiten unterstützt.

Darüber hinaus sollte der Zeitraum, in dem ein Menü sichtbar ist, festzulegen. Dauernd wählbare Optionen sollten auch dauernd sichtbar bleiben, nur aktuell verfügbare Optionen nur so lange, bis eine Auswahl getroffen ist und der Benutzer das Feedback über die getroffene Wahl zur Kenntnis genommen haben kann.

In graphischen Oberflächen haben sich mehrere Darstellungsformen herausgebildet, die als „pull-down“- , „popup“- bzw. „pin-down“-Menüs bezeichnet werden.

- **Pull-down-Menüs** machen die Optionen eines Menü-Eintrags aus einer Menü-Leiste sichtbar, solange eine Taste des Zeigeelements gedrückt ist und der Cursor sich im Darstellungsbereich des Untermenüs befindet. Die Position der Cursors in der Menü-Darstellung beim Loslassen der Taste bestimmt, welche Option gewählt wird. Diese Auswahl sollte in der Regel durch Hervorhebung der aktuellen Wahl (z.B. durch Einfärben) unterstützt.
- **Pop-up-Menüs** erscheinen dynamisch nach Betätigung einer speziellen Menü-Darstellungstaste des Zeigeelements in der Nähe von ausgewählten Objekten oder nahe dem Cursor. Sie verschwinden nach getroffener Auswahl wieder.
- **Pin-down-Menüs** sind das gleiche wie „Pop-up-Menüs“, nur dass das Menü explizit dauerhaft sichtbar gemacht werden kann („Anpinnen“).

Die Zahl der zu wählenden Angaben ist meistens eins, kann aber auch mehrere Angaben umfassen, z.B. wenn in einem Menü Alternativen für verschiedene Textattribute angeboten werden. Zur Erhöhung der Verständlichkeit sollte in jedem Menü an seiner äußeren Form erkennbar sein, wie viele Auswahlen zu treffen sind und welches die jeweiligen, sich gegenseitig ausschließenden Alternativen sind.

Als Auswahltechniken für Menü-Optionen sind das Zeigen auf die Darstellung der Option oder die Tastatureingabe üblich. Bei der Tastatureingabe werden meistens folgende Auswahl-Codes verwendet:

- laufende Nummer
- laufender Buchstabe
- mnemotechnische Kürzel

Aus Benutzersicht ist es wichtig, dass solche Codes leicht und fehlerfrei eingegeben und mit dem Inhalt gut assoziiert werden können. Sie sollten durch die Menü-Darstellung leicht ersichtlich sein.

Vorteile von Menüs

1. Entlasten das Gedächtnis, weil der Benutzer die Alternativen nur erkennen und nicht erinnern muss.
2. Geringer Eingabeaufwand (Codes oder Zeigen).
3. Keine Modus- oder Syntaxfehler treten auf, weil nur zulässige Eingaben in der richtigen Form angeboten werden.
4. Der Umgang mit Menüs ist leicht erlernbar.
5. Unterstützen das entdeckende Lernen, da sie bisher nicht verwendete und nicht erinnerte Alternativen sichtbar machen.
6. Besonders gut für Anfänger und gelegentliche Benutzer geeignet.

Nachteile von Menüs

1. Die einzelnen Menü-Angaben müssen verständlich benannt bzw. bildlich beschrieben werden.
2. Benötigen Platz auf dem Bildschirm für die Darstellung.
3. Bei partieller Darstellung von strukturierten Menüs ergeben sich Orientierungs- und Navigationsprobleme.
4. Ein mehrstufiger Auswahlweg (z.B. in einem hierarchischen Menü) kann u.U. aufwendiger sein als eine kommando-orientierte Eingabe. Dies gilt allerdings nur für Routine-Benutzer und Experten.

2.1.4 Masken und Formulare

Masken und Formulare dienen zur 2-dimensionalen Informationsdarstellung und -erfassung [Shn92].

Masken sind 2-dimensionale Anzeigen des Zustandes von interaktiven Systemen, die meistens die gesamte Bildschirmfläche belegen und in bestimmte Bereiche unterteilt sind (z.B. Arbeitsbereich, Steuerungsbereich, Meldungsbereich). Häufig wird der Arbeitsbereich von Masken als Formular angelegt. Formulare sind 2-dimensionale Anordnungen von Feldern, in denen Information aus- und eingegeben werden kann. Im Unterschied zu Menüs, die sich auf Einzelaufgaben konzentrieren, können mit Formularen Kombinationen von Angaben erfasst und angezeigt werden.

Ausgaben in Formularen können sowohl Zustandsbeschreibungen, Handlungsanweisungen wie Erläuterungen zu anderen Feldern der Formulare sein. Die Eingaben erfolgen in der Regel über die Tastatur in die dafür vorgesehenen Textfelder, die syntaktischen und semantischen Einschränkungen unterliegen können.

Formulare werden generell verwendet, um Angabekombinationen für Aufträge zusammengefasst zu erheben, Standardkombinationen abzuspeichern und sie erneut anzubieten. Damit dienen die Formulare der sprachlichen Formulierung der Aufträge.

Kapitel 3

Individualisierung von Benutzungsschnittstellen

Ein vollkommenes Fehlen von Freiheitsgraden wie bei völlig systemgesteuerten Dialogen führt mit seiner Gefahr der Abstumpfung und Dequalifizierung genauso in die Irre, wie ein völliges Öffnen der Handlungsmöglichkeiten ohne Strukturierungshilfe.

Individualisierung soll die wünschenswerte Vielfalt von Systemfunktionalitäten zur Unterstützung eines angemessen breiten Aufgabenspektrums dem Benutzer vermitteln helfen. Sie dient damit der Unterstützung von abwechslungsreichen und anregenden Arbeiten als Ziel allgemeiner software-ergonomischer Bemühung. Begründet ist die Suche nach Individualisierbarkeit durch die grosse Bandbreite der Eigenschaften von Benutzer und einer ebenso grossen Bandbreite der Eigenschaften von Aufgaben.

Elemente der Anpassung

- **Synonyme:** Bestimmte Funktionsbezeichnungen lösen bei Benutzern gelegentlich falsche Assoziationen aus. Ihre Anpassung an den jeweiligen Sprachgebrauch kann Missverständnisse beseitigen und Klarheit schaffen.
- **Defaultwerte:** Zur Vereinfachung der Handhabung und Erhöhung der Effizienz werden in Anwendungen häufig Vorbelegungen angeboten. Diese Eintragungen können bereits vom Systementwickler aufgrund der Anforderungsanalyse vergeben worden sein oder vom Benutzer während der Benutzung eingetragen werden. Dabei könnte bei neuen Werten der

jeweils letzte Wert oder u.U. auch der bisher häufigster Wert als Voreinstellung übernommen werden.

- **Hilfestellung:** Hilfesysteme bieten dem Benutzer hinsichtlich des Ausführlichkeitsgrades, des Aktivierungszeitpunktes der vermittelten Inhalte und der Präsentationsart unterschiedliche Unterstützungen, die vom Benutzer oder vom System bedarfsentsprechend gesteuert werden können.
- **Interaktionsformen:** Die Anwendungssysteme sollten überwiegend redundante Interaktionsformen anbieten, indem sie die verschiedenen Interaktionstechniken parallel verwenden, damit dem Benutzer jederzeit die freie Wahl überlassen wird, die eine oder andere Form der Interaktion zu wählen.
- **Interaktionsinitiative:** Benutzer unterscheiden sich in ihrer Abhängigkeit von Interaktionsaufforderungen. Fest steht, dass Fortgeschrittenen und Experten Systeme mit benutzerinitiierten Interaktionsstilen bevorzugen. Die Anfänger dagegen benötigen mehr Interaktionsführung, was im Konflikt mit dem Bedürfnis, ein System frei explorieren zu können, steht. Eine komplexe Form der benutzergesteuerten Interaktionsinitiative stellen die sogenannten *Makros* dar. Damit kann der Benutzer eine Reihe von elementaren Interaktionsschritten zu einem Paket zusammenfassen und mit einem Aufruf ausführen lassen.
- **Modifikation und Erweiterung der Funktionalität:** Ein System ist nur so gut wie die Funktionalität dahinter. Gute Benutzbarkeit ersetzt nicht die aufgaben gerechte Funktionalität. Darüber hinaus sollte die Möglichkeit der Anpassung des Systems vorhanden sein (z.B. das Anlegen von Makros oder das Kombinieren von verschiedenen Anwendungen unter einer gemeinsamen Interaktiosoberfläche) [Haa92].

Kapitel 4

Aufgabenorientierte Systemgestaltung

Die Gestaltung von interaktiven Softwaresystemen muss sich an den Aufgaben, Arbeitsweisen, Fähigkeiten und Nutzungssituationen des Systembenutzers orientieren. Die Berücksichtigung dieser Faktoren bildet den Kern der software-ergonomischen Ansatzes zur Systemgestaltung. Eine aufgabenorientierte Systemgestaltung sollte verschiedene Zielsetzungen verfolgen: Die Verteilung der Aufgaben zwischen Mensch und Softwaresystem sollte so erfolgen, dass die grundlegend unterschiedlichen Fähigkeiten berücksichtigt werden, d.h. die kreativen sowie die Sachkompetenz und Flexibilität erfordernden Aufgabenteile dem Menschen zugeordnet werden, während das System die repetitiven, formalisierbaren oder mengenmäßig aufwendige Tätigkeiten übernimmt [Zie93].

4.1 Perspektiven und Methoden der Aufgabenanalyse

Die verschiedenen Methoden zur Analyse von Arbeitsaufgaben sind durch jeweils unterschiedliche Zielsetzungen und Perspektiven gekennzeichnet. Eine Perspektive berücksichtigt jeweils einen ganz bestimmten Aspekt des Mensch Rechner Systems, während andere Aspekte eher vernachlässigt werden. Da auch schon relativ einfache Aufgaben bei Betrachtung aller möglichen Aspekte eine hohe Komplexität aufweisen, ist eine solche Reduktion der Betrachtungsaspekte entsprechend der jeweiligen Zielsetzung erforderlich. Den Methoden liegen dementsprechend unterschiedliche Abstraktionsebenen und -prinzipien zugrunde.

Die Analyseebenen werden folgendermassen nach ihren Schwerpunkten unterschieden:

- **Organisatorische Ebene:** Zuordnung von Aufgaben auf organisatorische Einheiten und Rollen, Informationsflüsse, Kommunikation
- **Aufgabenebene:** Tätigkeitsspektrum, Beurteilung von Tätigkeiten, Belastungen
- **Funktionale Ebene:** Auslegung der Systemfunktionalität, Funktionsteilung zwischen Mensch und Rechner
- **Interaktionsebene:** Mensch-Computer-Interaktion, Dialogtechniken und -abläufe.

Prinzipiell lassen sich die Methoden der Aufgabenanalyse nach folgenden Herangehensweisen unterscheiden. Die *systemorientierten* und *benutzerorientierten* Herangehensweisen, je nachdem, ob die Anforderungen an die Funktionalität des Systems oder der Benutzer mit seinen Eigenschaften im Mittelpunkt der Betrachtung stehen.

4.2 Beschreibungsmerkmale von Aufgaben

Zur Untersuchung der einzelnen Aufgaben müssen für die jeweilige Analysezielsetzung relevanten Merkmale erfasst und in geeigneter Form dargestellt bzw. modelliert werden. Dabei lassen sich fünf Hauptgruppen von Merkmalen unterscheiden, die jeweils verschiedene strukturelle oder kontextbezogene Aspekte von Aufgaben beschreiben.

1. Die **statische Aufgabenstruktur** beschreibt im wesentlichen die Beziehungen zwischen einer Aufgabe, ihren Teilaufgaben und der Oberaufgabe, in der sie selbst wieder eine Teilaufgabe bildet. (Aufgabenhierarchie).
2. Die **dynamische Aufgabenstruktur** gibt die (zeitlichen) Abläufe an, beschreibt also die möglichen Übergänge zwischen einzelnen Aufgaben und die jeweiligen Ereignisse und Bedingungen, die zu diesen Übergängen führen.
3. Unter rechnergestützten Aufgaben werden die **Informationsobjekte** betrachtet. Deshalb werden in der Aufgabenanalyse die zu bearbeitenden (ein- und ausgehenden) Informationsobjekte mit ihrer jeweiligen Struktur und den Beziehungen untereinander bestimmt.

4. Unter Regeln zur Ausführung werden die **Vor- und Nachbedingungen** der Aufgabenausführung verstanden. Eine Aufgabe kann nur dann gestartet werden, wenn die Vorbedingungen erfüllt sind. Nach dem Abschluss der Aufgabe sollten gültige Nachbedingungen (Ergebnisse) erzeugt worden sein.
5. Zur Gestaltung aufgaben angemessener Systeme ist es unerlässlich, die Einbettung von Aufgaben in den realen Arbeitskontext zu betrachten. Darunter ist eine Vielzahl von Parametern gemeint, die als **Kontextparameter** bezeichnet werden können. Solche Parameter bestehen in der Regel in der Priorität, Häufigkeit und Repetitivität von Aufgaben, den Mengengerüsten hinsichtlich der zu bearbeitenden Daten oder Faktoren wie Unterbrechungshäufigkeit.

Für den Zweck der Systemgestaltung ist es sinnvoll, die Aufgabenanalyse unter zwei Aspekten zu betrachten: Zum einem sollten nur die wirklich gestaltungsrelevanten Merkmale erfasst werden, um das Ganze zielgerichtet und ökonomisch zu halten und zum anderen sollte vermieden werden, überflüssigen Aufwand auf die Analyse der Ist-Situation zu verwenden. Obwohl diese wertvolle Ausgangsinformation liefert, sollte das Schwergewicht auf die Analyse der Soll-Konzeption gelegt werden [You89].

4.3 Analyse kognitiver Aufgaben

Kognitive Aufgabenanalyse hat zum Ziel, die mentale Repräsentation von Aufgaben durch den Benutzer zu bestimmen und zu modellieren, um daraus Aussagen über Leistungsgrößen, die Erlernbarkeit von Benutzungsschnittstellen, Transfer von Wissen sowie die Gedächtnisbelastung und Fehlermöglichkeiten zu gewinnen. Die Modellierung einer kognitiven Aufgabenstruktur erlaubt die Abbildung der einzelnen Aufgaben in Handlungssequenzen.

Die kognitiv orientierte Analyseverfahren haben nur begrenzten Nutzen, obwohl teilweise Vorhersagen relevanten Benutzerverhaltens möglich wären, sind Analysen kompletter Systeme oder Systemkomponenten allein schon aus Aufwandsgründen praktisch nicht möglich.

Kapitel 5

Fehlermanagement und Hilfesysteme

5.1 Grundlagen und Problemfelder

Die Anpassung von technischen Systemen an den Menschen (und nicht umgekehrt) ist ein Hauptziel der Software-Ergonomie. Der Benutzer von technischen Systemen wird als aktiv Handelnder und das System selbst als Werkzeug angesehen. Wenn der aktiv handelnde Benutzer bei der Bearbeitung von Aufgaben unterstützt werden soll, so sind Grundkenntnisse über den menschlichen Handlungsprozess erforderlich.

5.1.1 Der menschliche Handlungsprozess

Bei der Arbeit am Computer werden Ziele durch die zu erledigende Arbeitsaufträge und durch die Benutzer formuliert. Die Benutzer bestimmen mehr oder weniger darüber, wie sie ihre Aufgaben im einzelnen erledigen möchten. Sie regulieren ihre Handlungen, indem sie Grobziele und Teilziele entwickeln (Zielbildung), entsprechende Pläne zur Realisierung aufstellen (Planung), ausführen (Handlungsausführung) und vergleichen ob die tatsächlichen Handlungsergebnisse von den erwarteten abweichen (Rückmeldungsverarbeitung) [MGP91].

Bei der *Zielbildung* möchten die Benutzer gerne wissen, welche ihrer Ziele sie mit dem vorliegenden System erreichen können und welche nicht. Bei der *Planung* geht es um die Frage, wie sich die Ziele erreichen lassen und welche Teilziele angestrebt werden müssen. Während der *Handlungsausführung* werden die ursprüngliche Pläne schrittweise weiterentwickelt, verfeinert

und gegebenenfalls korrigiert. Bei der *Rückmeldungsverarbeitung* geht es um die Erkennung, ob die vorhandenen Ziele mit den Handlungsergebnissen übereinstimmen.

5.1.2 Fehlhandlungen

Die Fehlhandlungen sollten im Sinne einer mangelhaften Anpassung verstanden werden und nicht so, dass der Benutzer etwas falsch gemacht hat. Weder der Benutzer noch das Computer-System für sich alleine genommen lassen sich als Urheber für die Fehlhandlungen identifizieren [Fre91].

Die Qualität der Unterstützungskomponenten von Software-Produkten wird gerade daran gemessen, wie oft die Benutzer in Mismatch-Situationen gerät und wie gut sie bei der Bewältigung von Mismatch-Situationen unterstützt werden. Mismatch bedeutet, dass es Mängel an der gegenseitigen Anpassung zwischen Mensch und Computer gibt. Durch Software-Evaluation kann die Art und die Anzahl von Fehlhandlungen sowie die Dauer der Fehlerbewältigungszeit bei der Bearbeitung von Standardaufgaben bestimmt werden [Bro91].

Beim Fehlerbewältigungsprozess lassen sich drei Schritte unterscheiden: Fehlerentdeckung, Fehlerdiagnose und Fehlerkorrektur. [ZLW91]

- *Fehlerentdeckung*: Fehlerbewältigung wird eingeleitet durch den Verdacht oder durch das genaue Wissen, dass ein Fehler aufgetreten ist. Dabei ist es meistens unklar, worin der Fehler besteht oder wie er korrigiert werden kann.
- *Fehlerdiagnose*: Hier werden die Vorstellungen über den richtigen Handlungsverlauf mit dem tatsächlichen Handlungsverlauf verglichen. Aus diesem Vergleich leiten sich Vermutungen ab, wie ein Fehler zustande gekommen sein könnte.
- *Fehlerkorrektur*: Kann entweder direkt oder kompensatorisch erfolgen. Bei der direkter Fehlerkorrektur wird der fehlerhafte Handlungsschritt rückgängig gemacht, oder inverse Aktionen werden ausgeführt. Diese Art der Korrektur wird nur bei Fehlern eingesetzt, deren Konsequenzen unmittelbar erkannt werden und nur wenige Teilhandlungen betreffen. Bei der kompensatorischen Korrektur werden Fehlerkonsequenzen in zukünftige Handlungen einbezogen oder neue Handlungen geplant, die einen Zustand herstellen, von dem aus das ursprüngliche Ziel ebenfalls

erreicht werden kann. Kompensatorische Fehlerkorrektur bezieht sich vor allem auf Fehler, die sehr spät entdeckt werden und deren Konsequenzen eine grössere Anzahl von Handlungsschritten betreffen.

5.2 Fehlervermeidung

Bei der Betrachtung des menschlichen Fehlebewältigungsprozesses wird von der Existenz eines Fehlers ausgegangen. Es ist jedoch möglich einiges zu tun, damit es erst gar nicht zu Fehlern kommt, oder dass die Fehler leicht entdeckt werden können.

Konsistente Dialoggestaltung

Die wichtigste Strategie zur Vermeidung von Routinefehlern ist die *konsistente Gestaltung der Benutzungsschnittstelle* eines Softwareprodukts in Bezug auf unterschiedliche Aspekte wie Informationsdarstellung, Aktivierung von Funktionen, Rückmeldungen.

- Informationsdarstellung: Gleichartige Informationen werden stets an der gleicher Stelle, in gleicher Formulierung, Typografie und Layout dargestellt (z.B. Menüs).
- Aktivierung von Funktionen: Gleichartige Funktionen in unterschiedlichen Arbeitskontexten/-modi werden auf gleiche Weise aktiviert.
- Rückmeldungen: Beim Umschalten in einen anderen Arbeitsmodus oder bei der Ausführung von Funktionen erfolgt eine entsprechende Rückmeldung.

Eine konsistente Gestaltung der Benutzungsschnittstelle sorgt dafür, dass Routinehandlungen des Benutzers seltener zur Gewohnheitsfehlern führen. Darüber hinaus wird auch das Erlernen neuer Funktionalität vereinfacht, da sich bekannte Handlungsmuster übertragen lassen.

Einschränkung von Freiheitsgraden

Bei der Einschränkung von Freiheitsgraden werden i.d.R. unzulässige Eingaben abgewiesen. Z.B. werden in einem Eingabefeld wo ein Datum eingegeben ist, alle Eingaben ausser Ziffern mit Punkten abgelehnt. Oder die Funktionen, die in der aktuellen Situation nicht anwendbar sind, werden im Menü gesperrt.

Die Einschränkungen von Freiheitsgraden steht im Widerspruch zum Modell des aktiven Benutzers. Hier besteht ein Konflikt zwischen dem Ziel, dem Benutzer die vollständige Kontrolle über ein Softwaresystem zu ermöglichen, und dem Ziel, Fehler zu vermeiden. Eine Entscheidung darüber, welchem dieser Ziele die höhere Priorität zukommt, muss in jedem Einzelfall in Abhängigkeit von der Aufgabe getroffen werden.

Sicherheitsrückfragen

Sicherheitsrückfragen werden zur Fehlervermeidung eingesetzt, wenn der Benutzer Funktionen aktiviert, die unbeabsichtigte Effekte haben können oder zur Verfälschung bzw. zum Verlust von Daten führen können und die nicht ohne weiteres umkehrbar sind. In einer solchen Rückfrage werden die möglichen Konsequenzen der aufgerufenen Funktion beschrieben und der Benutzer muss entweder bestätigen, dass er die Funktion weiter ausführen will, oder die Funktion abbrechen.

5.3 Fehlermanagement

Ziel von Fehlermanagement ist es, den Aufwand für die Fehlerbewältigung zu reduzieren, da diese einen beträchtlichen Teil der Arbeitszeit in Anspruch nimmt. Eine gelungene Reduktion wäre durch eine Verringerung der Fehlerbewältigungszeit oder durch eine geringere Beanspruchung von Beratungsleistungen erkennbar. Im folgenden werden einige Strategien zur Beseitigung von bestimmten Arten von Fehlern beschrieben.

Routinefehler

Konsistente und transparente Systemgestaltung hilft nicht nur viele Routinefehler (Gewohnheits-, Erkennens-, Bewegungsfehler) zu vermeiden, sondern sorgt auch dafür, dass solche Fehler leichter entdeckt und identifiziert werden. Für ihre Korrektur ist eine Funktionalität von Vorteil, die es erlaubt unbeabsichtigte Handlungsergebnisse rückgängig zu machen bzw. zu kompensieren (inverse Funktionen, allgemeine Undo-Funktion).

Wissensfehler

Bei der Fehlerdiagnose und der Fehlerkorrektur von Wissensfehlern sollten im wesentlichen Suchprozesse und exploratives Agieren unterstützt werden. Dazu dienen insbesondere Hilfesysteme, Apropos- und Explorationsfunktionen. Mit einer Apropos-Funktion werden Begriffe, Prozeduren, Funktionen

und dergleichen verfügbar gemacht, die mit dem im System verwendeten Begriffen synonym oder verwandt sind. Damit kann der Benutzer auch mit einer vagen Vorstellung Funktionsnamen und spezifische Informationen ausfindig machen. Unter explorativen Agieren werden bestimmte Funktionen gemeint, die sich anhand der Testdaten vom Benutzer ausprobieren lassen, um die Wirkungsart und -weise der im System vorhandenen Funktionen zu erläutern.

Darüber hinaus gibt es noch die Denkfehler, Merk- und Vergessensfehler, Urteilsfehler etc.. Die jeweiligen Strategien sind den oben Vorgestellten teilweise ähnlich und werden hier nicht weiter beschrieben. Entscheidend ist nur, dass mit Hilfe des Fehlermanagements die einzelnen Fehler effektiv bewältigt werden können, wenn die entsprechenden Strategien richtig angewendet werden.

5.4 Hilfesysteme

Ein Hilfesystem besteht aus Programmen oder Programmteilen und Datenbanken, die eine Hilfeleistung in Form von Texten, Abbildungen etc. mit Informationen und Erläuterungen zur Benutzung eines oder mehrerer Softwareprodukte und Hilfsfunktionen (z.B. Anzeige-, Such- und Navigationsfunktionen) zur Erschließung und Nutzung der Hilfeleistungen bereitstellen. Das Hauptmerkmal eines Hilfesystems besteht darin, dass es den Benutzer während der Arbeit mit dem betreffenden Produkt schnell und unmittelbar unterstützt, ohne den aktuellen Arbeitskontext zu verlassen [Rup87].

Es existieren zwei Bereiche in denen Hilfesysteme Unterstützung bieten können:

- Wissenserwerb - Benutzer lernen neue Sachverhalte, Wissensprobleme werden beseitigt.
- Fehlermanagement - Benutzer orientieren sich über Informationen und Funktionen die zur Fehlerbehebung notwendig sind.

Darüber hinaus gibt es zwei Gestaltungsarten bei der Entwicklung von Hilfesystemen:

- Passive Hilfe: die Initiative geht vom Benutzer aus. Er ruft die Hilfe auf und fordert damit die Hilfeleistung an.

- Aktive Hilfe: die Initiative geht von Hilfesystem bzw. dem Benutzten Softwareprodukt aus. Die Hilfeleistung wird ohne explizite Anforderung durch den Benutzer bereitgestellt.

Hilfesysteme eignen sich für die Bereitstellung von Informationen lokaler Bedeutung, d.h. zur Ausführung einzelner Arbeits- und Bedienungsschritt, zum aktuellen Arbeitskontext oder zur aufgetretenen Fehlern. Sie sind weniger geeignet für Informationen globaler Bedeutung, etwa zur Planung komplexer Arbeitsvorgänge und Abläufe. Denn letzteres kann besser in Handbüchern vermittelt werden.

Literaturverzeichnis

- [Bal88] H. Balzert, *Der fehlerbewältigungsprozess*, de Gruyter, 1988. 3
- [Bro91] F.C. Brodbeck, *Fehlerbewältigungsdauer und die nutzung von unterstützungsmöglichkeiten.*, Huber, 1991. 16
- [Fre91] M. Frese, *Fehlermanagement: Konzeptionelle Überlegungen*, Huber, 1991. 16
- [Haa92] D. Haaks, *Anpassbare informationssysteme*, Verlag für Angewandte Psychologie, 1992. 11
- [Mar92] A. Marcus, *Graphic design for electronic documents and user interfaces*, ACM, 1992. 4
- [MGP91] G.A. Miller, E. Galanter, and K.H. Pribram, *Strategien des handelspläne und strukturen des verhaltens*, Klett-Cotta, 1991. 15
- [Obe91] H. Oberquelle, *Perspektiven für die gestaltung der mensch-computer-interaktion*, Teubner, 1991. 1
- [Rup87] Rupiatta, *Benutzerdokumentation für softwareprodukte*, BI-Wissenschaftsverlag, 1987. 19
- [Shn92] Shneiderman, *Designing the user interface. strategies for effective human computer interaktion*, Addison-Wesley, 1992. 6, 8
- [Vol93] W. Volpert, *Von der software-ergonomie zur arbeitsinformatik*, Teubner, 1993. 1
- [You89] E. Yourdon, *Modern structured analyses*, Prentice Hall, 1989. 14
- [Zie93] J. Ziegler, *Entwurf graphischer benutzungsschnittstellen*, Oldenbuorg, 1993. 12
- [ZLW91] D. Zapf, T. Lang, and A. Wittman, *Der fehlerbewältigungsprozess*, Huber, 1991. 16

Index

- akustischen Ausgabe, 4
- Aufgaben angemessene Gestaltung,
1
- Aufgabenanalyse, 12
- Defaultwerte, 10
- dynamische Aufgabenstruktur, 13
- Ein-/Ausgabeelemente, 3
- Eingabeelemente, 5
- Fehlermanagement, 15, 19
- Formulare, 8
- GUI, 3
- Hilfesysteme, 19
- Individualisierbarkeit, 10
- Interaktionsformen, 11
- Interaktionsinitiative, 11
- Kognitive Aufgabenanalyse, 14
- Kommandodialogen, 5
- Masken, 8
- Menü, 6
- Mensch-Computer-Interaktion, 1
- menschengerechte Gestaltung, 1
- organisationsgerechte Gestaltung, 1
- Software-Ergonomie, 1, 2
- statische Aufgabenstruktur, 13
- technikbewusste Gestaltung, 1
- Visualisierungsinstrumente, 4
- visuelle Ausgaben, 3
- Wissenserwerb, 19