

# STATUSANZEIGEN

Diplomarbeit von Peter Thielmann

Februar 2001

Universität Hamburg

Fachbereich Informatik

**Betreuer:** Prof. Horst Oberquelle, Prof. Christian Freksa

## Inhaltsverzeichnis

Vorwort.....	1
1 Einleitung.....	2
1.1 Problemumfeld.....	2
1.2 Aufgabenstellung.....	3
1.3 Aufbau der Arbeit.....	3
2 Grundlagen.....	4
2.1 Zweck von Statusanzeigen.....	4
2.2 Definition von Status und Statusanzeigen.....	7
2.3 Abgrenzung von Statusanzeigen zu anderen Bereichen.....	7
2.4 Einordnung in die Software-Ergonomie.....	9
3 Wahrnehmung und Informationsverarbeitung.....	12
3.1 Begriffe.....	12
3.2 Optische Wahrnehmung (Sehen).....	13
3.3 Akustische Wahrnehmung (Hören).....	20
3.4 Menschliche Signalverarbeitung.....	23
3.5 Mentale Modelle.....	28
3.6 Menschliches Handeln.....	35
4 Informationsdarstellung.....	42
4.1 Allgemeine Überlegungen.....	42
4.2 Optische Ausgaben.....	48
4.3 Akustische Ausgaben.....	69
4.4 Äußere Einflüsse auf die Anzeige.....	78
4.5 Bedienelemente.....	81
4.6 Visuelles Feedback außerhalb des Bildschirms.....	83
5 Statusanzeigen in Literatur und Praxis.....	85
5.1 Statusanzeigen in der wissenschaftlichen Literatur.....	85
5.2 Statusanzeigen in kommerziellen Style Guides.....	90
5.3 Beispiele für Statusanzeigen.....	96
5.4 Auswertung der gezeigten Beispiele.....	120

---

6	Kritische Betrachtung existierender Statusanzeigen.....	129
6.1	Informationsumfang.....	129
6.2	Wahrnehmbarkeit.....	131
6.3	Verständlichkeit.....	135
6.4	Nützlichkeit.....	139
6.5	Die schwerwiegendsten Fehler.....	143
6.6	Testen von Statusanzeigen.....	143
6.7	Hardware.....	145
7	Eigene Beispiele.....	148
7.1	Fortschrittsanzeige.....	148
7.2	Datei kopieren.....	155
8	Zusammenfassung.....	160
	Danke.....	161
	Literaturverzeichnis.....	162
	Anhang.....	167
	Benutzerbefragung.....	167
	Inhalt der CD.....	172
	Abbildungen in Farbe.....	173

## Vorwort

Motivation für diese Arbeit gaben mir die „*User's Bill of Rights*“ ([Wildstrom, 1998]):

1. *The user is always right. If there is a problem with the use of the system, the system is the problem, not the user.*
2. *The user has the right to easily install software and hardware systems.*
3. *The user has the right to a system that performs exactly as promised.*
4. *The user has the right to easy-to-use instructions for understanding and utilizing a system to achieve desired goals.*
5. *The user has the right to be in control of the system and to be able to get the system to respond to a request for attention.*
6. *The user has the right to a system that provides clear, understandable, and accurate information regarding the task it is performing and the progress toward completion.*
7. *The user has the right to be clearly informed about all system requirements for successfully using software or hardware.*
8. *The user has the right to know the limits of the system's capabilities.*
9. *The user has the right to communicate with the technology provider and receive a thoughtful and helpful response when raising concerns.*
10. *The user should be the master of software and hardware technology, not vice-versa. Products should be natural and intuitive to use.*

Auch wenn viele dieser Punkte – nicht zuletzt dank der EU-Bildschirmarbeitsplatzverordnung – hier in Europa eine Selbstverständlichkeit sind, so zeigten mir eigene Erfahrungen, dass nicht alle Punkte gleich gut erfüllt werden. Das „sechste Gebot“, das Recht auf Information über den Systemzustand, bildete damit den Aufhänger für diese Arbeit.

# 1 Einleitung

## 1.1 Problemumfeld

Statusanzeigen verrichten vom Einschalten bis zum Ausschalten eines PCs überall unauffällig ihren Dienst: LEDs zeigen, dass der PC eingeschaltet ist oder dass auf die Festplatte zugegriffen wird. Unterstützt werden diese Informationen zusätzlich durch Geräusche, so dass der Benutzer über den groben Status seines PCs auch dann informiert ist, wenn er nicht hinsieht: Das Geräusch der Lüfter informiert darüber, dass der PC eingeschaltet ist, ein klackend-knisterndes Geräusch informiert über den Zugriff auf die Festplatte.

Beim Start des Betriebssystems sieht der Benutzer einen ständigen Wechsel verschiedener Bilder. Dieser zeigt ihm, dass etwas passiert. Wer den Ablauf bereits kennt, weiß auch in etwa, an welcher Stelle der Prozess sich gerade befindet. Auch beim Herunterfahren des Betriebssystems sieht der Benutzer einen Wechsel verschiedener Bilder. Diese bleiben zwischendurch jedoch immer wieder für eine längere Zeit unverändert, so dass nicht mit Sicherheit festgestellt werden kann, ob das System „hängt“, oder ob es diesmal nur etwas länger dauert.

Bei einigen PCs ist ohne sehr genaues Hinsehen nicht zu erkennen, ob der Benutzer online ist oder nicht. Eventuell bezahlt er dadurch überflüssige Online-Zeit.

Die Kombination eines langsamen PCs mit einer Textverarbeitung, die den Ruf hatte, nicht besonders stabil zu sein, führte dank fehlender Rückmeldung dazu, dass der Benutzer nach zwei, drei Minuten glaubte, das Programm wäre beim Einfügen eines etwas größeren Bildes in seinen Text eingefroren. Nur per Zufall hat er irgendwann herausgefunden, dass es lediglich „etwas“ länger dauerte, das Bild einzufügen (es dauerte eine knappe Viertelstunde).

Statusanzeigen mögen zwar unauffällig sein, aber bereits diese wenigen Beispiele zeigen, dass sie den Benutzer bei der Arbeit unterstützen, indem sie ihm Sicherheit geben. Fehlen sie, fühlt der Benutzer sich unsicher – unsicher über den Fortschritt eines Prozesses, unsicher über den momentanen Modus des Systems (beispielsweise online oder offline). Diese Unsicherheit führt zu Frustration bei der Arbeit mit dem Computer, sie kann zum Teil sogar unnötige Kosten verursachen.

Statusanzeigen informieren den Benutzer hauptsächlich über den Fortschritt von Prozessen sowie über einen momentanen Modus.

Aussagen zum Prozessfortschritt sind in der wissenschaftlichen Literatur meist unbestimmt („Der Benutzer soll informiert werden“), es gibt in der Regel keine kon-

kreten Vorschläge, wie dieses geschehen soll. In kommerziellen Style Guides finden Fortschrittsanzeigen erst in letzter Zeit erhöhte Beachtung.

Kommerzielle Style Guides für grafische Betriebssysteme geben schon seit langem Hinweise, wie der Mauszeiger in Abhängigkeit von wenigen bestimmten Modi zu ändern ist. Für weiter, nicht im Style Guide berücksichtigte Modi wird jedoch keine Empfehlung gegeben. In der wissenschaftliche Literatur sind zu diesem Thema nur vereinzelte Werke zu finden.

## 1.2 Aufgabenstellung

Aufgabe dieser Diplomarbeit ist es zu untersuchen, wie Statusanzeigen beschaffen sein müssen, damit sie den Menschen bei seiner Arbeit am Computer unterstützen.

Vermittlung von Prozessfortschritt und Information über einen Modus wurden bisher nicht zusammen betrachtet, sondern entweder einzeln behandelt oder im Rahmen eines übergeordneten Themas gestreift. In dieser Arbeit beschäftige ich mich mit Statusanzeigen aller Art, wozu außer Prozessfortschritt und Anzeige eines Modus noch weitere Punkte gehören. Dadurch ist eine ausführliche Betrachtung mit allen zugehörigen Aspekten möglich.

## 1.3 Aufbau der Arbeit

In dieser Diplomarbeit befasse ich mich zuerst mit der Frage, was Statusanzeigen eigentlich sind und wie sie im Umfeld der Software-Ergonomie einzuordnen sind. Ich untersuche, welche physiologischen und psychologischen Eigenschaften des Menschen sowie welche Regeln der Informationsdarstellung beachtet werden müssen, damit Statusanzeigen den Benutzer bei seiner Arbeit unterstützen.

Im Anschluss daran stelle ich Statusanzeigen aus der Praxis vor. Ich analysiere sie in Bezug auf die Art und Weise, wie und welche Statusinformationen angezeigt werden und zeige häufig gemachte Fehler.

Für einen Teilaspekt von Statusanzeigen, nämlich die Vermittlung von *Prozessfortschritt*, entwickle ich eigene Anzeigen und lasse sie von Benutzern beurteilen.

Die Arbeit befasst sich primär mit dem Visualisieren von Dingen, die für das interaktive Arbeiten benötigt werden, d.h. mit Anzeigen, die zur Unterstützung der Interaktion dienen. Ich gehe aber auch auf Statusvermittlung durch Audiosignale ein. Der Begriff *Statusanzeige* wird von mir der Einfachheit halber unterschiedslos sowohl für optische als auch für akustische Ausgaben verwendet.

## 2 Grundlagen

### 2.1 Zweck von Statusanzeigen

Bereits ein kurzer Blick auf den Arbeitsalltag eines PC-Benutzers (siehe Einleitung) zeigt, dass der Benutzer vor seinem PC oft wie vor einer Black Box sitzt und über den momentanen Status seines Systems nicht informiert ist. Die eher unbestimmte Frage des Benutzers nach dem momentanen Status seines Systems kann zunächst einmal präzisiert werden:

- Passiert gerade etwas, oder ist das System „tot“? (Wird momentan ein Prozess ausgeführt? Welcher Prozess wird gerade ausgeführt?)
- Wie lange dauert es noch? (Wann ist dieser Prozess fertig? Welche Prozesse folgen eventuell noch, wie lange dauern diese Prozesse?)
- Was kann ich momentan tun, was nicht? (Befindet sich das System in einem Modus, der mir nur eine bestimmte Auswahl an Aktionen erlaubt? Falls ja, in welchem Modus befindet sich das System, und welche Aktionen stehen mir zur Verfügung?)
- Wie viel Platz ist noch auf der Festplatte, wie stark ist das System ausgelastet? (Wie viel einer bestimmten Ressource steht noch zur Verfügung?)

Mit Hilfe von Statusanzeigen kann der Benutzer die Antworten auf diese Fragen erhalten. Statusanzeigen machen dem Benutzer das System transparent und vermeiden dadurch Frustration und Unsicherheit. Alan Cooper hat den Zweck einer Statusanzeige am Beispiel eines Fortschrittsbalkens recht humorvoll beschrieben:

*The process dialog box alerts the user to the program's inability to respond normally. It also warns the user not to be overcome with impatience and to resist banging on the keyboard to get the program's attention. ([Cooper, 1995], Seite 315)*

Natürlich ist die Sorge um die Tastatur nicht der einzige Grund für Statusanzeigen. Ein Dialog mit Informationen über den Prozessfortschritt beispielsweise soll

- den Benutzer informieren, dass ein zeitaufwendiger Vorgang läuft und wie lange er voraussichtlich noch dauert,
- dem Benutzer versichern, dass der Systemzustand völlig normal ist und
- dem Benutzer eine Möglichkeit geben, diesen Vorgang abubrechen ([Cooper, 1995], Seite 316).

Cooper verlangt grundsätzlich, dass Programme den Benutzer über ihre Aktionen informieren. Sie sollen Lichter, Skalen oder andere Anzeigen im Interface eingebaut haben, um dem Benutzer Statusinformationen zu vermitteln, falls dieser sie wünscht. Der Benutzer soll fortwährend über den Systemstatus informiert sein, so wie der Autofahrer durch die Instrumente auf dem Armaturenbrett über den Zustand seines Autos informiert ist ([Cooper, 1995], Seiten 443 und 448).

Und wenn Alan Cooper über heutige Programme meint:

*We think that programs must be dependable because most programs don't offer sufficient visual cues for the user to supervise its actions. ([Cooper, 1995], Seite 428),*

dann bleibt mir angesichts der berühmt-berüchtigten Unzuverlässigkeit eines Teils heutiger Software mit nur wenig Aussicht auf Besserung nur der boshafte Umkehrschluss, dass die unzuverlässigen Programme wenigstens ausreichend Informationen über ihren Status geben sollen, damit der Benutzer weiß, dass sie noch „leben“.

Andere Autoren hingegen gehen nicht näher auf Statusanzeigen ein, sondern betrachten Rückmeldungen allgemein. Brodbeck und Rupiotta beispielsweise zeigen auf, dass Rückmeldungen zur Fehlervermeidung dienen und empfehlen:

*Beim Umschalten in einen anderen Arbeitsmodus oder bei der Ausführung von Funktionen erfolgt eine entsprechende Rückmeldung. ([Brodbeck und Rupiotta, 1994], Seite 205)*

Die Bedeutung von Rückmeldungen sowie die Auswirkungen beim Fehlen ausreichender Rückmeldungen für den Benutzer sind bereits seit längerer Zeit bekannt. Nievergelt hat bereits 1983 über fehlende Rückmeldung zeitgenössischer Programme geklagt und einige Vorschläge zur Verbesserung gemacht:

*Fehlende Rückmeldung zum Benutzer ist die meistverbreitete Pflichtverletzung von Autoren interaktiver Programme. Der Benutzer gibt einen Befehl ein und wartet; falls die Antwort nicht unmittelbar erfolgt, wird er unsicher. Er fragt sich, ob er wohl vergessen habe, die RETURN-Taste zu drücken und das Programm noch darauf wartet, oder ob es „gestorben“ sei. Ohne Rückmeldung vom Programm steht dem Benutzer keine sinnvolle Möglichkeit des Eingreifens offen: Falls er ein zweites Mal die RETURN-Taste drückt, könnte dies als Antwort auf die nächste Programmaufforderung gelesen werden; bricht er die Operation ab, kann das Programm in einen undefinierten Zustand geraten; falls er warten will, wie lange soll er dies tun?*

*Systematische Rückmeldung aller Benutzer-Eingaben durch das System ist die einzige Art zu vermeiden, dass der Benutzer im Ungewissen gelassen wird. Oft genügt das „Echo“ der Eingabe auf dem Bildschirm. Bei Befehlen, deren Ausführung längere Zeit in Anspruch nimmt, empfiehlt es sich, zusätzlich In-*



*formationen über den Ablauf der Befehlsausführung auszugeben. Schon einfache Effekte wirken äusserst beruhigend auf den Benutzer: ein Pfeifton, um das Ende einer Operation anzukündigen, oder eine wachsende Folge von Punkten, um zu zeigen, dass das System arbeitet. Der Programmierer muss nur daran denken, die Rückmeldung dort auf den Schirm zu schreiben, wo sie gesehen wird; gewöhnlich nahe der letzten Benutzer-Eingabe, oder an einem eigens für Systemmeldungen reservierten Platz. ([Nievergelt, 1983], Seiten 32-33)*

Und 1988 forderte Galitz, dass ein System alle Aktionen des Benutzers bestätigen soll, entweder durch sofortige Ausführung, durch einen geänderten Zustand oder Wert, durch eine Fehlermeldung oder durch eine In-Bearbeitung-Anzeige. Seiner Ansicht nach ist Rückmeldung ein notwendiger Bestandteil, der die menschliche Performance erhöht und Vertrauen schafft. Rückmeldung erfolgt normalerweise, indem das System die Anforderung des Benutzers ausführt und kann entweder implizit (ein geänderter Zustand) oder explizit (eine Meldung) sein. Der Bildschirm sollte nie mehr als ein paar Momente leer sein, da der Benutzer ansonsten denken könnte, das System hätte versagt. Benötigt ein Prozess mehr Zeit, als der Benutzer erwarten würde, sollte das System die Anforderung des Benutzers bestätigen und eine In-Bearbeitung-Meldung ausgeben ([Galitz, 1988], Seite 38).

Sichergestellt werden muss aber auf jeden Fall, dass Statusanzeigen akkurat arbeiten. Ein Fortschrittsbalken, der schon nach der Hälfte einen Fortschritt von 90% anzeigt, ist schlichtweg ein Ärgernis ([Wildstrom, 1998]). Und eine Restzeitanzeige, die den Countdown für die letzten Sekunden mehrmals wiederholt, ist ebenso wie die Anzeige eines bereits zu 123% erledigten Jobs höchstens beim ersten Mal ein Quell der Belustigung.

Einige Anwendungsbereiche benötigen deutlich mehr Feedback, um dem Benutzer ein zufrieden stellendes und Erfolg versprechendes Arbeiten zu ermöglichen, eine Arbeit ohne Statusanzeigen wäre zum Teil gar nicht möglich. Zu diesen Anwendungsbereichen gehören beispielsweise CSCW-Anwendungen<sup>1</sup> oder Anwendungen im Bildungsbereich. Auch bei ferngesteuerten Maschinen ist ein Feedback durch Statusanzeigen unbedingt notwendig, ganz besonders bei außergewöhnlichen Zwischenfällen wie beispielsweise einem Programmcrash. Sicherzustellen ist hier unbedingt, dass das Feedback und damit auch die Statusinformation ohne größere Verzögerung den Benutzer erreicht ([Shneiderman, 1998], Seiten 219-220).

---

<sup>1</sup> CSCW bedeutet Computer Supported Cooperative Work, computerunterstützte Gruppenarbeit

## 2.2 Definition von Status und Statusanzeigen

Was ist *Status*? Laut [Brockhaus, 1993] (Band 21, Seite 104) ist Status ein anderes Wort für Lage oder Situation. Status ist also zunächst einmal die momentane Situation eines Computersystems. Die genauere Betrachtung der Fragenliste zum Status des Systems (Kapitel 2.1, *Zweck von Statusanzeigen*) zeigt, dass der Begriff *Status* zwei recht unterschiedliche Aspekte enthält:

- Dynamische Zustände (Fortschritt eines Prozesses, zeitlicher Verlauf der Auslastung einer Ressource)
- Statische Zustände (aktueller Modus eines Prozesses oder eines Systems, Auslastung einer Ressource, Ereignisse<sup>1</sup>)

*Status* kann für diese Arbeit also definiert werden als die Vereinigung dynamischer und statischer Zustände.

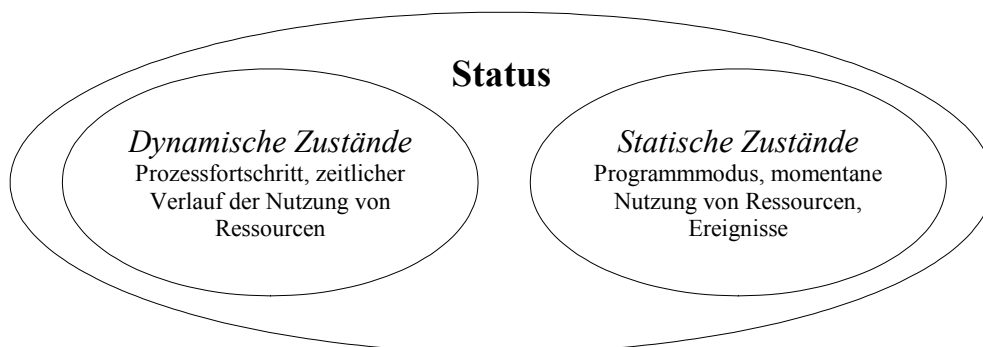


Abbildung 1: Menge der Systemzustände

Analog haben auch die in dieser Arbeit untersuchten *Statusanzeigen* zwei verschiedene Funktionen. Sie sollen den Benutzer sowohl über dynamische Zustände informieren – dazu gehört beispielsweise der Programmablauf – als auch ihm einen statischen Zustand anzeigen – beispielsweise die momentane Auslastung einer Ressource,.

## 2.3 Abgrenzung von Statusanzeigen zu anderen Bereichen

Statusanzeigen haben Berührungspunkte mit anderen Bereichen:

- Wie Fehlermeldungen, so können auch Statusanzeigen über fehlerhafte Systemzustände informieren. Im Gegensatz zu Fehlermeldungen, die nur bei kritischen Systemzuständen sichtbar sind, zeigen Statusanzeigen den Systemzustand jedoch

<sup>1</sup> Für den Benutzer ist lediglich interessant, ob ein bestimmtes Ereignis bereits eintrat oder noch nicht. Es gibt also zwei für den Benutzer interessante Zustände oder Modi, nämlich *Ereignis trat noch nicht ein* und *Ereignis trat ein*, so dass Ereignisse den statischen Zuständen zugeordnet werden können.

ständig an und informieren auch über Normalität. Es ist also stets eine eindeutige Zuordnung möglich.

- Bedienelemente werden zum Einstellen von System- und anderen Parametern gebraucht, während reine Statusanzeigen lediglich über den Systemzustand informieren und dem Benutzer keine direkte Manipulationsmöglichkeit bieten. Viele Bedienelemente jedoch zeigen gleichzeitig Statusinformationen an: Ein Button in einer Symbolleiste informiert über seinen Zustand („gedrückt“ oder „nicht gedrückt“) über den Status der zugehörigen Funktion, eine Markierung neben einem Menüeintrag teilt dem Benutzer mit, dass die Funktion zur Zeit aktiv ist, ein Schieberegler wie der unter Windows zum Einstellen der Bildschirmauflösung informiert gleichzeitig über die gewählte Auflösung. Weiterhin können Bedienelemente nicht nur direkt über den Zustand des Programms oder einer Funktion informieren, sondern auch ihren eigenen Status anzeigen: Eingabefelder beispielsweise können eine Anzeige haben, die über die Gültigkeit der Eingabe informiert. Ein Bedienelement in einem Dialog zeigt durch in der Regel dezente Hervorhebung, dass es den Fokus hat, also das momentan aktive Element ist. Einige Bildelemente können daher nicht eindeutig als Bedienelement oder als Statusanzeige klassifiziert werden.
- Auch wenn einige Autoren die Meinung vertreten, die Unterscheidung zwischen Online-Hilfe und der Benutzungsschnittstelle sei nicht immer eindeutig und jede Art von Feedback könne auch als Hilfe für den Benutzer verstanden werden, so ist doch die Hilfe durch Feedback implizit, während Online-Hilfe explizit ist ([Sellen und Nicol, 1991], Seiten 151-152). Eine eindeutige Zuordnung zu entweder Online-Hilfe oder Statusanzeige ist also in jedem Fall möglich.

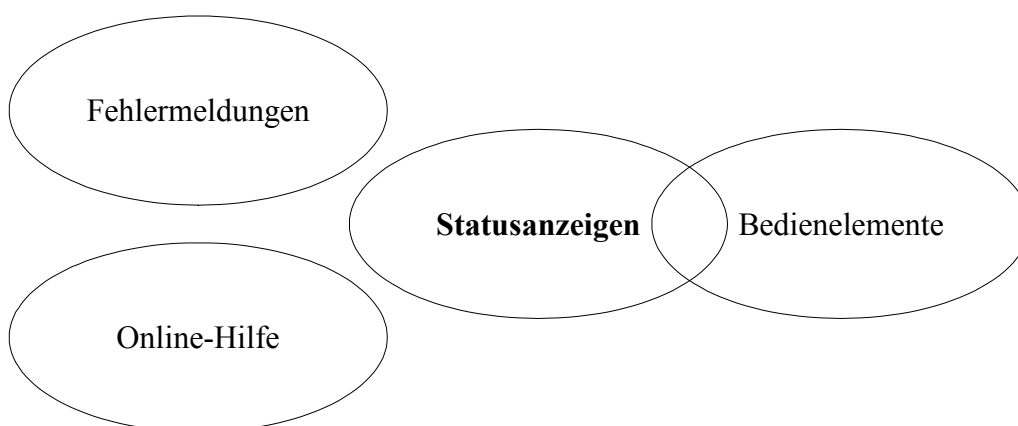


Abbildung 2: Abgrenzung zu anderen Bereichen

Eine Statusanzeige ist demnach permanent sichtbar bzw. kann jederzeit aufgerufen werden, nicht nur in Ausnahmesituationen. Sie bietet höchstens implizite Hilfe an und kann auch als Bedienelement ausgeführt sein.

## 2.4 Einordnung in die Software-Ergonomie

Für den Versuch einer Einordnung von Statusanzeigen in die Software-Ergonomie muss Software-Ergonomie erst einmal definiert werden:

*Im Sinne von Ergonomie als Wissenschaft von der Anpassung von Technik an ihre Benutzer beschäftigt sich Software-Ergonomie disziplinübergreifend speziell mit der benutzergerechten Gestaltung der Mensch-Computer-Interaktion (MCI), d.h. mit der Gestaltung der Teile eines interaktiven Computersystems, die von Software gesteuert werden und an der sogenannten Benutzungsoberfläche wirksam werden. Dabei sind die Benutzungsoberflächen, mit denen die Benutzer unmittelbar interagieren, ein wichtiger Gegenstand der Gestaltung. ([Eberleh, Oberquelle und Oppermann 1994], Seite 1)*

Eine benutzergerechte Gestaltung von Software berücksichtigt vier Aspekte ([Eberleh, Oberquelle und Oppermann 1994], Seiten 1 f.):

- Menschengerechte Gestaltung: Berücksichtigung der Benutzer als Menschen mit Stärken, Schwächen, Bedürfnissen, Unterschieden und individueller Entwicklung
- Aufgabenangemessene Gestaltung: Arbeitsgestaltung, aufgabenangemessene Funktionalität.
- Technikbewusste Gestaltung: Einsetzen neuer technischer Möglichkeiten zum Wohl des Benutzers.
- Organisationsgerechte Gestaltung: Rücksichtnahme nicht nur auf Einzelbenutzer und ihre Aufgaben, sondern auch auf ihre organisatorische Einbindung.

Nach meiner Definition aus Kapitel 2.2, *Definition von Status und Statusanzeigen*, sollen Statusanzeigen den Benutzer sowohl über den Programmablauf informieren als auch über den momentanen Modus des Programms. Mit dieser Definition lassen sich Statusanzeigen keinem der vier Aspekte benutzergerechter Software-Gestaltung eindeutig zuordnen, sondern sie haben vielmehr mit allen vier Punkten zu tun:

- Bei der Gestaltung einer Statusanzeige muss auf die physiologischen und kognitiven Eigenheiten des Menschen Rücksicht genommen werden (Auge, Ohr, Signalverarbeitung).
- Statusanzeigen sollen dem Benutzer durch Vermitteln von Informationen bei der Erledigung seiner Aufgabe unterstützen.
- Die technischen Gegebenheiten des Systems müssen berücksichtigt werden, beispielsweise die Frage, welche Hardware vorhanden ist, oder wie viel Bildschirmfläche zur Verfügung steht.

- Statusanzeigen können auch über den „Zustand“ anderer Benutzer eines verteilten Systems informieren. Es müssen bei ihrer Gestaltung also auch organisatorische Gegebenheiten berücksichtigt werden.

Auch die Betrachtung von Rückmeldungen in der Literatur zeigt, dass der komplette Umfang von Statusanzeigen nicht vollständig einem einzelnen Punkt zuzuordnen ist. Die Nievergeltschen Fragen beispielsweise behandeln nur einen Teil der Fragen über den Status eines Systems ([Nievergelt, 1983], Seite 22):

- *Wo bin ich?*  
(Der Bildschirm sieht nicht wie erwartet aus)
- *Was kann [ich] hier tun?*  
(Die Menge der aktiven Befehle ist unklar)
- *Wie kam ich hierhin?*  
(Vielleicht wurde eine falsche Taste gedrückt)
- *Wo kann ich hin und wie komme ich dorthin?*  
(Die Möglichkeiten des Systems sollen erforscht werden).

Die Nievergeltschen Fragen betrachten zwei Bereiche, nämlich die Navigation in einem Programm (Fragen 1, 2 und 4) sowie den aktuellen Modus des Programms (Frage 2, zum Teil auch Frage 4). Die Information über den Prozessfortschritt ist hierin nicht enthalten.

Auch Oppermann und Reiterer erfassen elf Jahre danach mit ihrer Definition von Rückmeldung nicht den vollständigen Umfang von Statusanzeigen. Sie unterscheiden zwei Arten von Rückmeldungen:

*Rückmeldungen im Zuge der Aufgabenerfüllung bewirken, dass der Arbeitende Meldungen über den laufenden Fortgang seines Arbeitsablaufes (Ablauf-Feedback) und das Resultat seiner Arbeit (Resultat-Feedback) erhält.*  
([Oppermann und Reiterer, 1994], Seite 339)

Nach dieser Definition gehört ein Teil der Statusanzeigen zum Ablauf-Feedback, nämlich die Anzeigen von dynamischen Prozessen. Die Anzeigen von statischen Zuständen (Programmmodi) sind mit dieser Definition jedoch nicht erfasst.

Statusanzeigen vermitteln dem Benutzer Informationen. Damit sind sie Bestandteil vom Feedback des Systems an den Benutzer. Feedback ist für [Herzer und Noll, 1990] ein Teil der Mensch-Computer-Interaktion, welche sie als geregelten Handlungskreis verstehen (siehe hierzu in dieser Arbeit Kapitel 3.6, *Menschliches Handeln*). Feedback findet auf der kompletten Systemoberfläche statt, zu der nicht nur die Darstellungen auf dem Bildschirm gehören, sondern alle gestalteten System-

komponenten, die der Benutzer wahrnehmen kann – also beispielsweise auch Leuchtdioden am Drucker ([Herzer und Noll, 1990], Seite 1). Abweichend hiervon möchte ich den Begriff etwas weiter fassen. Wie ich später in Kapitel 5, *Statusanzeigen in Literatur und Praxis*, zeigen werde, kann der Benutzer auch von nicht gestalteten Systemausgaben wie beispielsweise Betriebsgeräuschen profitieren, die ich daher der Einfachheit halber ebenfalls als Feedback betrachte.

Durch das vom Systementwickler gestaltete Feedback und damit auch durch Statusanzeigen wird unter anderem Einfluss darauf genommen,

- *wie sich das System dem Benutzer darstellt,*
- *welche Einstellung der Benutzer zum System entwickelt,*
- *wie oft das System benutzt wird,*
- *wie mit dem System umgegangen wird,*
- *wie die Arbeitsaufgabe ausgeführt wird [...]. ([Herzer und Noll, 1990], Seite 1)*

Feedback ist also ein wesentlicher Punkt der Mensch-Computer-Interaktion. Das gilt dann natürlich auch für Statusanzeigen, die zum Feedback gehören.

## **3 Wahrnehmung und Informationsverarbeitung**

Die Darstellung von Statusinformation erfolgt für den Menschen. Erst seine Fähigkeiten, sensorische Reize schnell zu interpretieren und daraufhin komplexe Aktionen durchzuführen, macht interaktive Systeme möglich. Nun unterliegt das sensorische System des Menschen jedoch etlichen biologisch begründeten Einschränkungen. Je nach Art des sensorischen Reizes (also je nach Art des benutzten bzw. angesprochenen Sinns) sind bestimmte Regeln zu beachten – bei optischen Reizen beispielsweise unter anderem bezüglich der Farbwahl. Das Wissen um Wahrnehmung und Informationsverarbeitung durch den Menschen ist eines der Fundamente, auf dem der Designer eines interaktiven Systems aufbaut ([Shneiderman, 1998], Seite 20).

Von den verschiedenen menschlichen Sinnen (Sehen, Hören, Fühlen, Riechen und Schmecken) sind Sehen und Hören die für die Interaktion mit dem Computer wichtigsten Sinne. Von den übrigen Sinnen wird lediglich das Fühlen für Rückmeldungen von Computern eingesetzt. In der Praxis ist das bisher jedoch kaum relevant, taktile Rückmeldung wird nur selten verwendet und dann auch nur in wenigen Bereichen, überwiegend in virtuellen Realitäten und bei einigen Computerspielen. Daher zeige ich nur für Sehen und Hören, wie die Signalaufnahme funktioniert und welche Einschränkungen sich daraus ergeben.

Nach der Aufnahme eines Signals wird dieses verarbeitet. Wichtige Aspekte hierbei sind das Kurzzeitgedächtnis, das Langzeitgedächtnis mit der Fähigkeit zu lernen, Problemlösung und Entscheidungsfindung ([Shneiderman, 1998], Seite 20). Auch hier ergeben sich physiologisch bedingte Einschränkungen. Daher betrachte ich neben dem sensorischen System auch noch das für die Verarbeitung der empfangenen Reize zuständige Gedächtnis.

Psychologische Aspekte bei der Verarbeitung optischer oder akustischer Signale werden nicht beim eigentlich für die Signalverarbeitung zuständigen Gedächtnis, sondern bei dem jeweiligen Sinn betrachtet.

### **3.1 Begriffe**

#### **Wahrnehmung**

[Galitz, 1988] definiert Wahrnehmung als Bewusstsein und Verständnis der Dinge unserer Umwelt durch physikalische Reize verschiedener Art. Sie wird zum Teil beeinflusst durch die erworbenen Erfahrungen: Empfangene Reize werden aufgrund von im Gedächtnis gespeicherten Modellen klassifiziert. Auf diese Art erlangen wir Verständnis ([Galitz, 1988], Seite 6).

## Signale und Rauschen

Das sensorische System des Menschen empfängt ständig eine Vielzahl von Reizen. Wichtige Reize heißen *Signale*, unwichtige Reize heißen *Rauschen*. Vorhandenes Rauschen beeinflusst die Wahrnehmung der Signale: Signale werden schneller wahrgenommen, wenn sie einfach vom Rauschen unterschieden werden können. Im schlimmsten Fall kann ein Signal vom Rauschen verdeckt werden. Die Klassifizierung eines konkreten Geräusches als Signal oder Rauschen ist situationsabhängig: In einem Kontext ist ein Geräusch ein Signal, in einem anderen Kontext ist es bloß Rauschen ([Galitz, 1988], Seite 16).

## 3.2 Optische Wahrnehmung (Sehen)

Sehen ist ein informationsverarbeitender Prozess, der aus Bildern eine Beschreibung der Umwelt rekonstruiert. Bei diesem Prozess werden Annahmen über die Umwelt mit einbezogen ([Mallot, 1998], Seite 13). Die Leistungsfähigkeit eines Seh-Systems hängt von den Fähigkeiten zur Analyse optischer Signale wie Intensität und Wellenlänge und deren Zuordnung zu Helligkeit und Farbe ab ([Brockhaus, 1993], Band 20, Seite 61). Für optische Nachrichten müssen die Physiologie des Auges, der Prozess des Erkennens optischer Nachrichten sowie die Art menschlichen Denkens berücksichtigt werden ([Herzer und Noll, 1990], Seite 46).

### Informationsaufnahme

Eintreffendes Licht wird vom dioptrischen Apparat des Auges, bestehend aus Hornhaut, Augenkammern, Iris, Linse und Glaskörper, gebrochen und trifft auf die lichtempfindliche Retina, wo ein verkleinertes und umgekehrtes Bild erzeugt wird. Der größte Teil der Lichtbrechung erfolgt durch die Hornhaut, der kleinere durch die Linse. Licht unterschiedlicher Wellenlänge wird von Hornhaut und Linse unterschiedlich stark gebrochen, so dass ein blauer Gegenstand weiter entfernt sein muss als ein roter, damit beide gleichzeitig scharf gesehen werden. Die Retina registriert Intensität, Wellenlänge und räumliche Verteilung des einfallenden Lichts mit Hilfe von Photorezeptoren. Die Photorezeptoren empfangen Licht der Wellenlänge zwischen 350 nm bis 750 nm, das sogenannte sichtbare Licht. Die maximale Empfindlichkeit liegt bei etwa 560 nm. Es gibt zwei Arten von Photorezeptoren, nämlich Stäbchen und Zapfen. Der Mensch hat etwa 120 Mio. Stäbchen, die Hell-Dunkel-Unterschiede wahrnehmen, und etwa 6 Mio.

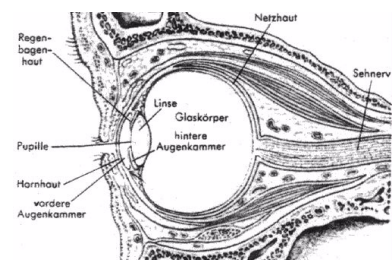


Abbildung 3: Schnitt durch die Augenhöhle (nach [Brockhaus, 1993], Band 2, Seite 318)



Zapfen, die Farbe erkennen. Es gibt drei verschiedene Typen von Zapfen, die Licht unterschiedlicher Wellenlängen empfangen. Obwohl der menschliche Blickwinkel beinahe  $180^\circ$  beträgt, wird lediglich in einem Bereich von etwa  $2^\circ$  um das Zentrum herum scharf gesehen. Dieser Bereich, in dem sich nur Zapfen in großer Dichte befinden, heißt Fovea. Ein fixierter Punkt wird in beiden Augen auf der Fovea abgebildet. Der übrige Bereich der Retina liefert peripheres Sehen, das zur Orientierung dient ([Brockhaus, 1993], Band 2, Seiten 318-319; [Brockhaus, 1993], Band 20, Seite 62; [Card, Moran und Newell, 1983], Seite 25; [Mallot, 1998], Seiten 29, 48, 98; [Oberquelle, 1997], Seite W-5).

Das Auge kann sich durch Adaption auf Entfernungen einstellen. Dabei wird die Linse mit Hilfe von Muskeln gedehnt oder gestreckt. Nah- und Fernpunkt bestimmen den Bereich, innerhalb dessen scharfes Sehen möglich ist. Mit Hilfe der Pupille, einer veränderlichen Öffnung vor der Linse, sowie durch Anpassen der Empfindlichkeit der Photorezeptoren stellt sich das Auge auf unterschiedliche Helligkeiten ein. Durch Augenbewegungen, ggf. unterstützt durch Kopf- und weitere Körperbewegungen, stellt sich das Auge auf Winkelpositionen ein. Augenbewegungen sind rasche kontinuierliche Ruckbewegungen (Saccaden). Der Fixationszeitraum beträgt zwischen 0,15 s und 2 s. Der Übergang von einem Punkt des Interesses zum nächsten dauert etwa 30 ms. Ist der neue Punkt des Interesses mehr als etwa  $30^\circ$  entfernt, wird der Kopf bewegt, um den Winkel des Auges zu reduzieren. Zwischen 200 und 600 Grad pro Sekunde können so betrachtet werden. Akkomodation, Pupillenänderungen und Augenbewegungen können sowohl willkürlich als auch unwillkürlich ausgelöst werden. Sind Augenbewegungen notwendig, können sie der dominierende Faktor für die benötigte Zeit zur Ausführung einer Aufgabe sein. Zentrales Sehen, peripheres Sehen, Augenbewegung und Kopfbewegung arbeiten als integriertes System zusammen, um dem Menschen eine kontinuierliche visuelle Repräsentation zu liefern ([Brockhaus, 1993], Band 20, Seite 62; [Card, Moran und Newell, 1983], Seiten 25-28, 50; [Mallot, 1998], Seite 48; [Tovée, 1996], Seiten 32-34). Ein Nebeneffekt der unterschiedlichen Pupillenöffnungen ist, dass der Schärfebereich sich verändert. Bei hellem Licht und damit kleiner Öffnung wird in einem größeren Bereich in der Tiefe scharf gesehen. Je dunkler es ist und je größer die Öffnung der Pupille ist, desto geringer in der Tiefe ist der Bereich, in dem scharf gesehen wird.

Das Anpassen der Empfindlichkeit der Photorezeptoren wird durch die Menge lichtempfindlicher Pigmente im Rezeptor gesteuert. Je mehr Pigmente vorhanden sind, desto empfindlicher wird der Rezeptor. Da der Aufbau von Pigment langsamer als der Abbau vonstatten geht, ist der Zeitbedarf für die Adaption auf geänderte Lichtverhältnisse asymmetrisch: Die Adaption von hellem Licht auf dunkles Licht dauert

30 bis 45 Minuten, während die Adaption von Dunkel nach Hell in der Regel weniger als 1 Sekunde, maximal eine Minute, dauert. Durch die recht lange Adaption von Hell nach Dunkel verschmelzen ab einer bestimmten Frequenz einzelne Bilder zu einer Sequenz. Diese Frequenz ist abhängig von der Lichtstärke. Bei dunklem Licht kann bereits 6 Hz als flimmerfrei empfunden werden, während bei hellem Licht die Grenze bei 60 Hz liegt ([Frumkes, 1990], Seite 181; [Oberquelle, 1997], Seite W-9; [Tovée, 1996], Seiten 32-34).

Das Auflösungsvermögen des menschlichen Auges hängt vom Pupillendurchmesser und der Wellenlänge des Lichts ab. Es liegt im Bereich von etwa 30 Bogensekunden. Das entspricht in etwa dem fovealen Zapfenabstand. Weil aufgrund von Abbildungsunschärfe stets mehrere Rezeptoren beteiligt sind, wird die Auflösung des menschlichen Auges durch die entstehende Überauflösung auf etwa 10 Bogensekunden verbessert ([Mallot, 1998], Seiten 67, 70).

Im Alter ändern sich die Eigenschaften des menschlichen Auges. Die Hornhaut wird trübe. Auch der Nahpunkt verschiebt sich mit dem Alter, da sich Größe, Form und Flexibilität der Linse ändern. Während er bei einem 10-jährigen bei 7 cm liegt, liegt er bei einem Menschen Mitte Vierzig bereits zwischen 50 cm und 1 m. Bei noch älteren Menschen liegt er bei mehreren Metern ([Tovée, 1996], Seiten 17, 20-21). Ältere Menschen können feine Details auf dem Bildschirm also nicht mehr so gut erkennen wie jüngere.

### Sehschärfe

Je weiter ein Objekt von dem Punkt entfernt ist, den das Auge momentan fixiert, desto weniger deutlich tritt es hervor. Untersuchungen zeigen, dass die Sehschärfe bei einem Winkel von etwa  $2,5^\circ$  vom Punkt, auf den das Auge fixiert ist, etwa halbiert ist. Ein Bereich von etwa  $5^\circ$  um den Punkt, auf den das Auge fixiert ist, kann als „Nahbereich“ angesehen werden. Bei einem Abstand zum Bildschirm von 47,5 cm entspricht das einem Radius auf dem Bildschirm von etwa 41,8 mm. Bei durchschnittlicher Schriftgröße kann damit ein Kreis von etwa sieben Zeilen und 15 Zeichen scharf gesehen werden, entsprechend Abbildung 4. Die Empfindlichkeit des Auges nimmt zu, je näher die Buchstaben am Zentrum sind, zum Rand hin nimmt sie ab. Die Wahrscheinlichkeit, dass ein Zeichen am Rand noch korrekt erkannt wird, liegt bei 50% ([Galitz, 1988], Seiten 7-18).

3213123  
54321212345  
6543211123456  
765432101234567  
6543211123456  
54321212345  
3213123

*Abbildung 4: Bereich scharfen Sehens auf dem Bildschirm. Die 0 ist der Mittelpunkt des momentanen Fokuss. ([Galitz, 1988], Seite 17)*

## **Kanten**

*Kanten sind Orte im Bild, an denen sich die Intensitäten abrupt oder wenigstens „schnell“ ändern. ([Mallot, 1998], Seite 75)*

Kanten treten bevorzugt an Objektgrenzen auf. Sie werden vom Betrachter zur Einteilung von Bildern in Segmente verwendet, von denen angenommen wird, dass sie Bilder separater Objekte sind. Menschen setzen also voraus, dass die Umwelt aus Objekten mit gleichbleibender Reflexivität besteht, deren Bilder durch Kanten begrenzt sind. Das menschliche Auge verstärkt an Kanten den gesehenen Kontrast. Im Gegensatz zu dem verstärkt gesehenen Kontrast bei Kanten ist der gesehene Kontrast bei gleichmäßigen Verläufen schwächer als der wirkliche ([Mallot, 1998], Seiten 76-78).

## **Farbwahrnehmung**

Farbe ist keine physikalische Eigenschaft von Objekten, sondern eine Sinnesempfindung. Daher kann Farbe nicht physikalisch gemessen werden. Ein Farbeindruck entsteht, wenn sichtbares Licht auf die Photorezeptoren der Netzhaut trifft und eines der möglichen Erregungsmuster in den Rezeptoren erzeugt. Licht verschiedener Spektren kann den selben Farbeindruck hervorrufen – wichtig ist alleine, dass das selbe Erregungsmuster erzeugt wird. Im Gegensatz zur Fotografie kann der Mensch vom Frequenzspektrum der Lichtquelle abstrahieren. Daher erscheinen Objekte bei unterschiedlicher Beleuchtung stets mit etwa der selben Farbe. Dieses Phänomen heißt *Farbkonstanz*. Der Eindruck, den Licht mit einem bestimmten Spektrum hervorruft, hängt auch sehr davon ab, welche Farben gleichzeitig oder kurz zuvor gesehen wurden. Zwei Objekte auf verschiedenen Hintergründen, die den selben Farbeindruck hervorrufen, sehen nebeneinander gelegt sehr unterschiedlich aus ([Brockhaus, 1993], Band 7, Seiten 106-107; [Mallot, 1998], Seiten 9, 96-98, 112).

Farben werden unterschieden in bunte (Rot, Gelb, ...) und unbunte Farben (Weiß, Grautöne, Schwarz). Unbunte Farben können eindeutig durch ihre Helligkeit beschrieben werden. Bei bunten Farben müssen drei voneinander unabhängige Merkmale angegeben werden ([Brockhaus, 1993], Band 7, Seite 107).

Da es drei verschiedene Typen von Zapfen mit unterschiedlichen Empfindlichkeitsmaxima gibt, kann Licht aus drei Grundfarben gemischt werden. Zu unterscheiden sind additive Farbmischung (die Überlagerung von Licht der verschiedenen Einzel Farben) und subtraktive Farbmischung (das Zusammenfügen von Pigmenten, deren Absorptionsspektren sich überlagern) ([Mallot, 1998], Seite 97-99).

Aufgrund der physiologischen Eigenschaften des Auges unterstützen bestimmte Farben die Wahrnehmung, andere wiederum bereiten Probleme. So sollten wegen der unterschiedlichen Brechung nicht Rot und Blau gesättigt nebeneinander verwendet werden. Auch wenn die psychophysische Empfindlichkeit für rotes Licht etwa 4,5 mal so groß wie für blaues Licht ist (wobei Intensität und Dauer des Reizes diesen Faktor beeinflussen), so sollten für Objekte im peripheren Sichtfeld dennoch Blau oder Gelb genommen werden, da das Auge am Rand für diese Farben am empfindlichsten ist. Zusätzlich zu diesen für alle Menschen gleichen physiologischen Eigenschaften sind jedoch auch noch individuelle physiologische Aspekte zu berücksichtigen – verschiedene Menschen haben also unterschiedliche Fähigkeiten zur Farberkennung. Folgende individuelle Faktoren können die Farberkennung beeinflussen:

- Ältere Menschen haben eine niedrigere Empfindlichkeit für Farbe und benötigen zum Erkennen von Farbe oft eine stärkere Helligkeit. Insbesondere lässt die Empfindlichkeit des Menschen für Licht kurzer Wellenlänge (Blau) nach. Ältere Menschen verlieren dadurch die Fähigkeit, verschiedene Blautöne zu unterscheiden.
- Acht Prozent aller Männer und 0,5% aller Frauen sind farbenblind. Am häufigsten ist eine Rot-Grün-Blindheit.
- Kulturelle Unterschiede können die Anzahl und die Kategorien der von einem Individuum wahrnehmbaren Farben beeinflussen. Farbnamen können in verschiedenen Kulturen verschiedene Schattierungen umfassen.
- Auch der ausgeübte Beruf entscheidet, ob ein Mensch subtile Farbunterschiede wahrnehmen kann oder nicht.

([Galitz, 1988], Seiten 205, 207, 212; [Mallot, 1998], Seiten 98-99; [Salomon, 1991], Seiten 270-271; [Shneiderman, 1998], Seite 401)

### **Objekterkennung**

Der Mensch teilt das Gesehene Bild in Regionen ein, von denen er annimmt, dass sie zum gleichen Objekt gehören. Er setzt dabei voraus, dass die Welt in Objekte eingeteilt werden kann. Diese Einteilung funktioniert auch bei nur teilweise sichtbaren Objekten. Durch die Verdeckung erhält der Mensch zusätzlich Informationen über die räumliche Tiefe ([Mallot, 1998], Seite 11).

Menschen erkennen nicht alle möglichen Objekte gleich gut. Einige Objekte werden besser erkannt als andere. Dieses psychologische Phänomen wird durch eine Sammlung von Regeln, die sogenannten Gestaltgesetze, beschrieben ([Glaser, 1994], Seiten

25-27; Abbildung 5 bis Abbildung 9 sind Ausschnitte aus [Glaser, 1994], Abb. 1-6, Seite 26):

- *Gesetz der Nähe*. Objekte, die näher beieinander sind, werden bevorzugt als Einheit wahrgenommen. In Abbildung 5 beispielsweise sind oben zwei Gruppen von Punkten und links drei Reihen zu erkennen, während die rechte Gruppe eine schwache Ahnung von Spalten zeigt. Die Abstände zwischen den Punkten sind aber insgesamt in der Horizontalen und der Vertikalen zu ähnlich, um eindeutig waagerechte oder senkrechte Linien erkennen zu können.



Abbildung 5: Gesetz der Nähe

- *Gesetz der Ähnlichkeit oder Gleichheit*. Je ähnlicher Objekte einander sind, desto bevorzugter werden sie als Einheit wahrgenommen. Daher sind in Abbildung 6 je drei Reihen dunkler und heller Quadrate zu erkennen.

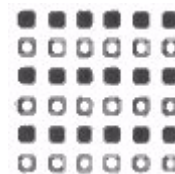


Abbildung 6: Gesetz der Ähnlichkeit oder Gleichheit

- *Gesetz der guten Fortsetzung*. Sich schneidende Konturen werden so interpretiert, dass die enthaltenen Linien möglichst wenig geknickt oder gebogen sind.

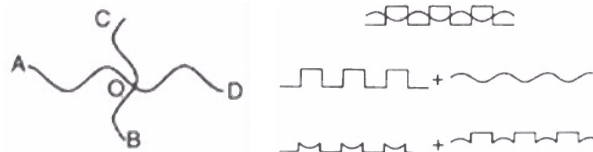


Abbildung 7: Gesetz der guten Fortsetzung

- *Gesetz der Schließung*. Fast geschlossene Konturen werden tendenziell ganz geschlossen wahrgenommen. Das Innere wird dabei als Figur, das Äußere als Hintergrund interpretiert. In Abbildung 8 sieht der Benutzer oben einen geschlossenen Kreis, darunter zwei sich überlappende Rechtecke. Bei der unteren Figur konkurrieren das Gesetz der Schließung und das Gesetz der guten Fortsetzung miteinander, ob drei Trapeze oder ein Rechtecksignal, überlagert von einer bogenförmigen Linie, erkannt werden.

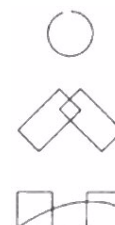


Abbildung 8: Gesetz der Schließung

- *Gesetz der Symmetrie.* Zwischenräume zwischen symmetrischen Konturen werden unter ansonsten gleichen Voraussetzungen eher als Figur wahrgenommen als Zwischenräume zwischen asymmetrischen Konturen. In Abbildung 9 sind daher bevorzugt drei Objekte zu erkennen: Ein schwarzes Objekt mit einer durchbrochenen Mittellinie (die beiden linken schwarzen Streifen), ein schwarzes Objekt, das von zwei gepunkteten Linien begrenzt wird (die beiden mittleren schwarzen Streifen), sowie der weiße Zwischenraum in der Mitte.

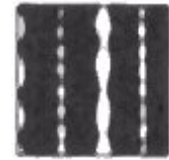


Abbildung 9:  
*Gesetz der  
Symmetrie*

Die Wahrnehmung aufgrund der Gestaltgesetze interpretiert das Gesehene so, dass die analytisch-geometrische Beschreibung so wenig Informationen wie möglich benötigt. [Herzer und Noll, 1990] (Seite 46) führen noch einige weitere Regeln für die Objekterkennung auf:

- Der Betrachter versucht immer, eine Figur auf einem Hintergrund zu erkennen.
- Linien um eine Fläche bilden mit dieser eine Einheit.
- Einfache Objekte werden eher erkannt als komplizierte.
- Bekannte Objekte werden eher erkannt als unbekannte.
- Je mehr sich ein Objekt von den anderen abhebt, desto leichter wird es erkannt.

Sollen bewegte Objekte erkannt werden, müssen sowohl zeitliche als auch örtliche Änderungen im Bild ausgewertet werden ([Mallot, 1998], Seite 11).

### **Folgerungen für die Gestaltung von Statusanzeigen**

Aus den diversen physiologischen und psychologischen Eigenschaften des Menschen bezüglich optischer Wahrnehmung kann die folgende Liste von Empfehlungen generiert werden. Diese Empfehlungen gelten natürlich nicht nur für Statusanzeigen allein, sondern für jede Art optischer Ausgabe.

- Scharfes Sehen ist nur in einem sehr geringen Radius um den Fokus herum möglich. Daher sollten im peripheren Gesichtsfeld keine feinen Strukturen verwendet werden, insbesondere keine Texte.
- Rot und Blau werden unterschiedlich gebrochen und sollten daher nicht direkt nebeneinander verwendet werden. Insbesondere feine blaue Strukturen auf rotem Grund sind zu vermeiden. Rote Objekte auf blauem Grund wirken hervorgehoben (näher), diese Kombination kann für gewisse Zwecke ihre Berechtigung haben.
- Im peripheren Sichtbereich werden Bewegungen leicht erkannt. Animationen am Bildschirmrand werden dadurch gut wahrgenommen.

- Im äußersten peripheren Sichtbereich gibt es keine Farberkennung. Daher kann nicht darauf vertraut werden, dass Farbe am Bildschirmrand ohne Fokussierung erkannt wird.
- Im peripheren Sichtbereich werden blaue oder gelbe Objekte bevorzugt erkannt.
- Der gesehene Kontrast ist an Kanten stärker. Sehr dünne Linien in einer kontrastierenden Farbe am Rand einer Fläche helfen daher beim Erfassen der Fläche.
- Damit die Farbe eines Objekts wie beabsichtigt erkannt wird, muss auf die Farben in seiner unmittelbaren Umgebung geachtet werden, da diese die Wahrnehmung beeinflussen können.
- Insbesondere bei einer Zielgruppe mit vielen älteren Menschen sollten nicht viele ähnliche Blautöne nebeneinander zur Unterscheidung verwendet werden.
- Auf Farbenblindheit, insbesondere auf Rot-Grün-Blindheit, ist Rücksicht zu nehmen. Daher sollten Statusänderungen nicht durch einen Rot-Grün-Wechsel dargestellt werden.
- Einfache und bekannte Objekte werden bevorzugt erkannt. Statusanzeigen sollten also einfache und bekannte Formen haben.
- Ein Objekt wird leichter erkannt, wenn es sich von anderen Objekten abhebt. Anzeigen, die wichtige Informationen vermitteln, sollten diese Tatsache nutzen.

### 3.3 Akustische Wahrnehmung (Hören)

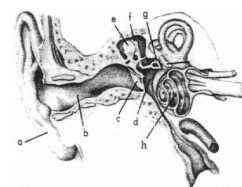
Wichtig für den Eindruck beim Hören sind Frequenzspektrum und Intensität der Schallwellen. Damit akustische Nachrichten verstanden werden können, müssen bei ihrer Gestaltung die Physiologie des Ohres sowie der Prozess der Geräuscherkennung und die Art menschlichen Denkens berücksichtigt werden ([Brockhaus, 1993], Band 10, Seite 241; [Herzer und Noll, 1990], Seite 51).

#### Informationsaufnahme

Die Schallwellen gelangen durch den äußeren Gehörgang zum Trommelfell, das die Schallenergie aufnimmt. Am Trommelfell hängt eine Kette von Gehörknöchelchen.

Trommelfell und die Kette der Gehörknöchelchen werden von den auftreffenden Schallwellen zu Schwingungen angeregt.

Die Schallwellen werden mechanisch verstärkt und auf die mit Lymphe gefüllte Schnecke übertragen.



Ohr: Schematische Übersicht des ganzen Ohres: a Ohrmuschel, b äußerer Gehörgang, c Trommelfell, d Paukenhöhle, e Hammer, f Amboß, g Steigbügel, h Schnecke

Abbildung 10: Schematische Übersicht des Ohrs (nach [Brockhaus, 1993], Band 16, Seite 135)

füllte Schnecke übertragen. In der Schnecke werden sie von feinen Härchen aufgenommen und an die Nerven weitergeleitet. Das Ohr ist nichtlinear, es erzeugt Verzerrungen ([Brockhaus, 1993], Band 10, Seite 242; [Zwicker und Feldtkeller, 1967], Seite 218).

### **Begriffe**

*Ein Klang ist ein Dauerschall, der sich aus mehreren Teiltönen zusammensetzt. ([Zwicker und Feldtkeller, 1967], Seite 4)*

Zwei Töne mit unterschiedlichen, nahe beieinander liegenden Frequenzen, verstärken sich je nach momentaner Phasenlage gegenseitig oder schwächen sich gegenseitig ab. Dadurch wird der resultierende Ton, zu dem die Einzeltöne verschmelzen, abwechselnd laut und leise gehört. Dieses Phänomen heißt *Schwebung* ([Zwicker und Feldtkeller, 1967], Seite 12).

Sinustöne und Klänge bestehen aus einzelnen Spektrallinien. Schallereignisse, die kontinuierliche Spektren enthalten, heißen *Geräusche*. Ein Geräusch, das zu keinem Zeitabschnitt eine Tonhöhe oder einen Rhythmus hat, heißt *Weißes Rauschen* ([Zwicker und Feldtkeller, 1967], Seite 13).

Menschen können Geräusche voneinander unterscheiden, selbst wenn sie gleichzeitig aufgenommen werden. Diese Fähigkeit wird jedoch beeinträchtigt, wenn die Geräusche zu ähnlich sind, oder wenn es zu viele Geräusche auf einmal sind. Änderungen eines Geräuschs werden besser erkannt, als verschiedene Töne unterschieden werden können ([Herzer und Noll, 1990], Seite 51).

### **Frequenz und Lautstärke**

Menschen hören Töne ab etwa 16 Hz bis maximal 21.000 Hz. Die maximale noch hörbare Frequenz ist altersabhängig und sinkt allmählich auf 12.000 Hz. Die Empfindlichkeit ist von der Frequenz abhängig. Bei jungen Menschen ist sie zwischen 2.000 Hz und 4.000 Hz am größten, bei älteren Menschen bei etwa 1.000 Hz. Die Empfindlichkeit für 10 kHz ist bei einem 20-jährigen etwa 20 dB niedriger als bei einem 60-jährigen. Der Mensch kann Schalldrücke zwischen  $10^{-4}\mu\text{b}$  und  $10^3\mu\text{b}$  wahrnehmen. Zur leichteren Handhabung wird der Schalldruck logarithmisch in den in dB gemessenen Schallpegel umgerechnet ([Herzer und Noll, 1990], Seite 51; [Zwicker und Feldtkeller, 1967], Seiten 1, 3, 55).

Die Fähigkeit zur Unterscheidung verschiedener Tonhöhen ist abhängig von Frequenz und Lautstärke. Der dazu notwendige prozentuale Unterschied der Frequenz ist bei tiefen Tönen größer. 3.000 bis 4.000 verschiedene Töne können nach Ton-



höhe, etwa 300.000 verschiedene Töne nach Tonhöhe und Lautstärke unterschieden werden. Das Gehör ist bei großen Pegeln empfindlicher gegen Amplitudenschwankungen als bei kleinen. Die wahrnehmbare Lautstärkeänderung ist außerdem abhängig von der Frequenz. Das Erkennen der Lautstärke dauert länger als das Erkennen der Tonhöhe ([Brockhaus, 1993], Band 10, Seite 243; [Zwicker und Feldtkeller, 1967], Seiten 66-68, 95).

### **Räumliches Hören**

Dank zweier Ohren kann räumlich gehört werden. Laufzeitunterschiede ab 0,03 ms erzeugen eine Richtungsinformation. Eine langsame Änderung der Lautstärke wird als Veränderung der Entfernung gedeutet. Auch unterschiedliche Frequenzspektren spielen eine Rolle beim räumlichen Hören ([Brockhaus, 1993], Band 10, Seite 243; [Herzer und Noll, 1990], Seite 51).

### **Verdeckung von Tönen**

Ein lauter Ton verdeckt gleichzeitig erklingende leisere Töne. Aber nicht nur gleichzeitig erklingende Töne können einander verdecken. Ein lauter Ton verdeckt ebenso einen kurze Zeit später erklingenden leiseren Ton, wobei der verdeckende Ton einen Einfluss bis etwa 200 ms nach seinem Ende hat. Ein Ton kann auch von einem kurz darauf folgenden lauterem Störgeräusch verdeckt werden. Der Einfluss eines Störgeräuschs ist bis etwa 20 ms vor seiner Entstehung feststellbar ([Zwicker und Feldtkeller, 1967], Seiten 63, 206, 210).

### **Folgerungen für die Gestaltung von Statusinformationen**

Auch aus den diversen physiologischen und psychologischen Eigenschaften des Menschen bezüglich akustischer Wahrnehmung kann eine Liste von Empfehlungen generiert werden. Auch diese Empfehlungen gelten natürlich nicht nur für Statusanzeigen allein, sondern für jede Art akustischer Ausgabe.

- Die Empfindlichkeit für hohe Frequenzen lässt mit dem Alter nach. Es sollten daher möglichst keine Töne über 2.000 Hz verwendet werden.
- Die höchste Empfindlichkeit liegt im Bereich der für Sprache wichtigen Frequenzen. Diese sollten bevorzugt verwendet werden.

- Die verwendeten Töne sollten sich deutlich voneinander unterscheiden. Insbesondere muss berücksichtigt werden, dass die Fähigkeit zur Unterscheidung von Tonhöhen und Lautstärke mit sinkendem Schallpegel nachlässt. Daher sollten am besten Oktavschritte verwendet werden. Die Lautstärke sollte in Schritten von mindestens 3 dB variiert werden.
- Damit keine Information wegen Verdeckung verloren geht, sollten einzelne Töne mindestens 250 ms Abstand voneinander haben.

### 3.4 Menschliche Signalverarbeitung

Die Signalverarbeitung findet im Gedächtnis des Menschen statt. Dieses kann in Kurzzeit- oder Arbeitsgedächtnis und Langzeitgedächtnis unterschieden werden. Das Kurzzeit- oder Arbeitsgedächtnis enthält die aktuell für die Verarbeitung benötigten Informationen. Das Langzeitgedächtnis speichert Inhalte (deklaratives Gedächtnis), Assoziationen (assoziatives Gedächtnis) und Handlungsabläufe (prozedurales Gedächtnis) ([Mallot, 1998], Seite 5). Tätigkeiten wie Nachdenken, Vortragen oder Zuhören führen zu fortwährendem Löschen und Überschreiben des Kurzzeitgedächtnisses ([Galitz, 1988], Seite 17).

#### Vereinfachtes Modell

Für die Verdeutlichung der menschlichen Signalverarbeitung zeige ich ein vereinfachtes Modell, den *Model Human Processor* von [Card, Moran und Newell, 1983]. Dieses Modell zeigt die für diese Arbeit wesentlichen Aspekte, ohne unnötig auf Details einzugehen und ist beschränkt auf die von mir betrachteten Sinne Sehen und Hören.

Der sensorische Reiz gelangt zunächst als

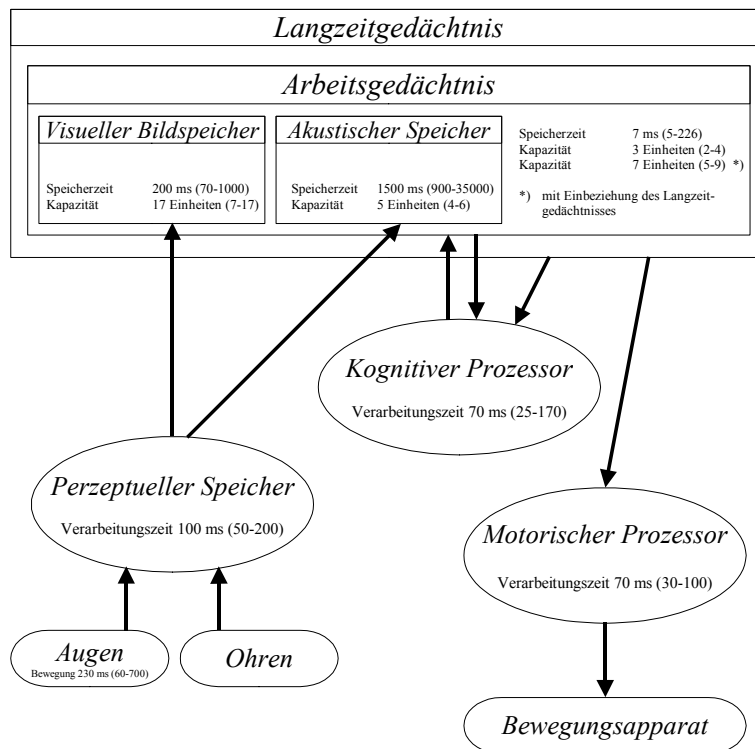


Abbildung 11: Vereinfachtes Modell der menschlichen Signalverarbeitung (nach [Card, Moran und Newell, 1983], Seite 26, Figure 2.1: *The Model Human Processor*)

nicht identifizierte physikalische Repräsentation in den entsprechenden perzeptuellen Speicher, also entweder in den visuellen Bildspeicher oder in den akustischen Speicher. Kurz darauf wird eine visuell bzw. akustisch kodierte Repräsentation von wenigstens einem Teil des Inhalts des perzeptuellen Speichers in den kognitiven Arbeitsspeicher aufgenommen. Ist der sensorische Reiz zu komplex, können nicht alle Reize als entsprechende Repräsentation in den kognitiven Arbeitsspeicher übernommen wurden. Der kognitive Prozessor entscheidet, welche Teile des perzeptuellen Speichers in den kognitiven Arbeitsspeicher aufgenommen werden. Perzeptueller Speicher und kognitiver Arbeitsspeicher sind flüchtige Speicher. Als Speicherzeit ist die Zeit angegeben, bis zu der die Wahrscheinlichkeit für das erfolgreiche Abrufen der gespeicherten Information mehr als 50% beträgt ([Card, Moran und Newell, 1983], Seiten 28 ff.).

Der kognitive Prozessor vergleicht den Inhalt des kognitiven Arbeitsspeicher mit dem Inhalt des Langzeitgedächtnisses. Dieser Vorgang heißt *Erkennen*. Die Verarbeitungszeit des kognitiven Prozessors ist nicht gleichbleibend. Sie ist kürzer, wenn der Mensch sich mehr anstrengen muss, sei es wegen der zugrunde liegenden Aufgabenstellung oder wegen einer extrem großen Informationsmenge. Auch Übung verringert die Verarbeitungszeit. Außerdem hängt die Verarbeitungszeit von der zu erkennenden Information ab:

Zu erkennendes Objekt	Benötigte Zeit je Einheit <sup>1</sup>
Ziffern	33 ms (27-39)
Farben	38 ms
Buchstaben	40 ms (24-65)
Wörter	47 ms (36-552)
Geometrische Formen	50 ms
Zufällige Formen	68 ms (42-93)
Sinnlose Silben	73 ms

Tabelle 1: *Benötigte Zeit zum Erkennen*

Das kognitive System kann zwar parallel mehrere Objekte erkennen, aber es kann nicht mehr als eine Tätigkeit gleichzeitig ausgeführt werden. Das scheinbar gleichzeitige Ausführen mehrerer Tätigkeiten ist in Wahrheit ein Time-Sharing-Verfahren, wie es auch bei Computern üblich ist ([Card, Moran und Newell, 1983], Seiten 41 ff.).

<sup>1</sup> Angegeben ist die typische Zeit, in Klammern Minimal- und Maximalwerte.

## Reaktionszeiten

Abhängig von dem Reiz sowie dem notwendigen Erkennen ist zu vermuten, dass die Reaktionszeit des Benutzers variieren wird. Dieses konnte experimentell bestätigt werden. Abhängig von dem Schlüsselreiz und der darauf notwendigen Reaktion wurden folgende Reaktionszeiten eines Benutzers am Computer bestimmt:

- Der Benutzer drückt die Leertaste, sobald irgendein beliebiges Symbol auf dem Bildschirm erscheint (einfache Reaktionszeit): 100-400 ms
- Der Benutzer erhält nacheinander zwei Symbole. Er drückt *Ja*, wenn die Symbole übereinstimmen, ansonsten *Nein*. Reaktionszeit: 130-640 ms
- Der Benutzer erhält nacheinander zwei Symbole. Er drückt *Ja*, wenn die Symbole den selben Namen haben (z.B. „a“ und „A“), ansonsten *Nein*. Reaktionszeit: 155-810 ms
- Der Benutzer erhält nacheinander zwei Symbole. Er drückt *Ja*, wenn die Symbole zur selben Klasse gehören (z.B. Buchstaben oder Zahlen), ansonsten *Nein*. Reaktionszeit: 180-980 ms

Die hier gelisteten Reaktionszeiten wurden für jeweils nur eine mögliche Reaktion gemessen. Muss der Benutzer eine Reaktion aus verschiedenen möglichen auswählen, müssen für jede weitere mögliche Reaktion etwa 70 ms addiert werden ([Card, Moran und Newell, 1983], Seiten 66 ff.).

## Lesegeschwindigkeit

Viele optische Nachrichten können nur dann vollständig erfasst werden, wenn die Augen bewegt werden. Insbesondere Texte sind so lang, dass sie nicht komplett im optischen „Nahbereich“<sup>1</sup> erfasst werden können. Die für die Augenbewegung notwendige Zeit dominiert dann die für die Verarbeitung benötigte Zeit. Mit einer Augenbewegung werden verschiedene Informationseinheiten aufgenommen: Einzelne Buchstaben, ganze Wörter oder Phrasen. Die Lesegeschwindigkeit beträgt 52 Wörtern je Minute beim Erfassen einzelner Buchstaben pro Augenbewegung, 261 Wörter pro Minute beim Erfassen kompletter Wörter und 652 Wörter je Minute beim Erfassen von Phrasen. Wie schnell ein Leser in einer konkreten Situation tatsächlich ist, hängt von seinen persönlichen Fähigkeiten sowie der Lesbarkeit des Textes ab, die wiederum von Typografie, Kontrast der Buchstaben zum Text, Umgebungshelligkeit und anderen Faktoren abhängt. Bei extrem schlecht lesbarem Text wird die Lesegeschwindigkeit nicht mehr von der für eine Augenbewegung notwendigen Zeit, son-

---

1 Siehe Seite 35 in dieser Arbeit

dem von der für die kognitive Verarbeitung notwendigen Zeit bestimmt ([Card, Moran und Newell, 1983], Seiten 50 f.).

Nicht berücksichtigt wurde von [Card, Moran und Newell, 1983] die inhaltliche Schwierigkeit des Textes. So werden selbst schnelle Leser bei unbekanntem Wörtern nicht mehr ganze Phrasen erfassen, sondern schlimmstenfalls nur noch einzelne Buchstaben – insbesondere bei Fremdwörtern oder unbekanntem Fachbegriffen. Auch durch inhaltliche Schwierigkeiten sinkt also die Lesegeschwindigkeit.

### **Tolerierte Antwortzeit**

Die kognitiven Fähigkeiten des Menschen bedingen bestimmte Mindestverarbeitungszeiten für bestimmte Aufgaben. Die Antwortzeiten eines Computers sollten sich an die jeweilige Verarbeitungsgeschwindigkeit seines Benutzers anpassen, damit dieser weder in seinem Arbeitsfluss gebremst wird, noch sich durch zu schnelle Reaktionszeiten gefordert fühlt und Stresssymptome entwickelt. In diesem Abschnitt betrachte ich nur die maximale Antwortzeit, die den Benutzer in seinem Arbeitsfluss noch nicht beeinträchtigt. Diese Zeit gibt einen Hinweis darauf, ab wann Interimsausgaben wie beispielsweise eine Fortschrittsanzeige notwendig werden. Die minimale Antwortzeit ist eher unter arbeitspsychologischen Gesichtspunkten interessant, für die Gestaltung von Statusanzeigen ist sie unwichtig.

Die vom Benutzer tolerierte Antwortzeit ist nicht konstant. Sie hängt stattdessen von der auszuführenden Aufgabe ab. [Shneiderman, 1998] (Seite 367) empfiehlt, dass einfache, häufig durchgeführte Ausgaben nach 1 s, gewöhnliche Aufgaben nach 2-4 s und komplexe Aufgaben nach 8-12 s eine Rückmeldung liefern sollen. Dauert es länger, soll der Benutzer darüber informiert werden. Auch [Galitz, 1988] (Seite 49) sieht die tolerierte Antwortzeit als abhängig von der zu bearbeitenden Aufgabe und setzt sie in Bezug zu der Geschwindigkeit und dem Fluss menschlicher Informationsverarbeitung:

- Ist kontinuierliches Denken erforderlich und muss Information während mehrerer Systemreaktionen behalten werden, sollte die Reaktionszeit unter 2 s liegen.
- Wenn der Mensch die Aufgabe oder eine Teilaufgabe als beendet ansehen kann, hohe Konzentration nicht erforderlich ist und das Kurzzeitgedächtnis nur leicht belastet ist, sind 2 s bis 4 s Reaktionszeit angemessen.
- Wenn die Aufgabe beendet ist und minimale Belastungen des Kurzzeitgedächtnisses existieren, darf die Reaktionszeit zwischen 4 s und 15 s liegen.

- Wenn der Benutzer zwischendurch andere Dinge machen kann und zurückkehren kann, wann er es möchte, kann die Reaktionszeit größer als 15 s sein.

Weiter empfiehlt [Galitz, 1988], den Bildschirm nie mehr als ein paar Momente leer zu lassen, da der Benutzer ansonsten denken könnte, das System hätte versagt. Generell sollte ein System alle Aktionen des Benutzers augenblicklich bestätigen, entweder durch sofortige Ausführung, einen geänderten Zustand oder Wert, eine Fehlermeldung oder eine In-Bearbeitung-Anzeige ([Galitz, 1988], Seite 38). Auch [Tognazzini, 1996] fordert, dass Ausgaben sofort, also nach weniger als einer halben Sekunde, gemacht werden sollten. Ist eine sofortige Ausgabe nicht möglich, soll der Benutzer bis zur Fertigstellung des Prozesses unterhalten werden, um die subjektive Wartezeit zu verkürzen (Seite 308). Automatisch nach einer Wartezeit stattfindende Aktionen sollten die restliche Wartezeit zeigen. Wenn sich also ein Dialog automatisch nach einer Wartezeit schließt, sollte er einen Countdown zeigen ([Tognazzini, 1992], Seite 217). Die Antwortzeit sollte für den Benutzer vorhersagbar sein, damit er seinen Arbeitsrhythmus darauf einstellen kann ([Blake, 1991], Seite 293).

### **Folgerungen für die Gestaltung von Statusinformationen**

Menschen erkennen verschiedene Dinge unterschiedlich schnell. Am schnellsten werden einzelne Ziffern wahrgenommen, sinnlose Silben am langsamsten. Diese Tatsache kann man bei der Gestaltung von Statusanzeigen ausnutzen, um das Erkennen eines Status zu vereinfachen. Abgesehen von Ziffern werden Farben am schnellsten erkannt. Eine erste Überlegung wäre also, Statusänderungen durch Farbänderungen darzustellen. Leider muss diese Möglichkeit mit Rücksicht auf Farbenblindheit verworfen werden. Buchstaben werden ebenfalls recht schnell erkannt. Jedoch würde ich mit Rücksicht auf die Ästhetik eine Statusanzeige nicht mit Buchstaben realisieren. Als nächste Positionen in der Liste stehen Wörter, geometrische Formen und zufällige Formen. Auch wenn die durchschnittliche Zeit für das Erkennen einzelner Wörter hier am günstigsten erscheint, so ist doch der große obere Grenzwert beachtenswert, der fast das 12-fache des Durchschnittswerts beträgt. Für mich folgt hieraus, dass von Wörtern als Statusanzeige vorsichtshalber Abstand gehalten werden sollte. Statusänderungen können also am besten durch Formänderung dargestellt werden. Eine Kombination mit Farbänderung macht einen geänderten Status, egal ob mit Ziffern, Buchstaben, Formen oder Wörtern angezeigt, für nicht farbenblinde Benutzer sogar noch schneller erkennbar.

Wenn Statusanzeigen Texte enthalten, ist die Lesegeschwindigkeit zu berücksichtigen. Sie sinkt, wenn nicht mehr Phrasen, sondern nur noch Buchstaben erfasst

werden. Dies ist immer bei unbekanntem Wörtern der Fall. In diesem Fall also sollten die verwendeten Wörter möglichst einfach sein, um das Lesen zu erleichtern.

Beachtet werden sollten auch die Unterschiede zwischen der Verarbeitung optischer und der Verarbeitung akustischer Nachrichten. Bedingt durch die für Augenbewegungen nötige Zeit dauert es unter Umständen länger, bis eine optische Nachricht erkannt wird. Ein weiterer Vorteil einer akustischen Nachricht ist ihre längere Speicherzeit. Der größte Nachteil einer akustischen Nachricht jedoch dürfte in der geringen Speicherkapazität für akustische Nachrichten liegen, die etwa ein Drittel der Kapazität für optische Nachrichten beträgt. Die Gefahr, dass ältere Nachrichten von neueren verdrängt werden, ist dadurch für akustische Nachrichten größer als für optische Nachrichten. Akustische Nachrichten könnten also zum Beispiel extrem kritische Statusänderungen anzeigen, wenn es auf schnelle Reaktion ankommt.

Vorhersagbare Antwortzeiten können auf zwei Wegen erreicht werden. Der offensichtliche, aber technisch nicht immer machbare Weg ist eine konstante Antwortzeit. Bei unregelmäßigen Antwortzeiten helfen ausführliche Statusanzeigen dem Benutzer beim Vorhersagen der konkreten Antwortzeit.

### **3.5 Mentale Modelle**

Mentale Modelle spielen eine wesentliche Rolle bei der Interaktion des Benutzers mit dem Computer:

- Für das Verständnis vom System
- Beim Erlernen des Systems
- Beim Finden von Problemlösungen während der Arbeit mit dem System

Es ist also wichtig, dass der Benutzer ein konzeptuelles oder mentales Modell entwickeln kann. Dabei hilft ihm unter anderem das notwendige und angemessene Feedback, egal, ob optischer oder akustischer Natur: Ein effektiver Aufbau des mentalen Modells findet nur statt, wenn Feedback bezüglich der Korrektheit der getroffenen Handlungen gegeben wird. Feedback erhöht auf diesem Weg die menschliche Leistungsfähigkeit ([Herzer und Noll, 1990], Seite 61; [Galitz, 1988], Seiten 22, 25).

Aufgrund der großen Bedeutung mentaler Modelle bei der Interaktion mit dem Computer gehe ich ausführlich auf sie ein. Zuerst werde ich klären, was ein mentales Modell ist und anschließend zeigen, wie es gebildet und verwendet wird.

## **Begriffsklärung**

Kognitionswissenschaftler gehen davon aus, dass menschliches Denken und Lernen mittels mentaler Modelle erfolgt ([Herzer und Noll, 1990], Seite 59). Mentale Modelle sind hypothetische Konstrukte zur Erklärung von Verhalten; in der Software-Ergonomie sollen sie menschliche Lern- und Arbeitsleistungen erklären. Ein mentales Modell ist kein gegenständliches Modell, sondern ein gedankliches. Es kann sowohl durch Anregung von außen (beispielsweise durch Lehrbücher) als auch vom jeweiligen Menschen selbst entwickelt werden ([Dutke, 1994], Seiten 1-2, 76).

Jedes Modell, also auch ein mentales, ist eine Abbildung eines Originals. Das bedeutet, dass Originalattributen Modellattribute zugeordnet sind. Es wird nur die Untermenge der Originalattribute im Modell abgebildet, die der Entwickler oder der Benutzer des Modells für relevant halten. Vollständige Modelle gibt es nur bei einfachen Sachverhalten, nicht aber bei komplexeren.

Die in einem mentalen Modell abgebildeten Merkmale hängen vom Vorwissen der Person sowie dem Zweck des mentalen Modells ab. Es können auch durch das Vorwissen bedingte zusätzliche, also nicht im Original enthaltene, Merkmale im Modell enthalten sein. Die Relationen zwischen Elementen eines Modells gleichen stets (zumindest teilweise) den Relationen des Originals. Modelle ersetzen also Originale für bestimmte Benutzer des Modells, innerhalb bestimmter Zeitintervalle und zu bestimmten Zwecken. Unterschiedliche Sichtweisen auf das Original führen daher zu unterschiedlichen Modellen ([Dutke, 1994], Seiten 4-6, 15, 76; [Herzer und Noll, 1990], Seite 59).

Menschen entwickeln mentale Modelle, um bestimmte Sachverhalte besser verstehen oder behalten zu können. Sie nutzen sie aber auch, um ihre Handlungen zu planen und zu steuern. Einige mentale Modelle sind daher eher verstehensorientiert, andere eher handlungsorientiert ([Dutke, 1994], Seite 2).

Der Benutzer eines Modells gewinnt Erkenntnisse, die er aus dem Original nicht oder nur schwer gewonnen hätte. Ein Modell kann außerdem anderen Menschen Dinge vermitteln. Mentale Modelle sind zeitlich instabil. Besonders bei unregelmäßiger Benutzung werden Details vergessen. Sie können aber als nützliche Heuristik bei Verständnisproblemen oder bei Routinehandlungen angesehen werden ([Dutke, 1994], Seiten 5, 13, 77-78).

Wie ein mentales Modell konkret aussieht, ist nicht geklärt. [Herzer und Noll, 1990] zeigen verschiedene Ansätze aus der Literatur, meinen aber, dass für die Betrachtung von Feedback lediglich interessant ist, dass mentale Modelle die Basis für das Handeln des Benutzers bilden (Seite 60). Meiner Ansicht nach haben sie hier jedoch Un-



recht, da die Annahme über das konkrete Aussehen des mentalen Modells impliziert, wie der Benutzer bei der Bildung unterstützt werden kann. [Dutke, 1994] beispielsweise glaubt, dass mentale Modelle bildhaft-anschauliche Vorstellungen sind oder zumindest erzeugen (Seiten 49, 109) und konzentriert sich auf Visualisierungen zum Erzeugen mentaler Modelle:

*Die Bildung eines mentalen Modells kann durch bildliche Mittel im weiteren Sinne unterstützt werden. ([Dutke, 1994], Seite 110)*

Ich halte angesichts der Tatsache, dass wesentliche Teile der Interaktion zwischen Mensch und Computer optisch stattfinden, die Auffassung von Dutke zur Betrachtung von Feedback und damit auch von Statusanzeigen für praktikabler.

### **Metaphern und konzeptuelle Modelle**

Mentale Modelle werden häufig mit Hilfe von Analogien entwickelt. Durch die Analogiebeziehung kann bereits bekanntes Wissen aus anderen Bereichen auf einen neuen Bereich übertragen werden. Das Konstruieren oder Entdecken einer Analogiebeziehung und damit die Entwicklung eines mentalen Modells kann durch Metaphern und konzeptuelle Modelle gefördert werden ([Dutke, 1994], Seite 82).

Eine Metapher ist ein didaktisches Mittel, auf eine Analogiebeziehung hinzuweisen. Metaphern greifen auf Wissen bekannter anderer Bereiche zurück. Ein konzeptuelles Modell ist ebenfalls ein Mittel, auf eine Analogiebeziehung hinzuweisen. Im Gegensatz zu Metaphern stellen konzeptuelle Modelle die Zusammenhänge jedoch in den Begriffen des Zielbereichs dar. Man kann also sagen, dass Metaphern gefunden, konzeptuelle Modelle hingegen konstruiert werden ([Dutke, 1994], Seite 19).

Metaphern sind brauchbar, wenn die Ähnlichkeit zwischen dem Basisbereich der Metapher und dem Zielbereich mittelgroß ist. Ist die Ähnlichkeit zu gering, wird die Metapher zwar erkannt, aber die Analogiebeziehung zum Zielbereich kann nicht oder nur schwer hergestellt werden. Ist eine Metapher dem Zielbereich zu ähnlich, wird sie wörtlich interpretiert und nicht als Metapher erkannt. Die Nützlichkeit sinkt, wenn Merkmale des Basisbereichs übertragen werden, die zu ungünstigen Relationen im Zielbereich führen. Der Benutzer muss also wissen, welche Merkmale übertragen werden können. Kann dieses Wissen nicht vorausgesetzt werden, sollte die Verwendung der Metapher kritisch betrachtet werden. Ähnlich gilt für ein konzeptuelles Modell, dass es die funktionalen Zusammenhänge zwischen den Systemkomponenten enthalten sollte ([Dutke, 1994], Seiten 20, 92, 99-100). Die Qualität einer auf dem Bildschirm visualisierten Metapher hängt also nicht davon ab, wie gut die ein-

zelenen Icons die realen Objekte abbilden, sondern davon, wie gut die funktionalen Beziehungen zu erkennen sind:

*Kann man dem Drucker-Icon beispielsweise ansehen, daß der Drucker gerade nicht angesteuert werden kann, daß eine Reihe von Dokumenten sich in der Warteschlange befindet, oder aus welchen Systemkomponenten heraus der Drucker überhaupt angesprochen werden kann, erhöht dies die Transparenz des Systems beträchtlich. ([Dutke, 1994], Seite 141)*

Die Verwendung von Metaphern hilft Computerlaien beim Erlernen der Bedienung eines Computers, da bereits bekanntes Wissen aus anderen Bereichen genutzt werden kann. Metaphern sollten so beschaffen sein, dass sie eine positive Einstellung zum Computer bewirken. Insbesondere, wenn an Bekanntes angeknüpft wird, kann Unsicherheit beim Benutzer reduziert werden. Auch die Emotionen des Benutzers gegenüber dem Basisbereich der Metapher können in den Zielbereich übertragen werden. Ein „harmloser“ Basisbereich kann also dazu führen, dass auch der Zielbereich als nicht bedrohlich erscheint. Besonderes Augenmerk sollte darauf gelegt werden, keine negativen Emotionen zu übertragen, beispielsweise durch Metaphern militärischen Ursprungs ([Dutke, 1994], Seiten 82-83, 101-102). Weitere Probleme können bei der Internationalisierung auftreten – nicht jede Metapher wird in allen Ländern verstanden.

### **Entwicklung mentaler Modelle**

Da mentale Modelle häufig aufgrund von Analogien gebildet werden, ist ein brauchbarer erster Schritt für die Entwicklung eines mentalen Modells das Finden einer Metapher oder das Konstruieren eines konzeptuellen Modells. Unterschieden wird zwischen inneren Modellen, die eine Person sich selber bildet, und externen Modellen, die andere für sie bilden, z.B. durch eine geschriebene Dokumentation ([Dutke, 1994], Seite 7).

Mentale Modelle sind nicht statisch, sondern sie entwickeln sich bei der Beschäftigung mit der Aufgabe ständig weiter, indem versucht wird, neue Fakten entweder mit dem bisherigen Modell in Einklang zu bringen oder ein neues Modell zu konstruieren – gegebenenfalls auch durch Kombination mit einem anderen Modell. Ist ein neuer Fakt vollständig in ein mentales Modell integriert, erfolgt eine Konsistenzprüfung. Diese Prüfung der internen Konsistenz stellt jedoch nicht die objektive Richtigkeit des mentalen Modells sicher: Oftmals jedoch wird das mentale Modell auch dann nicht geändert, wenn neue Fakten eindeutig im Widerspruch zu ihm stehen. Vor einer Änderung des mentalen Modells scheinen Lernaufwand und Nützlichkeit gegeneinander abgewogen zu werden: Hält eine Person ihr mentales Modell für nützlich, ist

sie nur schwer dazu zu bringen, es zu ändern, selbst wenn es objektiv betrachtet falsch ist; dem Aufwand zur Korrektur des Modells steht subjektiv oft kein (angemessener) Nutzen gegenüber. Allerdings können auch objektiv falsche Modelle unter bestimmten Einschränkungen nützlich sein – das mentale Modelle ist zwar technisch nicht exakt, aber es ist funktional ([Dutke, 1994], Seiten 15, 45-46, 76; [Herzer und Noll, 1990], Seite 59).

### **Arbeiten mit mentalen Modellen**

Jeder Benutzer hat bereits seine Vorstellungen und damit ein rudimentäres mentales Modell von einem System, bevor er mit diesem System überhaupt zum ersten Mal arbeitet. Grundlagen hierfür sind technische Vorkenntnisse und das Aufgabenumfeld, aber auch Vorurteile spielen eine Rolle. Dieses Modell kann durch Anbieten einer Metapher beeinflusst werden. Dazu eignen sich beispielsweise Schulungen, aber auch die Gestaltung der Oberfläche. Jedoch sollte der Benutzer auch ohne Metapher bei der Bildung eines mentalen Modells unterstützt werden.

Besonders Personen, die ein System noch nicht sehr lange oder nur gelegentlich nutzen, sind leicht beeinflussbar. Die angebotene Metapher hilft diesen Benutzern beim Erlernen des Systems: Funktionen, die der Metapher entsprechen, werden schneller erlernt, abweichende Funktionen dafür aber langsamer. Die Verwendung einer Metapher hilft nur in der Anfangsphase des Lernens. Im Verlauf der Arbeit mit dem System baut der Benutzer dann ein mentales Modell auf, das über die Metapher hinausreichendes Wissen enthält. Auf dieses mentale Modell greift er zukünftig während der Arbeit mit dem System zu. [Dutke, 1994] beobachtete diesen Transfer besonders bei Routinehandlungen (Seiten 56-57, 85, 102).

Mentale Modelle sind wichtig beim Erlernen eines Systems. Benutzer können die Arbeit mit einem neuen System auf zwei Arten erlernen:

- Extern gesteuertes Lernen: Die Lerninhalte, die Reihenfolge und die Art der Wissensaufnahme werden nicht vom Lernenden, sondern von anderen festgelegt.
- Intern gesteuertes Lernen: Der Lernende selbst legt die Inhalte, die Reihenfolge und die Art der Wissensaufnahme fest.

Computernutzer bevorzugen intern gesteuertes Lernen. Dabei verwenden sie ihr mentales Modell zum Simulieren des Systems: Aufgrund einer Eingabe wird im mentalen Modell durchgespielt, wie das System nach der Eingabe aussieht. Wichtig ist, dass der Benutzer beim explorierenden Lernen weder Orientierung, noch Kontrolle verliert, also nicht in Systemzustände gerät, die mit seinem mentalen Modell nicht zu erklären sind, da Unsicherheiten über Systemzusammenhänge zu irrationalen

Überzeugungen führen können. Bedienfehler jedoch sind ein Bestandteil explorierenden Lernens. Sie zeigen einerseits dem Benutzer die Grenzen seines Wissens sowie Unstimmigkeiten in seinem mentalen Modell vom System auf, andererseits zeigen nicht mehr auftretende Fehler einen Lernfortschritt ([Dutke, 1994], Seiten 13, 77, 154, 158).

Ein weiterer Vorteil eines mentalen Modells liegt darin, dass die Benutzer Problemlösungen eigenständig entwickeln können. Sie müssen auch sich weniger anstrengen, Fakten zu erinnern, sondern können falsche oder fehlende Erinnerungen mit Hilfe des mentalen Modells kompensieren ([Dutke, 1994], Seite 91).

Ein mentales Modell kann für den Benutzer die Komplexität eines Vorgangs reduzieren. Der Benutzer kann sich dank der reduzierten Komplexität auf seine Aufgabe konzentrieren und muss sich nicht zusätzlich noch um die Eigenarten des benutzten Werkzeugs kümmern. Die Darstellung muss wichtige Eigenschaften des Objekts, die der Benutzer aufgrund seines mentalen Modells erwartet, zeigen. Jedoch verändert auch umgekehrt die Darstellung das mentale Modell des Benutzers. Werden beispielsweise Objekte in einer bestimmten Sortierung ausgegeben, wird das mentale Modell des Benutzers die Objekte als nach dem Sortierkriterium geordnete Liste enthalten, unabhängig von der tatsächlichen physischen Speicherung ([Herzer und Noll, 1990], Seiten 62-63).

### **Mentale Modelle in der Software-Entwicklung**

*Die Gestaltung von Benutzungsoberflächen ist in erster Linie ein Kommunikationsproblem. Die Konstrukteure wollen den zukünftigen Benutzern mitteilen, wie und wozu ein Softwareprodukt benutzt werden kann. ([Dutke, 1994], Seite 124)*

Nicht nur der Benutzer, sondern auch der Entwickler hat ein mentales Modell des Systems. Während jedoch das mentale Modell des Benutzers aufgabenbezogen ist, hat der Systementwickler ein eher technisch orientiertes Modell vom System, auf dessen Basis er das System entwickelt. Die Konsistenz des Systems zum mentalen Modell des Benutzers muss dann durch Feedback erreicht werden. Da das System jedoch nach dem technikorientierten mentalen Modell des Systementwicklers konstruiert wird, das vom aufgabenbezogenen mentalen Modell des Benutzers abweicht, und sich das mentale Modell des Benutzers zudem während der Arbeit verändert, kann diese Konsistenz nie vollständig gewährleistet werden. Sie wird größer, wenn der Benutzer bereits bei der Systementwicklung herbeigezogen wird und sein mentales Modell mit in die Systementwicklung einfließt ([Dutke, 1994], Seite 7; [Herzer und Noll, 1990], Seiten 60-61).

Das mentale Modell eines Benutzers sollte am besten durch die Wahl einer geeigneten Metapher berücksichtigt werden. Ob sich aber der Aufwand für die Entwicklung einer Metapher lohnt, hängt von verschiedenen Faktoren ab. Arbeiten nur wenige Benutzer sehr intensiv mit einem System, ist der Aufwand für die Entwicklung einer Metapher unangemessen hoch: Die wenigen Benutzer sind relativ schnell in einem Stadium, wo sie als Experten die Metapher nicht mehr benötigen. Arbeiten jedoch viele Benutzer nur gelegentlich mit dem System, erreichen diese Benutzer wahrscheinlich nie das Expertenstadium, in dem sie die Metapher nicht mehr nutzen. Damit ist der Nutzen einer Metapher für viele Benutzer und über einen längeren Zeitraum gegeben, so dass der Aufwand für ihre Entwicklung angemessen ist.

Bei sehr komplexen Systemen werden eventuell keine vollständig passenden Metaphern aus der realen Welt gefunden. Hier muss abgewogen werden, ob Ungenauigkeiten und eventuelle Inkonsistenzen der Metapher ihren Nutzen übersteigen. Eventuell ist es in solchen Fällen besser, ganz auf Metaphern zu verzichten. Ein anderer Weg ist, die Entwicklung eines Systems an möglichen Metaphern zu orientieren. Die Gefahr hierbei ist jedoch, die Funktionalität des Systems künstlich auf die Möglichkeiten der realen Welt zu beschränken. Auch bei einer Erweiterung der Funktionalität eines Systems muss abgewogen werden, ob bisher genutzte Metaphern erhalten bleiben können. Die Aufgabenangemessenheit ist hier wichtiger als die leichte Erlernbarkeit. Eine Alternative zur Verwendung einer Metapher ist die Konstruktion eines konzeptuellen Modells, auch wenn der Benutzer hier weniger auf bekanntes Wissen aus anderen Bereichen zurückgreifen kann. Einzelne Komponenten eines konzeptuellen Modells können weiterhin metaphorisch sein ([Dutke, 1994], Seiten 103-106).

### **Folgerungen für die Gestaltung von Statusinformationen**

*Eine gute Software ist daran orientiert, daß Menschen in ihrer Umwelt notorisch Sinn, Ordnung und Zusammenhänge sehen. Andeutungen genügen, um Deutungen und Handlungsweisen im Sinne der aktivierten mentalen Modelle auszulösen. ([Glaser, 1994], Seite 43)*

Statusanzeigen können helfen, dem Benutzer diese notwendigen Andeutungen zu geben. Bei der Gestaltung von Statusanzeigen muss darauf geachtet werden, dass sie einerseits einem bereits vorhandenen mentalen Modell des Benutzers nicht widersprechen, andererseits das mentale Modell des Benutzers in die vom Systementwickler gewünschte Richtung beeinflussen. Hilfreich für das Entwickeln eines mentalen Modells sind Metaphern. Diese Tatsache kann auch für Statusanzeigen ausgenutzt werden. Eine metaphorische Statusanzeige ist geeigneter als eine abstrakte.

Technisch betrachtet gibt es zwei Arten von Mappings zwischen der Wirklichkeit im Computer und der Anzeige auf dem Bildschirm. Das Erste ist ein konzeptuelles Mapping zwischen den Ereignissen im Computer und dem in einem Modell (z.B. einem Desktop). Das Zweite ist ein Mapping zwischen dem Modell und der Anzeige, sei es auf dem Bildschirm oder akustisch. Es ergibt sich also folgende Kette:

Computer → Modell → Anzeige ([Gaver, 1989], Seite 86).

Diese Kette bedarf allerdings noch einer Ergänzung um den Benutzer und sein mentales Modell:

Computer → Modell → Anzeige → Modell → Benutzer

Statusanzeigen sind ein Teil der Anzeige. Bei ihrer Gestaltung ist also darauf zu achten, dass sie das extern (beispielsweise durch den Systementwickler) vorgegebene Modell der Vorgänge im Computer so darstellen, dass es in das mentale Modell des Benutzers integriert werden kann.

### 3.6 Menschliches Handeln

Statusanzeigen sind Teil des Feedbacks. Menschliches Handeln wird als ein rückgekoppelter Kreislauf verstanden, in dem Feedback und damit auch die Statusanzeigen eine wichtige Rolle spielen. Unmittelbar aus dem Modell dieses Kreislaufs ergeben sich Erklärungen, warum Menschen Fehler machen. Statusanzeigen können dem Benutzer helfen, Fehler zu vermeiden.

#### Der menschliche Handlungszyklus

Menschliche Tätigkeit ist bewusst und motiviert und führt damit zu zielgerichtetem Handeln. Handlung und Tätigkeit sind zwei Aspekte des selben Sachverhalts. Eine Handlung ist das, was der Mensch konkret ausführt, ohne Berücksichtigung seiner Motive. Eine Tätigkeit ist das, was ein Mensch unter Berücksichtigung seines Motivs tut. Aktionen, die die selbe Handlung haben, können also vollkommen unterschiedliche Tätigkeiten sein ([Herzer und Noll, 1990], Seiten 67-68).

Menschliches Handeln wird als geregelter Kreis gesehen. [Brodbeck und Rupiotta, 1994] zeigen ein einfaches Modell:

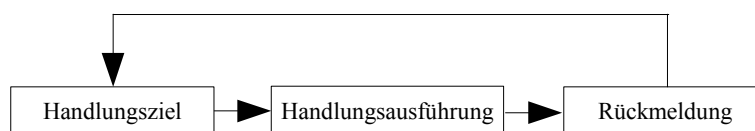


Abbildung 12: Der menschliche Handlungsprozess  
(nach [Brodbeck und Rupiotta, 1994], Seite 199, Abb. 5-1)

Der Benutzer bestimmt sein Handlungsziel. Nach der Handlungsausführung erhält er eine Rückmeldung, die in die Bestimmung des nächsten Handlungsziels eingeht.

[Norman, 1989] ergänzt dieses Modell zunächst um die Welt, also um die Objekte, die manipuliert werden (Abbildung 14). Die für diese Arbeit wichtige Phase der Auswertung wird von ihm in drei Unterpunkte eingeteilt (Abbildung 13).

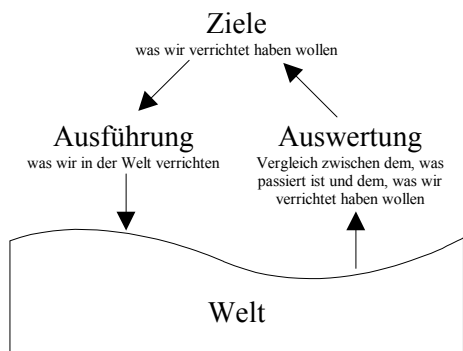


Abbildung 14: Der Zyklus des Handelns (nach [Norman, 1989], Seite 62, Abbildung 2.2)

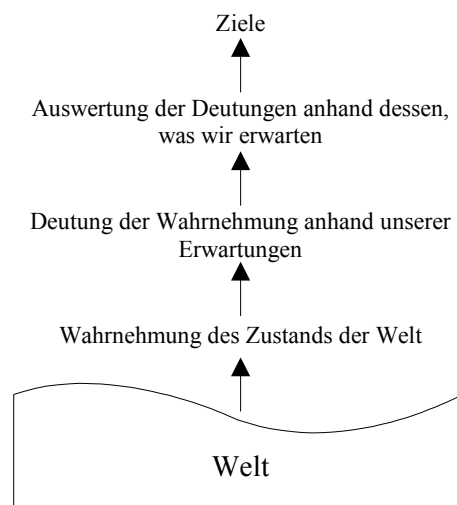


Abbildung 13: Stadien der Auswertung (nach [Norman, 1989], Seite 64, Abbildung 2.4)

Insgesamt unterteilt [Norman, 1989] menschliches Handeln in sieben Stadien:

1. Zunächst muss das Ziel der Handlung formuliert werden.
2. Anschließend wird die Intention formuliert. Die Intention spezifiziert das, was gemacht werden soll, während das Ziel festlegt, was erreicht werden soll.
3. Im Anschluss muss die Handlung spezifiziert und
4. danach ausgeführt werden.
5. Nach der Ausführung wird der Zustand der Welt wahrgenommen und
6. interpretiert.
7. Das Ergebnis dieser Interpretation wird im Hinblick auf das zu erreichende Ziel ausgewertet.

[Norman, 1989] weist darauf hin, dass diese sieben Stadien nur ein ungefähres Modell sind und sich mit an Sicherheit grenzender Wahrscheinlichkeit nicht eindeutig trennen lassen. Die meisten Ziele können nicht durch nur eine Handlung erreicht werden. Der Zyklus wird also mehrfach durchlaufen, wobei Ergebnisse einer Handlung genutzt werden können, um andere Handlungen zu steuern ([Norman, 1989], Seiten 64-65).

Aus dem siebenstufigen Modell lassen sich zwei Problembereiche ableiten: Die von [Norman, 1989] so genannte Kluft der Ausführung und die Kluft der Auswer-

tung. Die Kluft der Ausführung bezeichnet den Unterschied zwischen den Intentionen des Benutzers und den vom System zugelassenen Handlungen. Sie ist im Rahmen dieser Arbeit unwichtig. Die Kluft der Auswertung bezeichnet die Anstrengung des Benutzers, um den Zustand des Systems zu interpretieren und damit eine Rückmeldung über seine Handlung zu erhalten. Sie ist klein, wenn das System die notwendigen Informationen in einer leicht aufzunehmenden und leicht zu interpretierenden Form liefert, die mit den Vorstellungen des Benutzers vom System übereinstimmt ([Norman, 1989], Seiten 66-69).

### **Kontrolliertes und automatisches Handeln**

Die Steuerung einer Handlung geschieht teilweise bewusst, teilweise unbewusst. Die Handlungsvorbereitung findet auf verschiedenen Ebenen statt:

- *Intellektuelle Regulationsebene.* Hier finden intellektuelle Planungsvorgänge statt, also Analysen, Problemlösungen, Pläne.
- *Zwischenebene.* Hier gibt es flexible Handlungsmuster, die einerseits eine bestimmte Regelmäßigkeit aufweisen, andererseits aber flexibler anzuwenden sind. Diese Aktionen können bewusst gemacht werden, müssen aber nicht bewusst sein.
- *Sensumotorische Regulationsebene.* Hier finden kognitive Handlungsvorbereitungen statt. Diese sind nicht bewusst.

Nicht jede Aktion kann eindeutig einer bestimmten Ebene zugeordnet werden. Je häufiger eine Aktion ausgeführt wird, desto mehr wird sie auf einer niedrigeren Ebene reguliert. Jede Ebene benötigt unterschiedliche Unterstützung durch Rückmeldungen (nach [Frese, 1978], [Hacker, 1986] und [Volpert, 1979], beschrieben in [Herzer und Noll, 1990], Seite 69).

Menschen können Handlungen sowohl kontrolliert als auch automatisch ausführen. Die Grenze ist jedoch fließend. Viele, insbesondere elementare kognitive, Tätigkeiten können durch häufige Wiederholung automatisiert werden ([Glaser, 1994], Seite 45).

Automatische kognitive Prozesse werden zum Beispiel bei Rennwagen ausgenutzt. Rennfahrer wissen, dass „Nadel nach oben“ den Normalzustand anzeigt. Alle Abweichungen von der Senkrechten bezeichnen einen schlechten Zustand ([Cooper, 1995], Seite 129). Kritische Zustände sind daher unbewusst, also automatisch, zu erkennen.



Kontrollierte und automatische Handlungen unterscheiden sich in wesentlichen Punkten:

<b>Kontrollierte Handlung</b>	<b>Automatische Handlung</b>
Willkürlich gesteuert, bewusst	Keine willkürliche Steuerung, unbewusst
Aktiv zugewandte Aufmerksamkeit nötig	Keine aktive Aufmerksamkeit nötig
Nur eine Handlung ohne Interferenz möglich, Handlungen werden sequenziell ausgeführt	Mehrere Handlungen gleichzeitig ohne Interferenz möglich, Handlungen werden parallel ausgeführt
Im Kurzzeitgedächtnis ausgeführt, in der Kapazität begrenzt	Im Langzeitgedächtnis ausgeführt, in der Kapazität nicht begrenzt
Leicht programmiert, leicht geändert	Von ausgedehnter Übung abhängig. Schwer zu ändern, zu unterdrücken, oder zu ignorieren

*Tabelle 2: Kontrollierte und automatische Handlungen (nach [Glaser, 1994], Seite 47, Tab. 1-3)*

## **Fehler**

Der Benutzer legt zunächst ein Ziel fest und versucht dann, dieses mit einer oder mehreren Handlungen zu erreichen. Dieses gelingt ihm jedoch nicht immer. Häufig gibt es Abweichungen zwischen dem erwarteten Handlungsziel und dem tatsächlichen Ergebnis der Handlung. Diese Abweichungen werden von [Norman, 1989] als *Fehler* bezeichnet.

Fehler entstehen durch mangelhafte Information bei der Steuerung der Handlung. Mangelhafte Information ist entweder das Fehlen von notwendigen Informationen oder Mängel bei der Nutzung vorhandener Informationen. Mangelhafte Nutzung von Information beinhaltet fehlende Nutzung (übersehen oder vergessen, z.B. durch Zeitdruck oder durch Gewöhnung) und falsche Nutzung (z.B. durch getäuschte Erinnerung, falsche Entscheidungen). Fehler können einerseits vom Benutzer erkannt werden, andererseits können mögliche Fehler bereits vom Systementwickler vorweggenommen werden. Dazu muss der Entwickler mögliche Handlungsziele des Benutzers festlegen, so dass eine Abweichung von diesen festgelegten Zielen erkannt werden kann. Da dem Entwickler nicht alle tatsächlichen Handlungsziele der Benutzer bekannt sind, muss der Benutzer meist selbst entscheiden, ob ein Fehler vorliegt oder nicht. Diese Entscheidung kann er nur mit ausreichendem Feedback treffen ([Herzer und Noll, 1990], Seite 80).

Fehler können auf allen Ebenen der Handlungssteuerung auftreten:

<b>Regulationsgrundlage</b>	<b>Wissensfelder</b>		
Regulationsebenen	<b>Schritte im Handlungsprozess</b>		
	Ziele/Planung	Gedächtnis/ Monitoring	Rückmeldung
Intellektuelle Regulati- onsebene	Denkfehler	Merk- und Vergessensfehler	Urteilsfehler
Zwischenebene der fle- xiblen Handlungsmuster	Gewohnheits- fehler	Unterlassens- fehler	Erkennensfehler
Sensumotorische Regu- lationsebene	Bewegungsfehler		

Tabelle 3: *Regulationsgrundlage* (nach [Brodbeck und Rupiotta, 1994], Seite 202, Tab. 5-1)

Falls unbewusst durchgeführte Aktionen einen Fehler verursachen, können sie zur Fehlerbehebung wieder bewusst gemacht werden und dann verändert werden, um das Ziel der Handlung zu erreichen. Das Bewusstmachen unbewusster Aktionen heißt *Konzeptualisierung* ([Herzer und Noll, 1990], Seiten 70-71).

Auch [Norman, 1989] unterscheidet Fehler aufgrund der Ebene, auf der sie entstehen. Fehler bei vom Unterbewusstsein gesteuerten Aktionen nennt er *Fehlleistungen*, Fehler, die aus bewusster Überlegung herrühren, nennt er *Irrtümer*. Irrtümer gehen aus der Wahl falscher Ziele hervor. Eine Fehlleistung liegt vor, wenn ein angemessenes Ziel formuliert wird, aber die Ausführung fehlschlägt. Irrtümer sind schwerer zu entdecken als Fehlleistungen, da die ausgeführten Handlungen korrekt im Hinblick auf das falsche Ziel sind. Computerprogramme beispielsweise verlangen vor kritischen Operationen eine Bestätigung. Doch der Benutzer ist in diesem Augenblick mit der begonnenen Handlung zufrieden und bemerkt noch gar nicht, dass er z.B. die falsche Datei löschen wird. Die Gefahr einer Fehlleistung steigt, je besser die auszuführende Tätigkeit beherrscht wird: Je besser eine Tätigkeit beherrscht wird, desto weniger Aufmerksamkeit muss ihr gewidmet werden, desto automatischer und unbewusster läuft sie ab. Einander ähnliche Handlungen oder auch externe Ereignisse können dann schnell einen Fehler verursachen ([Norman, 1989], Seiten 127-129, 136).

[Norman, 1989] unterscheidet sechs verschiedene Arten von Fehlleistungen:

- *Fangfehler*: Eine oft ausgeführte Handlung übernimmt die Kontrolle über die gerade ausgeführte Handlung (Beim Zählen von Papier wird 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, Bube, Dame, König gezählt, weil die Person viele Kartenspiele gespielt hat).

- *Beschreibungsfehler*: Die ausgeführte und die beabsichtigte Handlung sind sehr ähnlich. Wenn die Handlung unvollständig beschrieben ist, kann die falsche Handlung ausgeführt werden („Wirf das Hemd in das Ding mit der runden Öffnung!“ und das Hemd fliegt in die Toilette, nicht in den Wäschekorb). Große Mengen identisch aussehender Bedienelemente sind prädestiniert für Beschreibungsfehler.
- *Datengesteuerte Fehler*: Externe Information kann eine automatische Handlung beeinflussen. Beispiel: Eine Person will eine andere anrufen, um ihr eine Nummer mitzuteilen. Statt die Telefonnummer zu wählen, wählt die Person jedoch die mitzuteilende Nummer, die auf einem Zettel vor ihr liegt.
- *Fehler durch assoziative Aktivierung*: Interne Gedanken und Assoziationen führen zu falschem Verhalten („Herein!“ beim Abheben des Telefons).
- *Fehler durch Aktivierungsverlust*: Die Person vergisst, was sie eigentlich tun wollte.
- *Modus-Fehler*: Der Benutzer eines Geräts wähnt das Gerät in einem anderen Modus und führt Aktionen aus, die in dem tatsächlichen Modus zu ungewollten Ergebnissen führen ([Norman, 1989], Seiten 129-133).

Jeden Fehler, den der Benutzer macht, macht er aus einem bestimmten Grund. Bei Irrtümern ist der Grund sehr wahrscheinlich unvollständige oder irreführende Information. Durch das Sichtbarmachen von Information kann der Designer den Benutzer vor Fehlern zu bewahren ([Norman, 1989], Seiten 135, 157-158, 167).

### **Folgerungen für die Gestaltung von Statusinformationen**

Im Rahmen des siebenstufigen Handlungszyklus werden Statusanzeigen für die ersten zwei Stadien der Auswertung benötigt, nämlich für die Wahrnehmung eines Zustands und für die Deutung dieser Wahrnehmung. Damit der Zustand der Welt wahrgenommen werden kann, müssen natürlich überhaupt erst einmal Statusanzeigen in ausreichender Menge vorhanden sein. Sie müssen außerdem auf die physiologischen Eigenarten des Menschen Rücksicht nehmen, um ihm die Wahrnehmung zu erleichtern. Die Deutung des Wahrgenommenen fällt leichter, wenn Statusanzeigen kongruent zum mentalen Modell des Benutzers sind. Damit helfen Statusanzeigen, die von [Norman, 1989] so genannte Kluft der Auswertung zu verkleinern.

Statusanzeigen können helfen, Fehler zu vermeiden. Zwei Arten von Fehlern gibt es: Irrtümer, die aufgrund eines falsch gewählten Ziels entstehen, und Fehlleistungen, die durch falsche Handlungen bei korrekt gewähltem Ziel entstehen.

Irrtümer können durch umfassende Information vermieden werden, bzw. aufgetretene Irrtümer können erst durch umfassende Information erkannt und korrigiert werden. Statusanzeigen müssen also so umfassend wie möglich informieren.

Um Fehlleistungen zu vermeiden, sollten Statusanzeigen

- möglichst verschieden voneinander sein, um Beschreibungsfehler zu vermeiden,
- einen Überblick über vergangene Aktionen des Benutzers liefern, um Fehler durch Aktivierungsverlust zu vermeiden und
- jeden Modus des Programms anzeigen, um Modusfehler zu vermeiden.

Fangfehler, datengesteuerte Fehler und Fehler durch assoziative Aktivierung können durch Statusanzeigen nicht vermieden werden.

Automatische Handlungen erlauben es, mehrere Aktionen gleichzeitig auszuführen, während nur eine einzige Aktion bewusst ausgeführt werden kann. Die Wahrnehmung und möglichst auch die Deutung einer Statusanzeige sollte daher nicht die aktive Aufmerksamkeit des Benutzers erfordern. Dieses würde ihn zum einen von seiner eigentlichen Aufgabe abhalten, zum anderen erlaubt das unbewusste Beobachten von Statusanzeigen auch das gleichzeitige Beobachten mehrerer Anzeigen.

## 4 Informationsdarstellung

Die Formen der Kommunikation zwischen Mensch und Computer sind im Wesentlichen seit Jahren unverändert. Auch wenn [Mountford, 1991] erwartete, dass Computer bald auf Sprache und Gesten reagieren, der Systemstatus mit Hilfe von Farben und Animation angezeigt und auf einen Blick erfasst werden könne, so dass der Umgang mit dem Computer und die Erwartungen an ihn sich gravierend ändern würden (Seite 248), so hat sich diese Erwartung in den letzten zehn Jahren nicht erfüllt. Zwar sind Zeigegeräte wie Mäuse allgemein üblich, aber dennoch werden mit ihnen weiterhin nur Objekte selektiert und Kommandos ausgewählt. Echte Gesten sind zwar auf Palmsize-Computern zu finden, diese haben aber noch keine sehr große Verbreitung gefunden. Spracheingaben sind noch seltener zu finden, im Büroalltag spielen sie keine Rolle. Und der Benutzer sitzt heute noch ahnungslos vor dem Computer, weil die notwendigen Statusinformationen fehlen.

Optische Ausgaben dominieren weiterhin die Kommunikation des Computers mit dem Benutzer und werden auch in Zukunft eine wesentliche Rolle spielen. Und auch wenn akustische Ausgaben bis heute noch nicht den Durchbruch geschafft haben, so sind sie dennoch ein vielversprechendes Medium. Daher beginne ich diesen Abschnitt mit einer Betrachtung, was für optische und akustische Ausgaben gleichermaßen zu beachten ist, und gehe anschließend auf die jeweiligen Besonderheiten ein. Statusanzeigen unterliegen jedoch auch anderen Einflüssen wie beispielsweise dem Programm, in dem sie verwendet werden. Daher betrachte ich anschließend die wichtigsten äußeren Einflüsse auf Statusanzeigen.

### 4.1 Allgemeine Überlegungen

[Herzer und Noll, 1990] verlangen, dass der Benutzer eine Rückmeldung über die Interpretation seiner Eingabe durch den Computer erhalten muss. Wird keine Rückmeldung gegeben, sollte die Interpretation durch den Computer aufgrund der ausgeführten Aktionen klar werden. Eine Rückmeldung sollte immer gegeben werden, wenn die Ausführung voraussichtlich länger dauert, kritische Aktionen wie beispielsweise Löschen durchgeführt werden sollen oder abgeschlossene Teile der Aktion erkannt wurden ([Herzer und Noll, 1990], Seite 97). Auch wenn diese Forderung allgemein gehalten ist, so gilt sie dennoch auch für Statusanzeigen.

[Oberquelle, 1994] fordert, dass eine Ausgabe wahrnehmbar sein soll. Dazu muss sie unter anderem erkennbar und mit Bezug auf die Bedeutung analysierbar sein (Seite 112). Die Wahrnehmbarkeit hängt auch von ihrem Kontext ab: Kann eine akustische Ausgabe trotz der Umgebungsgeräusche gehört werden, ist eine optische Ausgabe

auf dem Bildschirm sichtbar oder verdeckt? Analysierbarkeit bedeutet, dass aus der Darstellung die Semantik, also die objektive Bedeutung, und möglichst auch die Pragmatik für den Sender, also der Wert für sein Handeln, hervorgehen soll.

### **Inhaltliche Aspekte**

Für einen Programmierer ist es wichtig zu wissen, was das System gerade macht, und die Möglichkeit zu haben, Details des laufenden Prozesses kontrollieren zu können. Den normalen Benutzer jedoch verwirrt ein Übermaß an Information, schlimmstenfalls verunsichert es ihn sogar. Daher sollte der Benutzer lediglich erfahren, dass alles ordnungsgemäß durchgeführt wurde und nicht mit Details über die Ausführung belästigt werden. Detailliertes Feedback sollte auf eine subtile Art vermittelt werden, ohne den Arbeitsfluss des Benutzers unnötig zu unterbrechen. Insbesondere sollte der Arbeitsfluss nicht vom System unterbrochen werden, um einen normalen Zustand oder geringfügige Probleme mitzuteilen. Unterbrechungen sollten lediglich für schwerwiegende Ausnahmesituationen verwendet werden. Anstatt allen Benutzern solche Meldungen in einem Dialog aufzudrängen, sollte dem interessierten Benutzer eine Möglichkeit geboten werden, sich über den Status informieren zu können ([Cooper, 1995], Seiten 143-144).

Auch [Galitz, 1988] rät, nur Informationen auszugeben, die unerlässlich für das Treffen einer Entscheidung oder das Ausführen einer Handlung sind. Je mehr Informationen gleichzeitig präsentiert werden, desto mehr konkurrieren die einzelnen Informationen miteinander um die Aufmerksamkeit des Benutzers. Dadurch steigt die für das Finden der gerade gesuchten Information benötigte Zeit. Informationen sollten möglichst so präsentiert werden, dass ihre Bedeutung nicht erst in der Dokumentation nachgesehen werden muss oder der Benutzer die präsentierten Daten umwandeln muss, da auch dieses Zeit kostet ([Galitz, 1988], Seiten 69, 72). Das Gedächtnis des Benutzers sollte so wenig wie möglich belastet werden. Daher sollten Anzeigen für neue Benutzer oder Gelegenheitsnutzer Beschriftungen enthalten, damit er sich nicht an die Bedeutung erinnern muss ([Shneiderman, 1998], Seite 80).

### **Positives und negatives Feedback**

In der realen Welt ist Stille das Zeichen für „Fehler“, Geräusch ist der Normalzustand. Ein Beispiel hierfür ist das Zugriffsgeräusch der Festplatte. Unbewusste Wahrnehmung und Verarbeitung sind möglich. Dieses von [Cooper, 1995] so genannte *Positive Audible Feedback* entspricht der Natur des Menschen, der nicht gerne seine Fehler vorgehalten bekommt. *Positive Audible Feedback* gibt ihm entweder eine Bestätigung oder nichts. Im Gegensatz hierzu nennt [Cooper, 1995] die Ausgabe

eines Tons im Fehlerfall *Negative Audible Feedback*. Diese Töne sind in der Regel laut und unangenehm. In Gesellschaft anderer ist die Ausgabe von akustischen Fehlermeldungen oft peinlich für den Benutzer. *Positive Audible Feedback* darf nicht zu laut sein. Es sollte in der Lautstärke regelbar sein und einen angenehmen Klang haben ([Cooper, 1995], Seiten 454-456).

## Konsistenz

Für [Herzer und Noll, 1990] bedeutet Konsistenz, dass „*einmal eingeführte Arten des Dialogs oder Formen des Feedbacks für das gesamte System gelten.*“ ([Herzer und Noll, 1990], Seite 77) Konsistenz bedeutet für sie ganz pragmatisch nicht das strikte Einhalten von Prinzipien, sondern dass das System den Erwartungen des Benutzers möglichst nicht widerspricht. Konsistenz hilft dem Benutzer dabei, bekanntes Wissen in neuen, unbekanntem Situationen anzuwenden und sich so in ihnen besser zurechtzufinden. Fehlende Konsistenz ist schon lange ein Kritikpunkt. So beklagt [Nievergelt, 1983] beispielsweise „*mangelnde Konsistenz und das Fehlen von Systemverhaltensweisen, mit denen der Benutzer den Zustand des Systems bestimmen kann.*“ ([Nievergelt, 1983], Seite 12)

[Herzer und Noll, 1990] unterscheiden Konsistenz auf drei Ebenen<sup>1</sup>:

- *innerhalb eines Systems*
- *zwischen verschiedenen Systemen [...]*
- *zwischen System und mentalem Modell des Benutzers* ([Herzer und Noll, 1990], Seite 77)

Konsistenz kann bereits bei der Entwicklung eines Systems unterstützt werden, beispielsweise

- durch einen zentral definierten Standard, an den die Entwickler sich halten müssen, vorgegeben entweder lokal innerhalb einer Firma oder global durch Style Guides<sup>2</sup>,

---

1 [Nielsen, 1989] unterscheidet mehr Ebenen, auf denen Konsistenz existiert: Innerhalb einer einzelnen Applikation, innerhalb einer Produktfamilie, innerhalb aller Produkte eines Herstellers, innerhalb aller Programme, die auf einem bestimmten Computer laufen, innerhalb der Computerindustrie eines Landes und global, also für Jeden und überall. Außerdem weist er auf Konsistenz zwischen verschiedenen Medien hin, beispielsweise zwischen der Dokumentation und dem Programm auf dem Bildschirm. Ein Programm sollte außerdem konsistent zu den Erwartungen sein, die der Benutzer in anderen Bereichen erworben hat ([Nielsen, 1989], Seiten 5-7). Im Windows Style Guide von Microsoft wird Konsistenz ebenfalls auf drei Ebenen unterschieden: Konsistenz innerhalb des Anwendungsprogramms, innerhalb der Systemumgebung und Konsistenz mit Metaphern ([MSDN, 2000/2]). Für diese Arbeit reicht eine Unterscheidung auf drei Ebenen aus.

2 Ein Style Guide ist eine Richtlinie zur Gestaltung der Programmoberfläche, die vom Hersteller des Betriebssystems herausgegeben wird.

- durch Propaganda gegenüber den Entwicklern, bei der die Wichtigkeit herausgestellt wird (diese kann sowohl lokal innerhalb einer Firma als auch global durch sogenannte Evangelisten geleistet werden),
- durch gemeinsamen Code wie beispielsweise Bibliotheken oder Toolkits (auch hier ist sowohl die lokale als auch die globale Variante möglich), oder
- durch einige wenige Beispielprogramme, an denen die Entwickler weiterer Programme sich orientieren.

Bei einigen Systemen wird die Konsistenz nicht vom Entwickler, sondern vom Benutzer selbst sichergestellt. Die Benutzer passen also die Software an ihre Vorstellungen und Vorlieben an ([Nielsen, 1989], Seiten 5-7).

Konsistenz zwischen einem mentalen Modell des Benutzers und dem System ist nicht so einfach zu erreichen, hier gibt es keine Empfehlungen. Konsistenz innerhalb einer Anwendung und zwischen verschiedenen Systemen kann erreicht werden, wenn die folgenden Empfehlungen berücksichtigt werden:

- Verschiedene Bildschirme einer Anwendung sollten ein gleiches Grundformat besitzen (Kopf- und Fußbereich, Menübereich etc.).
- Bestimmte Informationen wie beispielsweise Systemmeldungen oder auch Hinweise, welche Aktionen zur Beendigung der momentanen Handlung durchgeführt werden müssen, sollten an gleicher Stelle mit gleicher Struktur angezeigt werden.
- Grafische Darstellung und Beschriftung von Daten sollten konsistent sein. Dazu gehören unter anderem ein einheitliches Farbschema und eine einheitliche Schrift. Auch Abkürzungen sollten standardisiert werden. Ausnahmen wie beispielsweise das Unterdrücken einer Passworteingabe sollten dem Benutzer intuitiv verständlich sein und nur selten vorkommen. Die Konsistenz der grafischen Darstellung und der Beschriftung von Daten kann beispielsweise aufgrund einer schriftlichen Richtlinie kontrolliert werden.
- Die Antwortzeiten des Systems sollten nicht variieren.
- Gleiche oder ähnliche Situationen sollten gleiche oder ähnliche Aktionen erfordern, gleiche oder ähnliche Aktionen sollten gleiche oder ähnliche Ausgaben zur Folge haben.
- Erklärungen sollten sich nicht widersprechen, insbesondere sollte sich das System verhalten, wie es im Benutzerhandbuch beschrieben wird ([Herzer und Noll, 1990], Seiten 77-78; [Shneiderman, 1998], Seiten 74, 80-81, 260, 386).



Interessant für die Darstellung von Statusanzeigen ist insbesondere die Empfehlung, Meldungen immer am selben Ort auszugeben. [Oberquelle, 1994] meint hierzu:

*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. So wie sich unsere räumliche Umgebung in der Regel nicht ohne unser Zutun verändert, sollte sich auch die visuelle Form der Ausgabe nur unter Benutzerkontrolle verändern. ([Oberquelle, 1994], Seite 113)*

Konsistenz ist also für das Bilden von Wahrnehmungsgewohnheiten wichtig und unterstützt die Bildung eines mentalen Modells beim Benutzer. Wenn es gute Gründe gibt, darf ein Programm jedoch auch inkonsistent sein ([Nielsen, 1989], Seite 7).

### **Relevanz**

Bei jeder Ausgabe ist zu prüfen, ob sie für den Benutzer relevant ist. Nicht relevante Ausgaben nehmen knappe Ressourcen wie beispielsweise den begrenzten Bildschirmplatz in Anspruch, oder sie tragen als überflüssige akustische Ausgabe zur akustischen Umweltverschmutzung bei. Dieses führt zu einer unnötigen Belastung des Benutzers, die ihn von seiner eigentlichen Aufgabe abhält. Berücksichtigung von Relevanz führt zu Empfehlungen wie in der von [Shneiderman, 1998] (Seiten 80-81) zitierten firmeninternen Richtlinie von Lockheed aus dem Jahr 1981, die empfiehlt, nur Daten anzuzeigen, die den Benutzer bei seiner Arbeit unterstützen.

Zunächst einmal ist eine Ausgabe nicht relevant, wenn sie keinen Bezug zu der vom Benutzer bearbeiteten Aufgabe hat. Aber auch eine Ausgabe, die der Benutzer selbst nach Lesen von Hilfe oder Handbuch nicht versteht, sowie eine Ausgabe, die keine Reaktion zulässt, sind für den Benutzer nicht relevant.

Ob eine Ausgabe für einen bestimmten Benutzer relevant ist oder nicht, hängt auch von dessen Kenntnisstand ab. Was für den Anfänger und den Gelegenheitsnutzer nicht relevant ist, kann für den Experten durchaus relevant sein. Ein generelles Fortlassen solcher Ausgaben ist nicht wünschenswert, da schließlich der Experte von ihnen profitiert. Sie sollten jedoch nicht unaufgefordert erscheinen, sondern explizit angefordert werden müssen. Damit ist sichergestellt, dass der Experte die für ihn relevanten Ausgaben erhalten kann, Anfänger und Gelegenheitsnutzer jedoch nicht mit nicht relevanten Ausgaben belästigt werden. Die Meldungen beim Start eines Systems sind ein Beispiel hierfür. Sie erscheinen in einer für Anfänger und Gelegenheitsnutzer unverständlichen Form und sind in der Regel auch für den Experten nur im Fehlerfall relevant ([Herzer und Noll, 1990], Seite 79).

Die Relevanz einer Ausgabe kann sich auch im Laufe der Zeit ändern: Eine Ausgabe, die zu einem bestimmten Zeitpunkt relevant ist, kann nach einiger Zeit nicht mehr relevant sein.

*Daher ist es sinnvoll zu fordern, dass jederzeit nur „wahre“, d.h. den momentanen Zustand beschreibende, Information auf dem Bildschirm stehen soll. ([Nievergelt, 1983], Seite 30)*

## **Redundanz**

Redundanz bedeutet, die selbe Nachricht über verschiedene Kanäle zu versenden ([Tognazzini, 1992], Seite 116). Redundante Nachrichten vermindern die Gefahr, dass die Nachricht verloren geht: Falls die Nachricht auf dem einen Kanal aus welchen Gründen auch immer nicht aufgenommen wird, steht ein anderer zur Verfügung. Redundanz bei Statusanzeigen kann erreicht werden, indem mehrere Sinne des Menschen gleichzeitig angesprochen werden, beispielsweise durch gleichzeitige optische und akustische Hinweise. Redundanz kann aber auch erreicht werden, indem verschiedene optische Merkmale kombiniert werden. So kann eine Statusänderung zum Beispiel durch einen Wechsel von einem roten Kreis zu einem grünen Dreieck visualisiert werden. Der Benutzer erhält die Information über den geänderten Status sowohl über die Änderung der Farbe als auch über die geänderte Form. [Tognazzini, 1992] (Seite 117) meint, Redundanz bei der Gestaltung würde die Schnittstelle angenehmer für den Benutzer machen.

Bei der Kommunikation sind nicht alle Kanäle gleichwertig. Einige Kanäle sollten lediglich zur Vermittlung redundanter Informationen verwendet werden. Farbe beispielsweise kann mit Rücksicht auf Farbenblinde nur als zusätzlicher, redundanter Kanal eingesetzt werden. Gleiches gilt für akustische Ausgaben: [Oberquelle, 1994] weist darauf hin, dass diese Signale von Unbeteiligten als störend empfunden werden können und daher abschaltbar sein müssen. Dabei muss aber die Kommunikation erhalten bleiben ([Oberquelle, 1994], Seite 114).

## **Stress**

Eine Person empfindet Stress bei einem unangenehm empfundenen Ungleichgewicht zwischen Anforderungen und persönlichen Voraussetzungen. Dieses Ungleichgewicht wird auch durch äußere Bedingungen beeinflusst. Ob ein Ungleichgewicht unangenehm empfunden wird, hängt von den Möglichkeiten ab, diese Bedingungen zu bewältigen. Das Gefühl der Kontrolle über die Situation ist ein Punkt, der stressmindernd wirkt. Stress kann also auf zwei Wegen vermieden werden: Entweder werden die Stress verursachenden Bedingungen eliminiert oder die Ressourcen zur

Bewältigung werden erhöht. An einem Computerarbeitsplatz können die folgenden Bedingungen Stress erzeugen:

- Nicht angemessenes Antwortverhalten des Systems, insbesondere zu lange oder sehr unregelmäßige Antwortzeiten.
- Leistungsdruck: Die ständige Eingabebereitschaft des Systems zusammen mit sehr kurzen Antwortzeiten können dem Benutzer das Gefühl geben, unter Druck zu stehen.
- Gefühl von Überwachung und Leistungskontrolle: Insbesondere Protokolle vom Arbeitsablauf tragen hierzu bei.
- Abstraktheit der Arbeit.

Ausführliche Information über den laufenden Prozess gibt dem Benutzer das Gefühl der Kontrolle und wirkt dadurch stressmindernd. Dazu sollte mindestens bei längeren Wartezeiten eine Abschätzung der Dauer gegeben werden. Durch das Verwenden von Metaphern bei der Gestaltung der Statusanzeige wird der Abstraktheit entgegengewirkt. Die Eigenschaften der verwendeten Metapher sollten möglichst konkret implementiert werden. Textuelle Anzeigen sollten in voller Länge und entschlüsselt dargestellt werden ([Herzer und Noll, 1990], Seiten 88-90).

### **Anpassen der Anzeige durch den Benutzer**

Auch wenn [Shneiderman, 1998] (Seite 80) seine Forderung, dass der Benutzer die Information der Anzeige in der für ihn angenehmsten Art und Weise entnehmen können sollte und die Anzeige dazu vom Benutzer anzupassen sein sollte, nur auf optische Anzeigen bezog, so gilt diese Aussage auch für akustische Ausgaben. Eine Anpassung der Lautstärke ist wichtig, um Andere nicht zu stören. Und auch eine individuelle Einstellung des Klangs ist nicht nur eine Geschmacksfrage, sondern kann in einigen Situationen ebenfalls unverzichtbar sein, beispielsweise zur Unterscheidung der Signale verschiedener Arbeitsplätze (man stelle sich ein Büro vor, in dem alle Telefone den gleichen Klingelton von sich geben).

## **4.2 Optische Ausgaben**

Optische Ausgaben sind für die Interaktion zwischen Mensch und Computer von sehr großer Bedeutung, nahezu sämtliche Ausgaben des Computers werden optisch präsentiert. Die optische Präsentation von Information ermöglicht eine Vielzahl verschiedener Betrachtungsweisen. Anders als Sprache sind optische Ausgaben ein vom Empfänger zu kontrollierender Kanal ([Tufté, 1990], Seite 31).

In diesem Abschnitt werde ich zunächst zeigen, mit welchen Eigenschaften optische Ausgaben beschrieben werden können. Auf einige für die Interaktion zwischen Mensch und Computer wichtige Eigenschaften gehe ich anschließend näher ein.

Optische Ausgaben können, unabhängig von physikalischen Eigenschaften, inhaltlich unterteilt werden in grafische Ausgaben, also Bilder, Zeichnungen etc., und Texte. Ich werde auch hierfür auf die wesentlichen Punkte eingehen.

### **Allgemeines**

Subjektive Beurteilungen, Augenbewegungen und Suchzeiten für Information auf dem Bildschirm hängen sehr stark von Faktoren wie lokaler Dichte, Gesamtdichte, Gruppenbildung und Komplexität des Layouts ab. Ein Bildschirm sollte möglichst alle wichtigen Informationen gleichzeitig anzeigen, jedoch sollte er auch nicht mit Informationen überladen werden. Die Reihenfolge, in der etwas ausgegeben wird, kann sich auf Entscheidungen des Benutzers auswirken. So wird bei gleichzeitiger Darstellung oft die zweite Position bevorzugt ([Herzer und Noll, 1990], Seite 48).

### **Häufige Fehler**

[Mullet und Sano, 1995] (Seite 29) führen häufige Fehler bei Programmen mit grafischer Oberfläche auf:

- Der Bildschirm wird überfrachtet, der Inhalt ist ungeordnet
- zu ausführliche Beschreibung von Elementen
- optische Strukturierung von unstrukturiertem Inhalt
- zu detaillierte Instruktionen, die höchstens einmal benötigt werden
- zu buchstäbliche Umsetzung von Gegenständen und Konzepten aus der realen Welt, ohne auf die anderen Bedingungen und Möglichkeiten des Rechners Rücksicht zu nehmen
- zu detailreiche Gestaltung mit zu viel Verzierungen
- unangebrachte Verwendung von Dreidimensionalität

### **Begriffsklärungen**

Die in diesem Abschnitt eingeführten Begriffe basieren auf der Arbeit von [Mullet und Sano, 1995] (Seiten 52 ff.). [Mullet und Sano, 1995] charakterisieren optische Information durch die Zahl der Dimensionen (die messbare Größe), die Skala der

Dimension (nominal, geordnet oder quantitativ) und die Länge der Dimension (die Anzahl der möglichen Werte).

### Eigenschaften/Dimensionen visueller Objekte

Visuelle Objekte können unter anderem durch folgende Eigenschaften bzw. Dimensionen beschrieben werden:

- *Farbton* (Orange, Blau, Rot, Gelb, Grün, ...)
- *Helligkeit* (helle, leuchtende Farbe oder dunkle Farbe)
- *Sättigung* (reiner Farbton oder gebrochener Farbton)
- *Textur* (Musterung)
- *Form* (Quadrat, Dreieck, Kreis, ...)
- *Größe* (Groß oder Klein)
- *Orientierung* (senkrecht, waagrecht, diagonal, auf dem Kopf stehend, ...)
- *Position* (oben, unten, rechts, links, Mitte, ...)

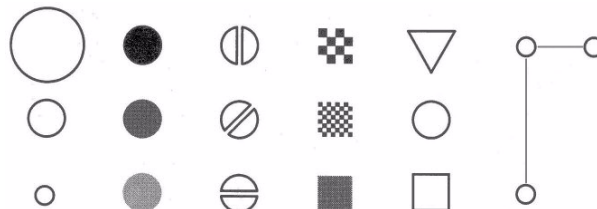


Abbildung 15: Wahrnehmungseigenschaften ([Mullet und Sano, 1995], Figure 52, Seite 54)  
 Von links nach rechts: Größe, Helligkeit, Orientierung, Textur, Form, (relative) Position. Nicht abgebildet sind Farbton und Sättigung.

### Skala einer Dimension

Bei einer *nominalen Skala* muss der Benutzer Dinge nur kategorisieren bzw. benennen. Farbton, Textur und Form sind nominale Eigenschaften. Bei einer *geordneten Skala* kann der Betrachter eine relative Reihenfolge, also eine „mehr als“-Beziehung, herstellen. Helligkeit, Sättigung, Größe, Orientierung und Position sind geordnet. Bei einer *quantitativen Skala* kann der Betrachter die Größe des Unterschieds zwischen zwei geordneten Werten augenblicklich bestimmen. Nur Größe, Orientierung und Position werden mit einer quantitativen Skala gemessen, denn nur diese Dimensionen erlauben eine gute Schätzung des Unterschieds allein aufgrund der visuellen Wahrnehmung.

### Länge einer Dimension

Die visuellen Variablen unterscheiden sich stark in ihrer Länge und der Anzahl der wahrnehmbaren Unterschiede. Form ist die Variable mit der größten Länge, es gibt praktisch unendlich viele unterscheidbare Formen. Theoretisch ist auch Position von

unendlicher Länge, in der Praxis jedoch setzen Größe und Auflösung des Bildschirms endliche Grenzen. Weil aber selbst kleinste relative Unterschiede einfach zu erkennen sind, unterstützt Position dennoch mehr feine Unterschiede als alle anderen Variablen. Orientierung ist die kürzeste Dimension. Mehr als vier verschiedene Stufen verwirren den Betrachter. Die anderen Dimensionen liegen irgendwo dazwischen: Sättigung und Textur unterstützen weniger als zehn Stufen, Farbe und Größe etwas mehr.

### **Assoziative Dimension**

Eine visuelle Dimension heißt *assoziativ*, wenn sie die Sichtbarkeit anderer Dimensionen nicht beeinflusst. Beispielsweise kann der Farbton unabhängig von der Ausrichtung des Objekts bestimmt werden, aber es ist sehr schwierig, den Farbton eines sehr kleinen Objekts zu bestimmen. Alle visuellen Variablen außer Größe und Sättigung sind assoziativ.

### **Selektive Dimension**

Eine visuelle Dimension heißt *selektiv*, wenn der Benutzer alle Objekte mit gleichem Wert der Dimension auf einmal wahrnimmt. Alle Dimensionen bis auf Form sind selektiv – Formen müssen individuell mit fokussierter Aufmerksamkeit erkannt werden.

### **Kontrast**

Ein deutlicher Unterschied des Werts einer Dimension heißt *Kontrast*. Kontrast bildet die Basis für visuelle Unterscheidungsmöglichkeiten, ohne die keine optischen Nachrichten möglich wären. Visueller Kontrast wird durch gezieltes Beeinflussen mindestens einer Dimension erreicht. Effektives visuelles Design bedeutet, für jede gewünschte Kommunikation diejenigen Dimensionen auszuwählen, mit denen das Ziel der Kommunikation am besten erreicht wird.

### **Farbe**

Vor einigen Jahren konnten Farbbildschirme noch nicht als selbstverständlich vorausgesetzt werden. Eine wichtige Empfehlung war daher, primär für Monochrom-Bildschirme zu entwerfen und anschließend farbige Akzente zu setzen. Heute sind Farbbildschirme Standard, so dass der Entwurf eines Bildschirmlayouts gleich unter Berücksichtigung von Farbe erfolgen kann. Das eröffnet dem Bildschirmdesigner neue Möglichkeiten, stellt ihn aber auch vor neue Herausforderungen. Der Einsatz

von Farbe bringt nämlich nicht nur Vorteile, sondern birgt auch (oft versteckte) Gefahren in sich.

### **Potenzielle Vorteile und Gefahren**

Auf den ersten Blick kann der Einsatz von Farbe viele Vorteile bieten:

- Farbe erleichtert die Erkennung bei vielen Aufgaben. Beispielsweise können in einer Liste von Buchungen die interessierenden Buchungen farbig hervorgehoben werden. Eine ungünstig gewählte Farbkodierung kann jedoch auch die Arbeit erschweren und die Leistung mindern ([Shneiderman, 1998], Seite 399). So muss bei Farbkodierung z.B. aufgepasst werden, dass nicht die unwichtigsten Werte die auffälligsten Farben erhalten, nur weil es durch eine Metapher so vorgegeben wurde. So wurde zum Beispiel versucht, das Altern von Dokumenten durch Vergilben darzustellen. Je älter das Dokument war, desto satter wurde das Gelb. Das Ergebnis war, dass das älteste Dokument die auffälligste Farbe hatte, während das neueste Dokument unauffällig weiß war ([Salomon, 1991], Seiten 272-273).
- Farbe kann beruhigen oder ins Auge fallen. Mit Farbe kann Leben in langweilige Bildschirmlayouts gebracht werden. Farbe ermöglicht feinste Unterscheidungen bei komplexen Bildschirmlayouts. Eine Farbkodierung kann die logische Struktur der präsentierten Daten hervorheben. Bestimmte Farben erregen Aufmerksamkeit für Warnungen. Farbe kann emotionale Reaktionen wie Freude, Aufregung, Furcht oder Ärger hervorrufen ([Shneiderman, 1998], Seite 402). Daher sind beispielsweise Schulen und Krankenhäuser in der Regel in bestimmten Farben gestrichen ([Salomon, 1991], Seite 274).
- Farbe kann Strukturen verbergen oder hervorheben. So wurde ein Wassertank in einem kalifornischen Ort von vielen Leuten erstmalig bemerkt, als seine Farbe von Himmelblau nach Grün geändert wurde. Analog hierzu könnte der Computer Objekte, die bestimmten Kriterien entsprechen, verbergen oder hervorheben. Unwichtige Objekte könnten beispielsweise buchstäblich im Schatten versteckt werden, oder mit schwächer gesättigten Farben und verschwommenen Konturen kann räumliche Tiefe dargestellt werden ([Salomon, 1991], Seiten 277-278).
- Farbe kann Assoziationen und Erinnerungen hervorrufen. Die Farbe der Filmschachtel beispielsweise führt zu einer Erinnerung an die Marke. Farben in Computerprogrammen können beim Wiedererkennen und Erinnern helfen, wenn sie in bestimmten Kontexten verwendet werden ([Salomon, 1991], Seite 275).
- Farbe ist ein natürliches Mittel zur Quantifizierung. Variationen der Sättigung und der Helligkeit führen zu einer kontinuierlichen Skala mit feinsten Unterschei-

dungsmöglichkeiten ([Tufté, 1990], Seite 91). Auch wenn Helligkeitskodierte Skalen einfach zu lernen und zu erinnern sind, so sind sie doch anfällig gegen Fehler beim Ablesen. Eine gebräuchliche Alternative zu einer Skala mit Helligkeitsstufen ist eine Skala mit Regenbogenfarben. Jedoch kann bei dieser Skala die natürliche Reihenfolge nicht spontan erfasst werden, sie muss bewusst erinnert werden. Eine weitere Alternative zu einer Skala mit Helligkeitsstufen ist eine zusätzliche leichte Variation des Farbtons, die dem Betrachter dabei hilft, Unterschiede zu erkennen ([Tufté, 1990], Seiten 92-93).

Der Einsatz von Farbe muss jedoch gut überlegt werden. Physiologische und physikalische Rahmenbedingungen verändern Wahrnehmung und Wahrnehmbarkeit einer Farbe, können aber nicht vom Systementwickler beeinflusst werden ([Salomon, 1991], Seite 270; [Shneiderman, 1998], Seite 402; [Tufté, 1990], Seiten 92-93):

- Bestimmte Farbkombinationen vermindern die Lesbarkeit, sei es wegen physiologischer Probleme, zu geringem Kontrast oder gleicher Eigenhelligkeit.
- Die Farbtreue an einem anderen Arbeitsplatz ist nicht gewährleistet, die Farbwiedergabe kann außerdem beim Drucken oder einer Konvertierung zu anderen Medien leiden.
- Die Wirkung einer Farbe hängt von der Beleuchtung ab: Bei Tageslicht, Neonlicht oder Glühlampenlicht wirkt die selbe Farbe unterschiedlich.
- Größe, Form und Platzierung beeinflussen die Wirkung einer Farbe, ebenso wie benachbarte Farben.

Zusätzlich müssen auch semantische Aspekte berücksichtigt werden: Der Einsatz von Farbe ist dann problematisch, wenn der Benutzer Farbunterschiede bemerkt und eine nicht vorhandene Bedeutung hineininterpretiert, oder wenn durch Verwendung der gleichen Farbe sachlich nicht zusammengehörende Dinge optisch zusammengefasst werden ([Galitz, 1988], Seite 205).

### **Konventionen**

Es gibt verschiedene Konventionen zur Verwendung von Farbe. Einige sind natürlich gegeben, andere sind sehr stark von der jeweiligen Kultur abhängig. Die Verwendung von Rot als Farbe für Warnungen beispielsweise gehört zu den genutzten natürlichen Instinkten: Rot ist eine der Farben des Feuers, und Feuer schmerzt ([Tognazzini, 1992], Seite 169).

Menschen sind es gewöhnt, dass Farbe Informationen über die Zeit gibt: Der Wechsel der Jahreszeiten geht einher mit sich verändernden Farben, ein Blick auf die



Blätter an den Bäumen reicht zur groben Bestimmung der Jahreszeit. Ein Blick auf den Himmel reicht, um die ungefähre Tageszeit zu ermitteln. Falls Genauigkeit nicht erforderlich ist, kann Farbe zur subtilen Vermittlung von Informationen genutzt werden. Sie kann den Fortschritt eines Computer-Prozesses anzeigen, oder die relative CPU-Kapazität, die ein bestimmter Prozess benötigt ([Salomon, 1991], Seite 277). Diese Assoziationen gelten zwar unabhängig von einem speziellen Kulturkreis, jedoch können andere Faktoren wie die geografische Lage das Erkennen erschweren oder gar unmöglich machen. In der Nähe des Äquators beispielsweise ist der Wechsel der Jahreszeiten nicht anhand der Farbe festzustellen.

Die Assoziation von Orange und Schwarz mit Halloween hingegen ist eine kulturelle Assoziation ([Salomon, 1991], Seite 273), ebenso die Assoziation eines gelben (New York) oder beigen Autos (Deutschland) mit einer Taxe. Die Assoziationen eines farbigen Objekts kann in verschiedenen Kulturkreisen unterschiedliche Bedeutung haben: Ein gelbes Auto in New York ist eine Taxe, ein gelbes Auto in Deutschland hingegen ein Postauto. Diese Assoziationen können erlernt werden und werden oft auch aus dem Kontext verständlich.

Viele Farben werden in verschiedenen Kulturen mit unterschiedlichen Begriffen in Verbindung gebracht. Tabelle 4 zeigt für einige Farben und Kulturen die zugehörigen Assoziationen:

Kultur / Farbe	Rot	Blau	Grün	Gelb	Weiß
<b>USA</b>	Gefahr	Maskulinität	Sicherheit	Feigheit	Reinheit
<b>Frankreich</b>	Aristokratie	Freiheit, Frieden	Kriminalität	Vorübergehend	Neutralität
<b>Ägypten</b>	Tod	Tugend, Glaube, Wahrheit	Fruchtbarkeit, Stärke	Glück, Wohlstand	Freude
<b>Indien</b>	Leben, Kreativität		Wohlstand, Fruchtbarkeit	Erfolg	Tod, Neutralität
<b>Japan</b>	Zorn, Gefahr	Schurkerei	Zukunft, Jugend, Energie	Anmut, Adel	Tod
<b>China</b>	Glück	Wolken des Himmels	Ming Dynastie, Wolken des Himmels	Geburt, Reichtum, Macht	Tod, Neutralität

Tabelle 4: Kulturelle Assoziationen zu Farben (nach [Russo und Boor, 1993], Tabelle 1, Seite 346)

Eine Farbe kann sogar innerhalb der selben Kultur in verschiedenen Kontexten verschiedene Bedeutungen haben: Aus dem Straßenverkehr ist Rot für Stop, Gelb für Warnung und Grün für OK bekannt. Ist Geld im Spiel, bedeutet Rot einen Verlust und Schwarz einen Gewinn. Und im Badezimmer ist Rot gleichbedeutend mit heiß und Blau gleichbedeutend mit kalt. Verschiedene Menschen können ein rotes Objekt in einem anderen Kontext sehen und ihm so widersprüchliche Bedeutungen zuspre-

chen. Beispielsweise kann der Konstrukteur eines Autos eine rote Kontrollleuchte einbauen mit dem Gedanken: „Rot bedeutet, der Motor ist warm und hat Betriebstemperatur.“, während der Benutzer denkt: „Rot bedeutet 'Stopp!', es gibt ein ernstes Problem mit dem Motor.“ Der Entwickler sollte die verwendete Farbkodierung also möglichst erklären ([Shneiderman, 1998], Seite 400).

### **Empfehlungen zur Verwendung von Farbe**

Ich unterteile die Empfehlungen zur Verwendung von Farbe in allgemeine Empfehlungen und Empfehlungen zur Farbkodierung<sup>1</sup>. Die allgemeinen Empfehlungen sollten unabhängig von der zu lösenden Aufgabe berücksichtigt werden. Bei der Farbkodierung sind einige spezielle Punkte zusätzlich zu berücksichtigen.

Die Empfehlungen in diesem Abschnitt wurden von [Shneiderman, 1998] (Seiten 399-402), [Herzer und Noll, 1990] (Seite 49), [Tuft, 1990] (Seiten 90, 94), [Galitz, 1988] (Seiten 204, 216) und [Salomon, 1991] (Seiten 272-273) ausgesprochen. Physiologische Aspekte sind in diesen Empfehlungen nicht enthalten, da diese bereits weiter oben in dem Abschnitt über Farbwahrnehmung (Seiten 16 ff.) enthalten sind.

#### Allgemeine Empfehlungen

- Es sollten nur wenige verschiedene Farben verwendet werden.
- Farbe sollte sparsam verwendet werden.
- Es sollten nur Farben mit ausreichendem Kontrast verwendet werden.
- Es muss bedacht werden, dass Farbe die Erledigung einer Aufgabe sowohl beschleunigen als auch verlangsamen kann.
- Farben sollten möglichst aus der Natur gewählt werden. Speziell hellere Töne, Blau, Gelb und Grautöne aus Himmel oder Schatten sind gut geeignet. Natürliche Farben sind dem Benutzer bereits bekannt, und sie passen zueinander.
- Dezentere Linien eines ähnlichen Farbtons um einfarbige Flächen erhöhen die Abgrenzung zu benachbarten Flächen und den Zusammenhalt der Fläche selbst. Generell gilt, dass Menschen bei der Bildverarbeitung sehr stark Konturen von Objekten nutzen (siehe Seite 16).
- Farbe kann zur Anzeige von Statusänderungen verwendet werden. Farbe erregt Aufmerksamkeit, was insbesondere bei einer großen Zahl gleichzeitig zu beob-

---

<sup>1</sup> Farbkodierung bedeutet, verschiedene, meist numerische Werte mit Hilfe von Farbe darzustellen. Diese Technik wird beispielsweise bei Landkarten verwendet, um Erhebungen und Meerestiefen zu zeigen.

achtender Werte hilfreich ist. Für normale Zustände sollten Grün, Weiß oder Blau verwendet werden, für Warnungen Gelb oder Gold, für Notfälle oder Ausnahmen Rot.

- Farbe kann zur Unterstützung beim Formatieren der Informationen auf dem Bildschirm verwenden.
- Farbe kann verwendet werden, um die Informationsdichte zu erhöhen.

#### Empfehlungen zur Farbkodierung

- Es muss sichergestellt werden, dass die Farbkodierung die aktuelle Aufgabe unterstützt. Daher sollten nur die für den Benutzer für diese Aufgabe relevanten Informationen farbig hervorgehoben werden. Andere Hervorhebungen können für die Bearbeitung der aktuellen Aufgabe kontraproduktiv sein.
- Im ganzen System sollte eine konsistente Farbkodierung verwendet werden.
- Allgemeine Erwartungen zur Farbkodierung sollten berücksichtigt werden.
- Werden wenig intuitive Farben zur Kodierung benutzt, sollte eine Legende angeboten werden.
- Die Farbkodierung sollte vom Benutzer kontrolliert werden können.
- Um verschiedene Werte einer Skala darzustellen, sollten Sättigung und Helligkeit variiert werden. Diese Abstufungen werden intuitiv verstanden. Eine Skala mit verschiedenen Farbtönen ist nicht intuitiv verständlich, selbst wenn die Ordnung der Farben dem natürlichen Farbspektrum entspricht. So kennen beispielsweise die wenigsten Menschen die Reihenfolge der Farben im Farbspektrum, die sogenannten Regenbogenfarben.

#### **Position auf dem Bildschirm**

Es gibt verschiedene Aspekte, mit denen man die Position einer Bildschirmausgabe beschreiben kann ([Shneiderman, 1998], Seite 376):

- Räumlich: Ist die Ausgabe zentral oder peripher? Oben oder unten? Rechts oder links?
- Aufgabenbezogen: Ist sie in der Nähe weiterer relevanter Informationen, an immer der selben nicht störenden Stelle oder ein separater Dialog in Bildschirmmitte?

Ein Beispiel zeigt die unterschiedlichen Sichtweisen: Drei Programme reagieren auf eine falsche Eingabe in einem Eingabefeld. Programm A gibt eine Fehlermeldung in der unteren Zeile des Bildschirms aus, Programm B öffnet einen Dialog in der Mitte

des Bildschirms und Programm C schreibt die Fehlermeldung rechts neben das Eingabefeld. Räumlich gesehen ist die Ausgabe von Programm A unten, die von Programm B in der Mitte und die von Programm C rechts. Aufgabenbezogen gesehen erfolgt die Fehlermeldung von Programm A immer an der selben nicht störenden Stelle, die von Programm B in einem separaten Dialog in Bildschirmmitte und die von Programm C in der Nähe der relevanten Daten.

Eine allgemein gültige Empfehlung für die Position von Bildschirmausgaben gibt es nicht. Hier kommt es sehr stark darauf an, welche Informationen ausgegeben werden sollen. [Shneiderman, 1998] (Seite 379) empfiehlt, für die Ausgabe von Fehlermeldungen immer die selbe Stelle zu verwenden. Diese Auffassung vertritt auch [Galitz, 1988] (Seiten 59-61, 165). Er empfiehlt, bestimmte Bereiche des Bildschirms für bestimmte Arten von Information zu reservieren, beispielsweise die vorletzte Zeile des Bildschirms für Status- und Fehlermeldungen. Diese Position solle dann konsistent bei allen Bildschirmen genutzt werden. Dieses diene nicht zuletzt der Identifikation dieser Informationen. [Brodbeck und Rupiotta, 1994] empfehlen, den aktuellen Programmmodus nicht in einer Statuszeile anzuzeigen, sondern stattdessen den Mauszeiger zu ändern. Diese Rückmeldung sei wesentlich günstiger, „weil sie dort erfolgt, wo die Aufmerksamkeit des Benutzers liegt.“ (Seite 206) [Cooper, 1995] fordert, dass Software Informationen präsentiert wie ein HUD<sup>1</sup>, das Programm also die Kanten des Bildschirms nutzt, um dem Benutzer Informationen über die Aktionen im Zentrum des Bildschirms zu präsentieren (Seite 131). [Nievergelt, 1983] hingegen fordert lediglich, dass der Benutzer die Ausgabe sehen können muss (Seiten 30-33).

Auf den ersten Blick gibt es keine eindeutige Empfehlung: Einige Autoren wie beispielsweise [Shneiderman, 1998] oder [Galitz, 1988] legen die Bildschirmausgabe anhand räumlicher Kriterien fest, andere wie beispielsweise [Brodbeck und Rupiotta, 1994] empfehlen, Statusinformationen direkt im Aufmerksamkeitsbereich des Benutzers auszugeben. Auch lässt sich nicht feststellen, dass sich die Auffassungen im Laufe der Zeit geändert haben. Die Forderung von [Nievergelt, 1983], die Ausgabe müsse so positioniert werden, dass der Benutzer sie sehen könne, also entweder nahe der letzten Eingabe oder an einer hierfür reservierten Stelle (Seiten 32-33), bedeutet nichts anderes als dass die Ausgabe dort stattfinden soll, wo sie am besten wahrnehmbar ist. Als Beispiel hierfür dient Windows: Einige Statusinformationen erscheinen konstant in der unteren Bildschirmzeile (geladene Programme, aktives Programm, geladene Treiber etc.), andere Informationen werden durch geänderte

---

<sup>1</sup> HUD = Head Up Display, eine in Kampfflugzeugen übliche Methode, Daten anzuzeigen. Die auszugebenden Informationen werden entweder auf die Scheibe des Cockpits oder des Helms projiziert, so dass der Pilot den Kopf nicht bewegen muss, um weiter unten liegende Instrumente zu sehen. Ähnliche Forschungen gibt es bereits seit einigen Jahren auch im Kfz-Bereich.

Mauszeigerform vermittelt (bei Drag-and-Drop-Aktionen beispielsweise, ob das Objekt kopiert, verschoben oder eine Verknüpfung zu ihm erstellt werden soll).

Die bisherigen Überlegungen bezüglich der Positionierung der Daten haben aufgabenbezogene Aspekte berücksichtigt. [Mullet und Sano, 1995] geben zu bedenken, dass Objekte im peripheren Gesichtsfeld, die nur für kurze Zeit erscheinen, zwar wahrgenommen werden („Da ist irgendetwas“), aber nur schwer zu erkennen sind (Seite 169).

Hieraus leite ich die Empfehlung ab, dass Statusinformationen, die direkten Einfluss auf die weitere Arbeit haben (insbesondere Informationen über den Modus des Programms), möglichst direkt im Aufmerksamkeitsbereich des Benutzers angezeigt werden sollten, während Informationen, die auf die weitere Arbeit keinen direkten Einfluss haben (mehr oder weniger zusätzliche Informationen wie beispielsweise Seitennummer oder Prozessorauslastung), an einem festen Ort am Bildschirmrand angezeigt werden sollten. Informationen mit direktem Einfluss auf die weitere Arbeit sollten vom Benutzer sofort wahrgenommen werden können und daher nicht am Bildschirmrand, sondern im Aufmerksamkeitsbereich des Benutzers stehen.

## **Animation**

### **Was ist Animation?**

Animationen auf Computerbildschirmen sind eine Folge einzelner Bilder, die in mehr oder weniger kurzen Intervallen aufeinander folgen. Animation können unter verschiedenen Gesichtspunkten betrachtet werden:

- Wie schnell folgen die einzelnen Bilder aufeinander? Die beiden Extreme sind filmähnliche Animationen mit etlichen Bildwechseln pro Sekunde auf der einen Seite sowie gelegentliche Bildwechsel, die in der Regel bei Zustandsänderungen des repräsentierten Objekts auftreten. Hier kann es Tage, Wochen oder noch länger dauern, bis sich etwas ändert.
- Handelt es sich um eine einmalige Animation, oder ist es eine endlos lang laufende Animation?
- Welchen Zweck verfolgt die Animation? Soll sie die Aufmerksamkeit des Benutzers erregen, Statusinformationen vermitteln, unterhalten oder lehren?

Für [Baecker und Small, 1991] ist Animation nicht die Kunst der Zeichnungen, die sich bewegen, sondern die Kunst der Bewegungen, die gezeichnet sind. Wichtiger als die einzelnen Bilder ist der Übergang zwischen ihnen. Animation ist für sie die Kunst, den unsichtbaren Zwischenraum zwischen zwei Bildern zu gestalten (Seite

251). Bei einfachsten Animationen wie beispielsweise einem blinkenden Cursor oder den glücklicherweise aus der Mode gekommenen blinkenden Bildern auf Internetseiten ist starker Kontrast extrem wichtig, um die Aufmerksamkeit des Benutzers zu erregen. Andere Animationen, die einen sich kontinuierlich ändernden Zustand vermitteln, sollten subtilere Übergänge zwischen den einzelnen Bildern haben. Die Änderung zwischen zwei Bildern einer Animation sollte die Änderung des angezeigten Zustands abbilden – vorausgesetzt, es soll mit der Animation Information vermittelt werden und der Sinn der Animationen ist nicht bloße Effekthascherei.

[Gonzales, 1996] definiert Animationen als eine Folge wechselnder Bilder, die aufgrund von Aktionen des Benutzers dynamisch präsentiert werden und dem Benutzer helfen, eine kontinuierliche Veränderung über einen Zeitraum wahrzunehmen und ein besseres mentales Modell der Aufgabe zu entwickeln (Seite 27).

Animation kann sowohl die Bewegung eines Objekts von einem Ort zu einem anderen als auch die Transformation eines Objekts bedeuten, beispielsweise Fenster, die aus einem Symbol „wachsen“ und wieder zu diesem Symbol „schrumpfen“. Das Icon einer zu druckende Datei könnte sich erst verdoppeln, eine Kopie könnte dann sukzessive im Drucker verschwinden. Nicht nur vom Benutzer gesteuerte Prozesse wie das Ändern der Fenstergröße sollten animiert dargestellt werden, sondern auch vom System gesteuerte wie beispielsweise das Einfügen einer Spalte in das Tabellenarbeitsblatt ([Dutke, 1994], Seite 137).

### **Was kann Animation?**

Animation kann die Komplexität der Bedienoberfläche reduzieren. Mit Animationen kann gezeigt werden, was getan wurde, was getan werden kann und was nicht. Animation kann den Benutzer auf notwendige Aktionen hinweisen, aber auch auf nicht erlaubte ([Baecker und Small, 1991], Seite 257). Animationen machen eine Benutzungsschnittstelle besser bedienbar, angenehmer und verständlicher ([Gonzales, 1996], Seite 27). Zehn grundlegende Fragen kann der Benutzer mit Hilfe von Animationen beantworten ([Baecker, Small und Mander, 1991], Seite 1):

- *Identifikation*: Was ist das?
- *Transition*: Woher komme ich, wohin gehe ich?
- *Orientierung*: Wo bin ich?
- *Auswahl*: Was kann ich jetzt tun?
- *Demonstration*: Was kann ich hiermit tun?
- *Erklärung*: Wie mache ich etwas?

- *Feedback*: Was passiert gerade?
- *Historie*: Was habe ich getan?
- *Interpretation*: Warum ist genau das passiert?
- *Anleitung*: Was sollte ich jetzt tun?

Eigenschaften eines Objekts können entweder statisch (unveränderlich) oder dynamisch (veränderlich) sein. Beispielsweise ist der Typ eines Objekts statisch, sein Inhalt dagegen dynamisch. Obwohl möglichst alle für den Benutzer relevanten Eigenschaften

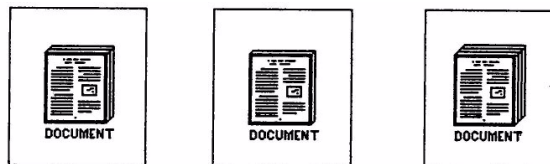


Abbildung 16: Größe einer Datei als dynamisches Icon ( [Herzer und Noll, 1990], Bild 20, Seite 64)

eines Objekts in seiner Repräsentation sichtbar sein sollten, werden dynamische Eigenschaften jedoch nur selten angezeigt. Animationen können diese Lücke schließen. Beispielsweise kann die Größe eines Dokuments wie in Abbildung 16 durch verschieden dicke Icons repräsentiert werden.

Mit Hilfe dynamischer Eigenschaften kann der Benutzer ein gesuchtes Objekt schneller wiederfinden, da seine Repräsentation sich von der anderer Objekte des selben Typs unterscheidet. Er muss sich hierfür erinnern, welche Eigenschaften das gesuchte Objekt von den anderen Objekten gleichen Typs unterscheidet und diese Eigenschaften auf dem Bildschirm suchen. Die mit den von [Tognazzini, 1996] so genannten „slow animations“ repräsentierten Eigenschaften helfen umgekehrt auch beim Wiedererkennen, d.h. der Benutzer sieht die Eigenschaften eines repräsentierten Objekts und kann sich anhand dieser Eigenschaften erinnern, um welches Objekt es sich handelt. Dynamische Eigenschaften, die mit Hilfe von Animationen dargestellt werden können, sind beispielsweise Alter, Anzahl der Zugriffe oder Größe ([Herzer und Noll, 1990], Seiten 63-64; [Tognazzini, 1996], Seiten 209 ff.).

Mit Animation können Informationen visualisiert werden, die mit anderen Mitteln nicht dargestellt werden können, beispielsweise der Fortschritt beim Ausführen eines Programms ([Mountford,

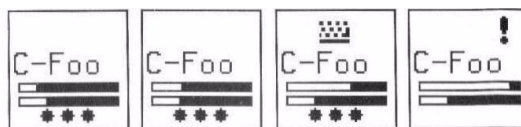


Abbildung 17: Animiertes Icon ([Baecker und Small, 1991], Seite 263)

1991], Seite 249). Animierte Icons können dem Benutzer Informationen über einen im Hintergrund laufenden Prozess vermitteln. Abbildung 17 zeigt vier Stadien eines animierten Icons eines C-Compilers beim Übersetzen der Datei *Foo*. Der obere Balken zeigt den Prozessfortschritt an, die drei Sterne informieren, dass das Fenster momentan nicht angezeigt wird. Im dritten Bild fordert der Compiler eine Tastatur-

eingabe des Benutzers an. Im vierten Bild zeigt das Ausrufezeichen, dass das nun sichtbare Fenster die Aufmerksamkeit des Benutzers benötigt.

Einfache Animationen können dem Benutzer Informationen mit Hilfe von Skalierung, Transparenz, Farbänderung und Balkengrafiken vermitteln. Ein Icon eines Prozesses, der die Aufmerksamkeit des Benutzers benötigt, könnte z.B. mit einem immer intensiveren Rot eingefärbt werden, je länger der Prozess bereits wartet. Farbe könnte auch anzeigen, welche Ressourcen ein Prozess benötigt. Ein Prozess, der lediglich Rechenzeit benötigt, könnte ein blaues Icon erhalten, ein Prozess, der nur Ein- und Ausgabe benötigt, ein gelbes. Prozesse, die beides benötigen, erhalten eine Farbe dazwischen. Mit der Größe des Icons könnte zusätzlich angezeigt werden, wie viel CPU-Leistung der Prozess beansprucht. Es liegt nahe, dass der Versuch, all diese Informationen für jeden laufenden Prozess zu vermitteln, zu einem wild blinkenden Display mit einer so riesigen Menge an Informationen führt, die der Benutzer nur schwer erfassen kann. Daher ist gutes Design hier extrem wichtig ([Baecker und Small, 1991], Seiten 262-263). Ein Benutzertest zeigte, dass Animationen möglichst nicht abstrakte, sondern realistische Bilder verwenden sollen. Die Bewegung sollte nicht abrupt, sondern sanft sein ([Gonzales, 1996], Seite 33).

Testpersonen fanden animierte Icons sehr hilfreich. Das Erinnern an die Funktionalität eines Objekts fiel dank der Animationen leichter. Allerdings muss jedes Stadium der Animation klar sein und darf keinen Anlass zur Fehlinterpretation geben. Dieser Test zeigte ebenfalls, dass das gleichzeitige Ablaufen vieler Animationen den Bildschirm zu unruhig werden lässt. Dieser Test zeigte auch, dass der Kontext des Benutzers und seine Erwartungen die Interpretation der Animation beeinflussen ([Baecker, Small und Mander, 1991], Seiten 4-5).

### **Räumliche Informationen**

Der Bildschirm an sich ist zunächst einmal ein zweidimensionales Medium. Er besteht aus einem Raster von Bildpunkten. Auch die aus diesen Rasterpunkten zusammengesetzten Ausgaben sind zweidimensional. Heute übliche Bildschirme haben ein Raster zwischen 800x600 Bildpunkten bis 1280x960 Bildpunkten<sup>1</sup>. Der Platz auf dem Bildschirm ist also nicht nur beschränkt, sondern auch knapp. Wird die zweidimensionale Darstellung um eine dritte Dimension ergänzt, kann die zur Verfügung stehende Fläche besser ausgenutzt und eine größere Zahl von Objekten übersichtlich auf dem Bildschirm dargestellt werden. Der Eindruck räumlicher Tiefe auf dem zweidimensionalen Bildschirm kann mit verschiedenen Mitteln erreicht werden:

---

<sup>1</sup> Für Spezialanwendungen gibt es auch größere Bildschirme mit mehr Rasterpunkten.



- Ein Objekt, das ein anderes überlappt, scheint weiter vorne zu liegen<sup>1</sup>.
- Kleinere Objekte scheinen weiter hinten zu liegen.
- Objekte mit weniger gesättigten Farben scheinen weiter hinten zu liegen.

Oft wird bei dreidimensionaler Darstellung auf Computerbildschirmen auch von einer 2 ½ D-Darstellung gesprochen, da die Objekte zwar in der dritten Dimension eine Sortierung haben, aber keine echte Ausdehnung in dieser Dimension besitzen: Es kann festgestellt werden, welches Fenster weiter vorne oder weiter hinten liegt, aber jedes Fenster für sich ist weiterhin zweidimensional ([Herzer und Noll, 1990], Seite 47; [Oberquelle, 1994], Seite 110).

Bei einzelnen Bildern können Schattierungen und Schatten<sup>2</sup>, Texturgradienten<sup>3</sup> und Perspektive, Verdeckung sowie die Größe bekannter Objekte Informationen über räumliche Tiefe geben ([Mallot, 1998], Seite 120).

Die Raum-Metapher bietet sich speziell zur Darstellung interner Systemzusammenhänge an. Funktionale Beziehungen, beispielsweise die logische Struktur abstrakter Grafen, können gut räumlich dargestellt werden. Räumliche Nähe auf dem Bildschirm bedeutet eine inhaltliche Nähe der dargestellten Objekte, funktionale Beziehungen können durch Verbindungen zwischen Objekten dargestellt werden. Für die Darstellung räumlicher Beziehungen auf Bildschirmen gibt es diverse grafische Mittel ([Dutke, 1994], Seite 143; [Oberquelle, 1994], Seite 110).

### **Raum und Bewegung**

Der Benutzer erlebt Raum, egal ob tatsächlichen oder symbolischen Raum, durch die in ihm mögliche Bewegung. Dabei können zwei Arten von Bewegungen unterschieden werden:

- Der Benutzer bewegt sich selbst im Raum (er navigiert beispielsweise durch ein Menüsystem).

---

1 Das Überlappen, also das teilweise Verdecken von Objekten, wurde besonders zu Beginn der grafischen Oberflächen als problematisch angesehen. Der Xerox Star beispielsweise, Urvater aller Computer mit grafischer Bedienoberfläche, hatte keine sich überlappenden Fenster. Jedoch zeigte sich bald, dass auch das überlappte, also nur teilweise sichtbare Objekt noch auf den ersten Blick wahrnehmbar ist. Heute übliche Fenstersysteme erlauben daher, dass sich Fenster gegenseitig überlappen (vgl. [Glaser, 1994], Seite 31).

2 Schattierungen sind Verdunkelungen, die dadurch entstehen, dass Teile des Objekts von der Lichtquelle abgewandt sind. Ein Schatten ist eine Verdunkelung, die aus einer Verdeckung der Lichtquelle durch ein anderes Objekt entsteht ([Mallot, 1998], Seite 145).

3 „Die Tiefenwahrnehmung aus Texturgradienten [...] ist eng mit der perspektivischen Projektion verknüpft, die zu einer Verkleinerung des Bildes mit größerem Abstand und damit zu einer Erhöhung der Dichte der Texturelemente im Bild führt.“ ([Mallot, 1998], Seiten 161-162)

- Der Benutzer bewegt Objekte durch den Raum (er verschiebt eine Datei in ein anderes Verzeichnis).

Funktionale Zusammenhänge, die durch räumliche Relationen dargestellt werden, können durch Bewegungen erkannt werden: Durch das veränderte Aussehen eines Mauszeigers in verschiedenen Bereichen des Bildschirms, das Aktivieren (nach vorne holen) eines anderen Fensters oder das Starten eines neuen Programms, das ein neues Fenster öffnet. Bewegungen als Abbildung von Prozessen vermitteln also Informationen über die Systemzusammenhänge ([Dutke, 1994], Seite 133).

Der Eindruck von Bewegung kann auf drei verschiedene Arten erzeugt werden: Durch die chronologisch korrekte Darstellung eines Prozesses, durch ein statisches Bild und durch animierte Objekte.

Chronologisch korrekte Darstellung bedeutet, dass ein Fenster, das den vorderen Teil eines Dokuments zeigt, zuerst aktualisiert wird und dann das Fenster, das einen weiter hinten liegenden Teil des Dokuments zeigt. Umgekehrt bedeutet es, dass zwei Fenster, die unterschiedliche Perspektiven des selben Gegenstands zeigen, so zeitgleich wie möglich aktualisiert werden sollen. Bei Ursache-Wirkung bedingten Fensteraktualisierungen sollte das Fenster mit der Ursache vor dem mit der Wirkung aktualisiert werden.

In statischen, also nicht animierten Bildern kann Bewegung mit folgenden Mitteln angedeutet werden ([Dutke, 1994], Seiten 134-136):

- Linien können den Bewegungsverlauf zeigen, wobei die Bewegungsrichtung auch Ursache und Wirkung deutlich machen kann. Weiter entfernte Objekte werden dabei kleiner und weniger detailliert dargestellt.
- Mehrfache Andeutungen des Objekts zeigen Schwingungen.

Bei bewegten Bildern können die Bewegungsparallaxe bei Bewegung des Beobachters, dynamische Verdeckung sowie ein kinetischer Tiefeneffekt bei der Bewegung eines Objekts einen Hinweis auf räumliche Tiefe geben ([Mallot, 1998], Seite 120).

## **Grafische Ausgaben**

### **Vorteile einer grafischen Darstellung**

Die grafische Präsentation nutzt die Fähigkeiten des Menschen zur Informationsverarbeitung besser als alle anderen Präsentationsmethoden. Sie kann die Notwendigkeit zur perceptuellen und mentalen Umkodierung sowie die Belastung des Kurzzeitgedächtnisses reduzieren. Auch wenn Visualisierungen nur unter zahlreichen Randbe-

dingungen gedächtnisentlastend wirken, so können sie dennoch die Transparenz des Systems erhöhen und damit das Verständnis von Zusammenhängen fördern. Da bei einer grafischen Präsentation mehrere Werte auf einmal vergleichend gezeigt werden können, wird die Informationsübertragung vom Computer zum Menschen beschleunigt. Grafiken reduzieren die Fehlerrate ebenso wie den Schulungsbedarf und helfen auch beim Problemlösen ([Dutke, 1994], Seite 140, [Galitz, 1988], Seite 219).

Bilder sind verbalen Nachrichten beim Repräsentieren von inhärent räumlichen Konzepten überlegen, beispielsweise Wegweisern, bei zeitlich begrenzter Darstellung und schlechter Sichtbarkeit. Sie können jedoch sehr schlecht abstrakte Konzepte oder Operationen darstellen ([Mullet und Sano, 1995], Seite 200).

### **Empfehlungen**

Zunächst muss abgewogen werden, ob eine grafische oder besser eine textuelle Darstellung, die potenziell detaillierter sein kann, verwendet werden soll. Kriterium hierfür ist der Lernaufwand für das Dekodieren der Darstellung ([Dutke, 1994], Seite 142). Für die Anzeige des freien Plattenplatzes beispielsweise ist eine grafische Anzeige besser als eine numerische, die bis aufs Byte genau informiert ([Cooper, 1995], Seite 139).

Für grafische Darstellungen empfehlen [Mullet und Sano, 1995], möglichst einfache geometrische Formen, vereinfachte Konturen und gedeckte Farben zu verwenden. Werden mehrere ähnliche Formen benötigt, sollten diese soweit wie möglich identisch in Bezug auf Größe, Form, Farbe, Textur, Linienstärke, Ausrichtung und Abstand untereinander sein. Die so entstehende Gleichförmigkeit kann nun genutzt werden, um einzelne Objekte gezielt hervorzuheben. Der Benutzer wird jede Abweichung von der Gleichförmigkeit als bedeutungsvoll interpretieren, auch wenn sie es nicht ist. Eine grafische Darstellung muss also extrem sorgfältig gestaltet werden. Sehr gut zum Erregen der Aufmerksamkeit geeignet sind auch leere Flächen um das Objekt herum.[Mullet und Sano, 1995] sehen räumliche Trennung als die beste Möglichkeit, Aufmerksamkeit auf ein bestimmtes Objekt zu lenken (Seiten 45, 126, 129). [Shneiderman, 1998] (Seiten 81, 386) erwähnt weitere Möglichkeiten, wie durch Hervorhebung die Aufmerksamkeit erregt werden kann und gibt gleich Empfehlungen für die jeweilige Verwendung:

- *Intensität*. Hier sollten nur zwei verschiedene Stufen benutzt werden. Die intensivere Darstellung sollte sparsam und nur zur Hervorhebung benutzt werden.

- *Größe*. Es sollten nicht mehr als vier verschiedene Stufen verwendet werden, wobei die nächste Stufe mindestens 50% größer sein sollte. Für eine Hervorhebung sollte das Objekt vergrößert werden.
- *Inverse Darstellung*.
- *Blinken*. Blinken mit einer Frequenz von 2 Hz bis 5 Hz sollte sehr vorsichtig und nur sparsam eingesetzt werden. Dabei sollte der sichtbare Anteil des Zyklus mindestens 50% betragen.
- *Farben*. Es können bis zu vier Standardfarben verwendet werden. Zusätzliche Farben sollten für gelegentliche Verwendung reserviert bleiben.
- *Farbwechsel (Farbblinken)*. Farbblinken sollte sehr vorsichtig und nur sparsam eingesetzt werden.

[Glaser, 1994] (Seite 40) empfiehlt, die Elemente einer grafischen Darstellung stets gleich anzuordnen, wobei die Anordnung jedoch nicht unbedingt tabellarisch sein muss.

Menschen sind geübt darin, Abweichungen von der Horizontalen zu entdecken. Grafische Darstellungen sollten davon Gebrauch machen ([Tufté, 1984], Seite 186).

### **Icons**

Icons sind für diese Arbeit nicht nur in animierter Form wichtig. Viele einfache Statusanzeigen sowie den Status anzeigende Bedienelemente verwenden ebenfalls Icons für die Darstellung.

Wichtigste Forderung bei der Gestaltung von Icons ist, dass die Klassenzugehörigkeit des Objekts einfach und eindeutig erkannt wird. Icons von Objekten und Operationen müssen also auf der einen Seite einander ähnlich, auf der anderen Seite aber auch unterscheidbar sein ([Herzer und Noll, 1990], Seiten 49-50).

[Mullet und Sano, 1995] listen die häufigsten Fehler beim Entwurf von Icons auf (Seiten 191 ff.):

- *Irreführende Syntax*. Ein Flugzeug wird beispielsweise in der Draufsicht gezeigt, ein Pfeil nach oben oder nach unten soll Start oder Landung bedeuten. Jedoch erkennt der Betrachter auf den ersten Blick eine Seitwärtsbewegung.
- *Schlechte Integration der einzelnen Elemente*. Die Bedeutung jedes Elements muss einzeln analysiert werden, das Icon kann nicht als ganzes interpretiert werden.

- *Untergeordnete Bildelemente dominieren* und lenken vom Kern ab.
- *Verwendung von Bildern für abstrakte Konzepte.* Icons wirken normalerweise aufgrund der direkten Relation, auf der das Bild basiert. Abstrakte Konzepte sind schwerer zu begreifen.
- *Bilder, die auf obskuren Anspielungen beruhen.* Dieser Fehler tritt häufig beim Versuch auf, ein Icon für ein abstraktes Konzept zu entwerfen.
- *Kulturelle oder sprachliche Abhängigkeiten.* Eine waagerechte Sortierichtung beispielsweise wird in Hebräisch oder Arabisch falsch verstanden, da die Lese-richtung umgekehrt ist.
- *Verwendung von Text.* Zum einen sind Texte keine symbolhafte Darstellung, zum anderen nicht von fremdsprachigen Benutzern zu verstehen.
- *Brutale oder schlüpfrige Bilder.*

### **Grafische Darstellung numerischer Daten**

Große Datenmengen können durch gutes Informationsdesign stark komprimiert werden ([Tognazzini, 1996], Seite 116). So kann eine umfangreiche Tabelle numerischer Daten beispielsweise in einer kompakten Grafik dargestellt werden.

Eine gute Grafik fasst viele numerische Werte auf kleinem Raum zusammen, ohne deren Aussage zu verschleiern. Solche Grafiken können die zugrunde liegenden Daten besser visualisieren als eine numerische Darstellung und damit Zusammenhänge deutlich machen. Jedoch ist jede grafische Darstellung nur so gut wie die Qualität der zugrunde liegenden Daten ([Tufté, 1984], Seiten 13, 20).

Bei der grafischen Darstellung numerischer Daten sollten die folgenden Punkte beachtet werden ([Tufté, 1984], Seiten 56, 71-77):

- Der in der optischen Darstellung messbare Unterschied sollte direkt proportional zu dem Unterschied der zugrunde liegenden numerischen Daten sein. Abweichungen hiervon zeigt der

$$\text{Täuschungsfaktor} = \frac{\text{Gezeigter Unterschied in der Grafik}}{\text{Unterschied in den Daten}}$$

Täuschungsfaktoren, die kleiner als 0,95 oder größer als 1.05 sind, weisen auf eine gravierende Verzerrung hin, die über mögliche Ungenauigkeiten beim Zeichnen hinausgeht.

- Die Grafik sollte klar und ausführlich beschriftet sein.

- Wichtig sind die Daten, nicht die Designkünste.
- Die Anzahl der Dimensionen in der Darstellung sollte die Anzahl der Dimensionen in den Daten nicht überschreiten. Zusätzliche Dimensionen verzerren den optischen Eindruck und vermitteln ein falsches Verhältnis der dargestellten Werte untereinander.
- Eine Grafik sollte möglichst viel vom Kontext der Daten zeigen, um Fehlinterpretationen vorzubeugen. Beispielsweise könnte ein ungünstig gewählter Ausschnitt bei periodisch schwankenden Werten dem Betrachter fälschlicherweise einen Anstieg oder einen Rückgang suggerieren.

Fortlassen „überflüssiger Tinte“ legt die eigentliche Information offen und erhöht die Übersichtlichkeit. Als überflüssige Tinte kann zunächst alles angesehen werden, was nicht zur Darstellung der Daten dient: Achsen, Linien durchs Diagramm, Schmuck. Aber auch die Darstellung der Daten kann auf überflüssige Tinte untersucht werden. Ähnlich wie bei Software, so sollte auch der Entwurf von Grafiken ein zyklischer Prozess sein, bei dem ein Zwischenergebnis immer wieder revidiert wird ([Tuft, 1984], Seiten 91 ff.).

### **Textausgabe**

Das Interpretieren alphanumerischer Meldungen dauert länger als das Interpretieren grafischer Darstellungen. [Shneiderman, 1998] zitiert eine firmeninterne Richtlinie für die Dialoggestaltung, nach der Informationen nur dann numerisch präsentiert werden sollen, wenn das Wissen der genauen Zahl notwendig und nützlich ist. Eine ausführliche beschreibende Darstellung verlangsamt den Benutzer in seiner Arbeit, verglichen mit einer strukturierten knapperen Darstellung ([Shneiderman, 1998], Seiten 80-81, 389). Daraus folgt, dass Texte nur ausgegeben werden sollten, wenn es unbedingt notwendig ist, ansonsten sollte auf andere Möglichkeiten zurückgegriffen werden. Ausgegebene Texte sollten so kurz und präzise wie möglich gehalten werden.

Werden Texte ausgegeben, sind zwei grundlegende Aspekte zu beachten: Einerseits der Inhalt, speziell die Formulierung, andererseits die Präsentation auf dem Bildschirm, also Typographie und Layout.

### **Inhaltliche Aspekte**

Textmeldungen sollten leicht verständlich sein. Die Beachtung einiger Regeln erhöht die Verständlichkeit:

- Nur Buchstaben und Zahlen sowie allgemein gebräuchliche Symbole wie #, % oder \$ verwenden.
- Kurze, gebräuchliche Wörter benutzen.
- Vollständige Wörter verwenden, keine Abkürzungen.
- Positiv formulieren: Keine negativen Vorsilben verwenden, statt dessen lieber ein passendes anderes Wort wählen. Keine negierten Aussagen formulieren, also nicht „Soll die Datei nicht gespeichert werden? Ja/Nein“, sondern „Soll die Datei gespeichert werden? Ja/Nein“.
- Vergleiche stets in die größere Richtung anstellen: “Mehr als, später als, schneller als” anstelle von “Weniger als, früher als, langsamer als” ([Galitz, 1988], Seite 74).
- Mitteilungen sollten keine Imperative sein, sondern die Kontrolle durch den Benutzer in den Vordergrund stellen: Die Systemmeldung sollte also nicht ENTER DATA, sondern READY lauten ([Shneiderman, 1998], Seite 376).

### **Formale Aspekte**

Die Größe der Buchstaben auf dem Bildschirm sollte passend zur Arbeitssituation gewählt werden ([Herzer und Noll, 1990], Seite 49). Mehr als zwei oder drei Schriftfamilien in wenigen unterschiedlichen Größen sollten nicht verwendet werden ([Mullet und Sano, 1995], Seite 45; [Shneiderman, 1998], Seite 81). Durch die Wahl von Schriftart und Attributen kann implizit (Status-) Information vermittelt werden. Beispielsweise können GROSSBUCHSTABEN für knappe, ernste Warnungen verwendet werden, wenn ansonsten gemischte Groß-Kleinschreibung verwendet wird ([Shneiderman, 1998], Seite 376). Texte können auch markiert werden, z.B. durch Unterstreichung, Einrahmung oder ein hinweisendes Symbol wie einen Pfeil. Außerdem stehen sämtliche Möglichkeiten der Hervorhebung zur Verfügung, die auch für grafische Ausgaben genutzt werden können (siehe Seite 64).

Damit der Benutzer die Information effizient aufnehmen kann, sollten gebräuchliche Ausrichtungen (linksbündig für Texte, rechtsbündig für Zahlen) und ausreichend Leerraum verwendet werden. Daten werden am besten in Zeilen und Spalten präsentiert ([Shneiderman, 1998], Seite 80).

### 4.3 Akustische Ausgaben

Optische Ausgaben sind in der Interaktion mit dem Computer selbstverständlich, akustische Ausgaben hingegen spielen bislang sehr geringe Rolle. Meiner Ansicht nach liegt ein großes Potenzial für die Mensch-Computer-Interaktion in der Nutzung akustischer Ausgaben, von dem auch Statusanzeigen profitieren können.

Ich werde zunächst einmal Unterschiede und Gemeinsamkeiten akustischer und optischer Ausgaben aufzeigen. Anschließend gehe ich auf die beiden Arten akustischer Ausgaben, nämlich auf nonverbale Nachrichten (Töne und Geräusche) und auf verbale Nachrichten (gesprochene Sprache), ein.

Dieser Abschnitt über akustische Ausgaben basiert im Wesentlichen auf den Arbeiten von [Buxton, 1989] (Seiten 1-8), [Gaver, 1989] (Seiten 67-92) und [Mountford und Gaver, 1991] (Seiten 319-334), die nicht mehr erwähnt werden.

#### Vergleich akustischer mit optischen Ausgaben

Beim Vergleich akustischer und optischer Ausgaben sind zunächst zwei Aspekte zu berücksichtigen: *Wo* kann die Nachricht aufgenommen werden, und *wann* kann sie aufgenommen werden?

Optische Ausgaben existieren an einem einzigen Ort. Sie benötigen daher die fokussierte Aufmerksamkeit des Benutzers. Akustische Ausgaben existieren im gesamten Raum und erreichen den Benutzer auch dann, wenn er seine Aufmerksamkeit gerade anderen Dingen zuwendet. Umgekehrt bedeutet das aber auch, dass der Benutzer optische Ausgaben momentan ignorieren kann, akustische jedoch nicht ([Herzer und Noll, 1990], Seiten 51-52).

Akustische Ausgaben sind vergänglich. Sie existieren an einem einzelnen Zeitpunkt und können daher nur einmal aufgenommen werden, nämlich genau zum Zeitpunkt der Ausgabe. Optische Ausgaben hingegen sind persistent. Sie existieren über längere Zeit und können demzufolge mehrfach aufgenommen werden<sup>1</sup>. Es bietet sich also an, akustische Ausgaben für sich ändernde Ereignisse zu nutzen, optische Ausgaben hingegen eher für stabile Zustände.

---

<sup>1</sup> Eine Ausnahme bilden dynamische optische Ausgaben, die über einen sich in der Zeit ändernden Zustand informieren, beispielsweise Zeigerinstrumente. Diese Ausgaben sind nicht persistent, sondern ebenso vergänglich wie akustische Ausgaben. Jedoch kann eine vergängliche optische Ausgabe ohne große Probleme durch eine persistente(re) Ausgabe ersetzt werden, indem die Historie des zu beobachtenden Werts ausgegeben wird. Es gibt dagegen jedoch keine Möglichkeit, akustische Ausgaben persistent zu gestalten.



Mehrere optische Ausgaben können gleichzeitig präsentiert werden. Der Benutzer kann sich entscheiden, welcher Ausgabe er seine Aufmerksamkeit widmen möchte, er kann auch mehrere Ausgaben gleichzeitig erfassen. Menschen können zwar mehrere akustische Ausgaben gleichzeitig beobachten. Es ist jedoch schwierig und bei größerer Zahl sogar unmöglich, die ungeteilte Aufmerksamkeit genau einem von mehreren simultanen akustischen Signalen zuzuwenden. Anders als bei einem überladenen Bildschirm, der gegebenenfalls durch einen größeren ersetzt werden kann, um mehr Informationen präsentieren zu können oder die Präsentation übersichtlicher zu gestalten, gibt es bei akustischen Ausgaben keine vergleichbaren Möglichkeiten, Ausgaben übersichtlicher bzw. besser hörbar zu präsentieren.

Akustische Ausgaben und optische Ausgaben ergänzen sich oftmals. Die Verwendung von Geräuschen kann viele Informationen vermitteln, ohne den Bildschirm zu überladen. So wie Sprache die Notwendigkeit für manuelle Eingaben verringern kann, so können Geräusche die Notwendigkeit für optische Ausgaben vermindern. Dennoch werden akustische Ausgaben in der Regel nicht als gleichberechtigt zu optischen Ausgaben angesehen: Im Gegensatz zu optischen Ausgaben erreichen akustische Ausgaben auch Unbeteiligte, die sich dadurch gestört fühlen können. Werden akustische Ausgaben mit Rücksicht auf Dritte so gestaltet, dass sie vom Benutzer ausgeschaltet werden können, dürfen sie zwangsläufig nur eine unterstützende Funktion haben.

### **Töne und Geräusche**

Es gibt keine absolute Stille. Wir sind tagtäglich und überall von Geräuschen umgeben, von denen einige für uns nützlich sind, andere nicht. Die nützlichen Geräusche sind Information für uns, die anderen Rauschen. Obwohl wir davon profitieren können, wenn wir die Geräusche um uns herum kontrollieren – wir können den Anteil von Rauschen verringern und dafür das Potenzial zur Informationsvermittlung steigern –, nehmen wir jedoch kaum Einfluss auf sie.

### **Potenzieller Nutzen**

Akustische Signale treten in typischen Alltagssituationen hilfreich in Erscheinung: Autofahrer treffen Entscheidungen „instinktiv“ aufgrund des Motorgeräusches, Fußgänger verlassen sich beim Überqueren der Straße (auch) auf Geräusche, das Geräusch beim Eingießen eines Glases verrät, wie voll das Glas bereits ist. Akustische Signale erfordern nicht die ungeteilte Aufmerksamkeit, so dass mehrere akustische Signale simultan verarbeitet werden können: Menschen können beim Auto fahren nebenbei Musik hören und sich gleichzeitig noch unterhalten. Das Auto fahren wird

davon nicht beeinträchtigt. Im täglichen Leben nutzen Menschen also viele nonverbale Audiosignale, die bisher jedoch kaum bei der Gestaltung der Interaktion mit dem Computer berücksichtigt werden.

Geräusche in der realen Welt können verschiedene Arten von Informationen vermitteln:

- Informationen über physikalische Ereignisse (das heruntergefallene Glas ist zer-sprungen),
- Informationen über nicht sichtbare Strukturen (das Klopfen an der Wand verrät Hohlräume),
- Informationen über dynamische Prozesse (beim Eingießen des Glases hören wir, wann es voll ist),
- Informationen über unnormale Zustände (ein defekter Motor hört sich anders an),
- Informationen über räumliche Ereignisse (am Geräusch erkennen wir, ob sich ein Krankenwagen nähert oder entfernt).

Auch am Computer informieren Töne und Geräusche den Benutzer über den Zustand seines Systems. Fehlen diese gewohnten Geräusche, ruft dieses beim Benutzer Unsicherheit hervor. Beispielsweise kann der Benutzer zweifeln, ob er sein Dokument tatsächlich gespeichert hat, wenn er keine akustische Rückmeldung von der Festplatte bekommt. Moderne PCs werden, nicht zuletzt dank arbeitsrechtlicher Bestimmungen, immer leiser. Die gewohnten akustischen Bestätigungen fallen damit mehr und mehr fort. Trotzdem verlassen die Benutzer sich auf die Geräusche, die übrig bleiben. Auch wenn die Betriebsgeräusche des Computers bisher nicht gestaltet wurden bzw. gar nicht gestaltet werden können, so verursacht ihr Fehlen dennoch Unsicherheit beim Benutzer und senkt damit seine Leistungsfähigkeit. Eine Untersuchung zeigte beispielsweise, dass Spielergebnisse von Computerspielen ohne nonverbale akustische Signale niedriger sind, ein klarer Hinweis darauf, dass diese Signale wichtige Informationen für den Benutzer (in diesem Fall für den Spieler) enthalten. Es ist daher wünschenswert, das durch nonverbale Signale gegebene akustische Feedback zu gestalten, anstatt sich auf zufällige Betriebsgeräusche der Hardware zu stützen ([Herzer und Noll, 1990], Seite 52). Der Benutzer kann außerdem bei gestalteter akustischer Rückmeldung Einfluss auf Parameter wie Klangfarbe oder Lautstärke nehmen und sie so seinen persönlichen Vorlieben oder der Arbeitsumgebung anpassen.

Systeminformation wird traditionell graphisch dargestellt und bleibt auf dem Bildschirm, auch wenn die Information kurz danach nicht mehr benötigt wird. Geräusche

zum Vermitteln von Systeminformationen erhöhen die Bandbreite der Kommunikation. Sie ergänzen die visuelle Ausgabe und können Informationen vermitteln, die besonders bei begrenztem Bildschirmplatz nur schwer zu visualisieren sind. Geräusche sind vergänglich und daher gut geeignet, Veränderungen im Systemstatus zu vermitteln oder einzelne Mitteilungen zu präsentieren, beispielsweise das Eintreffen einer E-Mail. ([Blattner, Sumikawa und Greenberg, 1989], Seite 12).

### **Verwendung nonverbaler Ausgaben**

Funktional betrachtet können nonverbale Audiosignale drei Typen von Informationen übermitteln:

- *Alarm und Warnung.* Diese Signale haben Priorität vor jeder anderen Art von Information. Sie sollen jede laufende Aktivität unterbrechen und die Aufmerksamkeit des Benutzers auf etwas lenken, das sofortige Aufmerksamkeit benötigt. Typischerweise sind diese Audiosignale laut und leicht zu identifizieren. Durch das Erregen von Aufmerksamkeit können sie den Benutzer eventuell in Verlegenheit bringen. Diese Nutzung von Audiosignalen sollte daher immer durch den Benutzer gesteuert werden können ([Shneiderman, 1998], Seite 377).
- *Status und Indikatoren von Prozessen.* Diese Signale liefern Informationen über laufende Prozesse im Vorder- oder Hintergrund. Sie treten nur bei signifikanten Änderungen in den Vordergrund: Der Mensch nimmt gleichbleibende Geräusche nicht bewusst wahr, ist aber sehr sensibel gegenüber Änderungen. Wenn also ein gleichbleibendes Geräusch eines Prozesses plötzlich aufhört, wird das Bewusstsein des Benutzers sofort eine Zustandsänderung wahrnehmen. Ein Mensch kann mehr als eines dieser Signale gleichzeitig im Hintergrund beobachten, vorausgesetzt, die Geräusche sind verschieden genug. Jedoch kann er nur auf höchstens zwei von ihnen zur selben Zeit reagieren. Geräusche können gehört werden, auch wenn das Geräusch erzeugende Objekt nicht sichtbar ist. Aufgrund dieser Analogie eignen sich Geräusche gut für die Ausgabe von Informationen über Hintergrundprozesse. Indikatoren, also kurze Klangereignisse, werden für zufällig eintretende Ereignisse wie beispielsweise das Eintreffen einer E-Mail verwendet, um die eventuell auf andere Aufgaben gerichtete Aufmerksamkeit des Benutzers auf die neue Nachricht lenken ([Herzer und Noll, 1990], Seite 52).
- *Kodierte Nachrichten.* Diese Signale werden genutzt, um numerische Daten (beispielsweise statistische Informationen) als Klangmuster darzustellen. Diese Art von Nachrichten nutzt das menschliche Vermögen für Mustervergleich und Mustererkennung aus.

[Shneiderman, 1998] (Seite 81) trifft die Unterscheidung nicht anhand funktionaler Kriterien. Seine Empfehlung, sanfte Töne für regelmäßiges positives Feedback und harte Töne für seltene Ausnahmefälle zu verwenden, steht jedoch im Einklang mit der oberen Liste.

### **Gestaltung nonverbaler Ausgaben**

Bei Geräuschen in der Umwelt hören Menschen nicht das Geräusch an sich (Lautstärke, Tonhöhe etc.), sondern die Geräuschquelle (wie schwer ist die zugeworfene Autotür, wie stark wurde sie zugeworfen)<sup>1</sup>. Geräusche im Computer sollten also Informationen über die das Geräusch erzeugenden Ereignisse vermitteln. Diese Strategie führt zu akustischen Icons. Akustische Icons sind Alltagsgeräusche wie beispielsweise das Knarren einer Tür oder das Schleifen eines Kastens über den Fußboden, die dem Benutzer über Analogien zu Alltagsereignissen intuitiv Informationen über Computerereignisse vermitteln. Sie erhöhen die Flexibilität bei der Benutzung des Systems, da der Benutzer nicht mehr mit ungeteilter Aufmerksamkeit den Schirm betrachten muss, sondern bewusst oder auch unbewusst auf die zur Aktion gehörenden Geräusche achten kann.

Akustische Icons werden oft auch Earcons genannt. Wie Icons, so können auch Earcons in repräsentative und abstrakte Earcons unterteilt werden. Repräsentative Earcons entsprechen natürlichen Geräuschen. Ihre Zuordnung zu Ereignissen ist entweder symbolisch oder metaphorisch. Symbolische Earcons beruhen auf Konventionen (Applaus für Zustimmung), metaphorische Earcons entweder auf physikalischen Gleichheiten (das Schließen einer Metalltür für das Schließen einer Datei) oder auf Ähnlichkeiten (fallende Tonhöhe für ein fallendes Objekt). Abstrakte Earcons werden systematisch aus Motiven zusammengesetzt. Diese Motive haben meist einfache Tonhöhen und Rhythmen. Jedes Earcon sollte lang genug sein, um die Aufnahme der transportierten Nachricht zu garantieren, aber nicht länger. Bei abstrakten Earcons beispielsweise sind zwei bis vier Noten angemessen, optimal sind drei Noten. Längere Motive werden als Musik wahrgenommen und vom Benutzer nach kurzer Zeit ausgeblendet ([Blattner, Sumikawa und Greenberg, 1989], Seiten 13, 21-22, 27-28).

Repräsentative Earcons sind besser zu erlernen als abstrakte, da der Benutzer sie von Kindheit an kennt und mit bestimmten Ereignissen verbindet. Sie sind gut geeignet für Systeme, die wenig Earcons benötigen. Kombinationen abstrakter Earcons sind schwer zu erlernen, jedoch kann eine Vielzahl recht komplexer Mitteilungen durch

---

<sup>1</sup> Die von Geräuschen transportierte Information ist also mehrdimensional, d.h. ein einzelnes Geräusch kann mehrere Attribute seiner Quelle übertragen ([Gaver, 1989], Seite 75).

das Aneinanderreihen der entsprechenden Motive erstellt werden<sup>1</sup>. Werden allerdings zu viele verschiedene Earcons verwendet, fallen Unterscheidung und Zuordnung schwer ([Blattner, Sumikawa und Greenberg, 1989], Seiten 34, 40).

Die Zuordnung von „komponierten“ Tönen zu Computerereignissen ist eher zufällig. Akustische Icons für Objekte und Aktionen im Computer sollten daher zunächst einmal Geräuschen der entsprechenden Objekte und Aktionen aus der realen Welt entsprechen. Ist dieses nicht möglich, weil es kein Gegenstück in der realen Welt gibt, dann sollte das Geräusch Eigenschaften verwandter Objekte oder Aktionen haben. Verursacht die Aktion oder das Objekt in der realen Welt kein Geräusch, sollten die für den Benutzer wichtigsten Eigenschaften durch ein Geräusch aus der natürlichen Welt dargestellt werden. Ein Beispiel hierfür ist das Geräusch vom Eingießen in ein Glas als akustische Repräsentation des Kopierens. Die für den Benutzer wichtige Eigenschaft der Aktion im Computer ist der Zeitpunkt, an dem die Kopie fertig ist. Der Fortschritt beim Kopieren wird durch das immer voller werdende Glas akustisch repräsentiert. Ist es voll, ist die Kopie fertig. Systemattribute sollten durch Parameter der Geräusche wie Nachhall dargestellt werden. Da die Geräusche auf Analogien zu Alltagsgeräuschen basieren, werden sie nur schwer wieder vergessen. Wird die selbe Analogie für optische und akustische Icons verwendet, steigert die so erzeugte Redundanz den Lerneffekt. Akustische und optische Icons können sich so ergänzen: Ereignisse im Computer können oft mit der einen Art Icon besser dargestellt werden als mit der anderen. Akustische Signale sollten so vielfältig wie möglich sein, da Wiederholungen langweilig werden. Entweder sollten sie zufällig variiert werden oder sehr viele Informationsdimensionen enthalten, so dass identische Signale sehr selten vorkommen. Gut designte Audio-Interfaces werden also eine natürliche Erweiterung der in jedem Büro bereits existierenden, aber nicht bewusst wahrgenommenen Umgebungsgeräusche sein. Die verwendeten Geräusche sollten unterscheidbar sein, aber sie sollten vom Benutzer nicht bewusst bemerkt werden oder ihn gar stören.

### Beispiele

Ein Beispiel für die Integration akustischer Signale in die Mensch-Computer-Schnittstelle ist der SonicFinder, eine experimentelle Entwicklung von Apple. Der SonicFinder ist eine Erweiterung des Finders, des damaligen grafischen Betriebssystems des Apple Macintosh, mit Unterstützung durch akustische Ausgaben.

---

<sup>1</sup> *Textdokument öffnen* beispielsweise wird durch das Earcon für *Textdokument*, gefolgt von dem Earcon für *Öffnen*, begleitet und hört sich anders an als *Textdokument schließen* oder *Bilddatei öffnen*.

Der SonicFinder gibt Benutzern zusätzlich zu visuellem Feedback ein Audio-Feedback. Dieses Feedback besteht aus einem sogenannten akustischem Icon, einem Geräusch, das eine intuitiv zu erfassende Analogie zwischen Ereignissen im Computer und der realen Welt herstellt – beispielsweise ein Kratzgeräusch, das das Verschieben eines Objekts auf dem Desktop begleitet. Dabei werden Parameter des Geräuschs verändert, um dadurch zusätzliche Informationen zu vermitteln, die das optische Interface nicht oder nur eingeschränkt vermittelt. Beispielsweise werden Dateitypen durch Materialien repräsentiert und die Größe des Objekts durch die Frequenz. Jedes Objekt hat also ein charakteristisches eigenes Geräusch, so dass das Verschieben eines großen Textes von einem anderen Geräusch begleitet wird als das eines kleinen Textes oder einer Programmdatei. Durch die Vielzahl der möglichen Geräusche werden einzelne Geräusche nicht bewusst und erst recht nicht störend wahrgenommen. Diese Geräusche werden akustische Icons genannt. Sie vermitteln Informationen, die das optische Interface nicht oder nur eingeschränkt vermittelt.

Umfragen unter Testnutzern zeigten, dass die Arbeit mit dem SonicFinder angenehmer und effektiver als mit dem normalen Betriebssystem empfunden wurde. Nutzer, die zum Teil seit mehr als zwei Jahren mit SonicFinder arbeiteten, beschwerten sich sogar, wenn die akustischen Informationen, an die sie sich gewöhnt hatten, fehlten. Kritisch wurden laute Geräusche beurteilt, die simultan aus mehreren Arbeitsplätzen kamen. Allerdings empfehlen die Entwickler des SonicFinders eine sehr niedrige Lautstärke, um ein natürliches und subtiles Arbeiten zu ermöglichen. Bei „ordnungsgemäßer“ Lautstärke dürfte diese Kritik daher nicht auftreten. Der Benutzer sollte aber auf jeden Fall die Kontrolle sowohl über die Klänge als auch über die Lautstärke haben.

Von seinen Machern wird der SonicFinder als Demonstration angesehen, dass

- Geräusche so in das Interface integriert werden können, dass sie als natürlich empfunden werden,
- Geräusche Informationen vermitteln können, die grafisch nicht oder nur deutlich schlechter dargestellt werden können,
- die akustischen Icons im SonicFinder selten wichtige Informationen vermitteln, die nicht bereits optisch vermittelt wurden,
- Geräusche also redundante Informationen geben können, die potenziell nützlich sind.

Ein weiteres Beispiel ist Audio Aura, vorgestellt in [Mynatt, Back, Want, Baer und Ellis, 1998]. Hier werden unauffällige akustische Signale im Hintergrund in einer

VRML-Umgebung eingesetzt, die mit physikalischen Aktivitäten am Arbeitsplatz verknüpft sind. Testpersonen empfanden das Konzept von Audio Aura als positiv, auch wenn an einzelnen Aspekten Kritik geäußert wurde.

### **Eignung für Statusanzeigen**

Nonverbale Audiosignale werden bisher überwiegend als Fehlermeldung eingesetzt und haben dadurch ein schlechtes Image bei den Benutzern erhalten. Auch die Spielereien von Windows, wo viel zu laute einzelne Geräusche viel zu viele Aktionen überflüssigerweise akustisch begleiten, haben den Einsatz nonverbaler Audiosignale in Verruf gebracht. Die meisten Windows-Nutzer versuchen zunächst, die Lautstärke zu regulieren. Diese kann aber nur in Abhängigkeit von der technischen Form des Audiosignals geregelt werden, die Lautstärke von Systemmeldungen ist nicht einzeln regelbar. Wer die Lautstärke wegen störender Systemmeldungen herunterregelt, hat damit gleichzeitig die akustischen Ausgaben von beispielsweise Spielen leiser gestellt. Daher stellen viele Benutzer die Windows-Audiosignale nach kürzester Zeit ab.

Ich finde es bedauerlich, dass nonverbale Audiosignale eine so unbedeutende Rolle in der Interaktion von Mensch und Computer spielen. Meiner Ansicht nach zeigt das Beispiel des SonicFinders, dass hier ein großes Potenzial brachliegt. Eigene Erfahrungen mit einem leiseren PC bestätigten mir, dass ich mich unbewusst zu einem nicht unerheblichen Teil auf die Betriebsgeräusche stütze.

Nonverbale akustische Signale sollten eingesetzt werden, um positives Feedback zu liefern – beispielsweise, um Vorgänge auf dem Bildschirm zu begleiten, das ordnungsgemäße Funktionieren des Systems anzuzeigen oder Aktionen wie das Schreiben oder Lesen einer Datei zu bestätigen. Wichtig ist hierbei, dass die Lautstärke dieser Signale separat, also ohne Nebenwirkungen auf akustische Ausgaben anderer Programme, geregelt werden kann.

## **Sprache**

### **Potenzielle Vorteile und Risiken**

Mit Hilfe von Sprachausgabe können beliebige Texte akustisch präsentiert werden, um

- besondere Aufmerksamkeit zu erregen,
- Sehbehinderte zu unterstützen,
- Erläuterungen zu geben, wenn die Augen anderweitig beschäftigt sind oder

- dem Benutzer eine komplexe Nachricht übermittelt werden soll, ohne den Inhalt des Bildschirms zu ändern.

Nützlich ist dies insbesondere bei Online-Tutorials, wenn Ereignisse auf dem Bildschirm mit Sprachausgabe erläutert werden ([Oberquelle, 1994], Seite 114; [Herzer und Noll, 1990], Seite 54). Ein weiterer Vorteil gesprochener Nachrichten ist, dass sie leichter aufgenommen werden können als visuell präsentierte Texte ([Blattner, Sumikawa und Greenberg, 1989], Seite 14).

Sprachausgabe kann leicht unterschieden und beispielsweise verschiedenen Prozessen zugeordnet werden, wenn verschiedene Sprecher gewählt werden. Denkbar ist auch, verschiedene Arten von Meldungen wie beispielsweise Fehlermeldungen oder Statusmeldungen von verschiedenen Sprechern ausgeben zu lassen. Die Bedeutung einer gesprochenen Meldung kann durch Parameter wie Lautstärke und Betonung weiter variiert werden, beispielsweise um dringende Meldungen besonders zu betonen ([Oberquelle, 1994], Seite 114; [Herzer und Noll, 1990], Seite 54).

Zu beachten ist aber, dass Menschen von einem „sprechenden“ System ein höheres Maß an Intelligenz und Verständnis erwarten. Daher sollte Sprache nur verwendet werden, wenn sichergestellt ist, dass die Meldungen verbindlich und genau sind ([Tognazzini, 1996], Seite 275). [Shneiderman, 1998] spricht sich generell gegen anthropomorphes Design aus, das seiner Meinung nach zu Verwirrung, Täuschung und Fehlleitung beim Benutzer führt, der beim Versuch, menschliche Gefühle in das Feedback hinein zu interpretieren, vom Wesentlichen abgelenkt wird. Feedback ist umso effektiver, je schlichter es ist ([Shneiderman, 1998], Seiten 380 ff.).

### **Eignung für Statusanzeigen**

Sprache spielt bei Statusinformationen bisher keine sonderlich große Rolle. Lediglich einige wenige Systeme wie AOL informieren mit einer gesprochenen Meldung über den Status („Sie haben Post“), teilweise ist Sprache auch bei Spielen zu finden. Eigene, nicht repräsentative Beobachtungen speziell von Benutzern dieses Systems zeigen, dass Sprache von Benutzern mit mehr Computererfahrung tendenziell eher als störend empfunden wird. Computerneulinge und Gelegenheitsnutzer hingegen stören sich nicht an der Sprache, sondern empfinden die gesprochene Meldung zum Teil sogar als angenehm.

Versuche einiger Automobilhersteller um 1990, Statusinformationen mittels gesprochenen Hinweisen zu übermitteln, wurden am Markt nicht akzeptiert. Psychologische Gründe waren mit ausschlaggebend für das Scheitern: Die Autofahrer wollten sich



ihre Fehler nicht ständig unter die Nase reiben lassen oder schlimmer noch, sich vor anderen bloßstellen.

Meiner Ansicht nach ist die Einsatzmöglichkeit von Sprache für die Übermittlung von Statusinformationen begrenzt. Sie können an einzelnen Stellen nonverbale akustische Ausgaben, die optische Ausgaben unterstützen, ersetzen. Der Vorteil ist, dass die Bedeutung eines gesprochenen Textes anders als bei einem nonverbalen Signal nicht erst erinnert werden muss („Ist das 'Pling' nun ein Zeichen für neue Mail, oder bedeutet es, die Kapazität der Festplatte ist erschöpft?“). Die Ausgabe komplexerer Statusinformationen ausschließlich per Sprache ist zwar denkbar für Aufgaben, bei denen die Aufmerksamkeit nicht auf einen Bildschirm gerichtet sein kann. Aufgrund der Flüchtigkeit akustischer Ausgaben und der geringen Speicherkapazität für akustische Nachrichten ist hiervon jedoch Abstand zu nehmen – die ersten übermittelten Statusinformationen werden von den später eintreffenden überschrieben und damit vergessen.

#### **4.4 Äußere Einflüsse auf die Anzeige**

Neben den physiologischen und psychologischen Eigenschaften des Menschen gibt es weitere Faktoren, die Einfluss auf eine Anzeige haben. Diese Faktoren sind nicht allgemein gültiger Natur, sondern können nur für ein konkretes Programm betrachtet werden. Einsatzort und -zweck eines Programms, aber auch seine Zielgruppe sind Faktoren, die bei der Gestaltung seiner Anzeigen berücksichtigt werden müssen.

##### **Der Einfluss der Systemumgebung**

Zwei Sachverhalte sind hierbei wichtig ([Herzer und Noll, 1990], Seite 102):

- Ein Anwendungsprogramm läuft immer auf einem konkreten Rechner in einer konkreten Systemumgebung ab.
- Ein Anwendungsprogramm ist Teil einer gesamten, meist betrieblichen EDV-Umgebung. Die Oberfläche des Gesamtsystems setzt sich hier aus den Oberflächen der einzelnen Anwendungsprogramme zusammen.

Heutige Programme laufen meist auf einem grafischen Betriebssystem ab. Dieses Betriebssystem liefert bereits die wesentlichen Elemente für das Design der Benutzungsoberfläche. Das vereinfacht auf der einen Seite die Erstellung eines neuen Programms und sorgt für konsistente Programme, auf der anderen Seite wird der Systementwickler hierdurch auch in seinen gestalterischen Möglichkeiten eingeschränkt. Fehlende Elemente zur Oberflächengestaltung beispielsweise müssen im Look&Feel zu den bereits existierenden, vom Betriebssystem bereitgestellten Elementen passen.

## Der Einfluss des Programms

Es gibt die verschiedensten Arten von Programmen, die auf einem Computer laufen. [Cooper, 1995] teilt die Programme anhand ihrer Präsentation dem Benutzer gegenüber in vier Gruppen ein und zeigt, dass die Gestaltung der Bedienoberfläche keine rein ästhetische Entscheidung ist, sondern primär das Verhalten des Programms gegenüber dem Benutzer berücksichtigt werden muss ([Cooper, 1995], Seiten 152 ff.):

- Ein souveränes Programm ist als einziges auf dem Bildschirm, es beansprucht daher die ungeteilte Aufmerksamkeit des Benutzers. Die Benutzer dieser Programme sind normalerweise erfahren. Da das Programm nicht nur als einziges, sondern dazu auch noch längere Zeit auf dem Bildschirm ist, sollten die verwendeten Farben zurückhaltend sein, um den Benutzer nicht nach kurzer Zeit zu nerven. Souveräne Programme enthalten viel visuelles Feedback wie Graphen, Zahlen, Indikatoren, virtuelle LEDs und anderes, z.B. in einer Statuszeile, am Ende von Scrollbars oder in der Titelzeile. Viele subtile Elemente der Benutzungsoberfläche erschließen sich erst dem erfahrenen Benutzer, der Neuling wird sie nicht unbedingt bemerken.
- Flüchtige Programme manipulieren ein Dokument, haben aber nur eine sehr spezielle Funktion wie beispielsweise das Einscannen eines Bildes. Ein flüchtiges Programm wird aufgerufen, wenn es gebraucht wird. Es erscheint, präsentiert eine einzelne Funktion mit einem begrenzten Satz Bedienelemente, erledigt seinen Job, verschwindet wieder und lässt den Benutzer anschließend seine normalen Aktivitäten tun. Da die Benutzer nicht oft mit diesem Programm arbeiten, ist jeder für dieses Programm ein Neuling oder Gelegenheitsnutzer. Die Gestaltung der Oberfläche darf daher nicht subtil sein, da die Feinheiten sich während der kurzen, gelegentlichen Nutzung nicht erschließen. Die Bedienelemente sollten außerdem größer als bei einem souveränen Programm sein. Ein flüchtiges Programm sollte sich nicht in langweiligem Grau präsentieren, aber auch nicht übermäßig bunt sein. Animationen können an der Bedienoberfläche verwendet werden.
- Dämonische Programme interagieren normalerweise nicht mit dem Benutzer, sie verrichten ihre Arbeit unauffällig im Hintergrund. Meistens handelt es sich um wichtige Systemaufgaben, für die keine Benutzerinteraktion erforderlich ist, beispielsweise ein Druckertreiber. Die Interaktion mit einem dämonischen Programm ist extrem selten, oft erfolgt sie nur bei der Installation. Wie flüchtige Programme, so haben auch dämonische Programme nur eine sehr spezielle Funktion und sind nicht permanent auf dem Bildschirm. Daher gilt für die Gestaltung der Bedienoberfläche eines dämonischen Programms das gleiche wie für flüchtige Pro-

gramme. Es muss jedoch noch mehr Sorgfalt darauf gelegt werden, dem Benutzer alle wichtigen Informationen für die Interaktion zu liefern, da dämonische Programme für das System lebenswichtige Funktionen kontrollieren.

- Parasitäre Programme haben sowohl von souveränen Programmen als auch von flüchtigen Programmen Eigenschaften. Diese Programme sind wie ein souveränes Programm zwar ständig präsent, erfüllen aber wie ein flüchtiges Programm nur eine unterstützende Funktion. Parasitäre Programme sind typischerweise unauffällige Reporter für einen laufenden Prozess, wobei die Berichterstattung manchmal eine Funktion ist, die sie zusätzlich zum Verwalten des Prozesses ausführen. Ein Programm, das ständig die Nutzung der Systemressourcen durch ein Chart anzeigt, aber auch eine Uhr oder ein Programm, das lediglich den schnellen Zugriff auf andere ermöglicht, sind parasitäre Programme.

Für das notwendige Feedback innerhalb eines Programms wird von [Shneiderman, 1998] (Seiten 74-75) eine Richtlinie gegeben. Er meint, dass grundsätzlich jede Aktion des Benutzers durch Feedback quittiert werden sollte. Dabei brauchen häufig ausgeführte und kleinere Aktionen nur ein mäßiges Feedback geben, während unregelmäßige und größere Aktionen substanzielleres Feedback geben sollten.

### **Der Einfluss des Benutzers**

Statusanzeigen werden für den Menschen gemacht. Nun sind aber keine zwei Menschen gleich, und auch ein und der selbe Mensch kann zu zwei verschiedenen Zeitpunkten unterschiedlich leistungsfähig sein. Beim Entwurf von Statusanzeigen muss daher sowohl das menschliche Leistungsspektrum bezüglich Wahrnehmung und Informationsverarbeitung berücksichtigt werden als auch auf die unterschiedlichen Bedürfnisse verschiedener Personengruppen Rücksicht genommen werden. Die Einteilung der Personengruppen kann dabei nach den unterschiedlichsten Aspekten erfolgen, beispielsweise nach Alter, Charakter oder Erfahrung mit dem System.

[Shneiderman, 1998] listet folgende Faktoren, die die Leistungsfähigkeit des Benutzers beeinflussen: Erregung und Aufmerksamkeit, Müdigkeit, momentane geistige Anstrengung, Kenntnis der Ergebnisse, Monotonie und Langeweile, Verlust sensorischer Fähigkeiten, Isolation, Alterung, Drogen und Alkohol. Er schreibt ihnen einen tief greifenden Einfluss auf den Entwurf der meisten interaktiven Systeme zu ([Shneiderman, 1998], Seiten 20-21).

Berücksichtigt man das Alter, stellt man schnell fest, dass Kinder ein anderes Feedback als Erwachsene benötigen<sup>1</sup> ([Shneiderman, 1998], Seiten 383-384). Während bei Kindern jedoch auf ihre kindliche Psyche Rücksicht genommen werden muss, sollte bei älteren Benutzern die nachlassende Leistungsfähigkeit beachtet werden. Anzeigen von Systemen für ältere Menschen sollten groß und kontrastreich sein ([Shneiderman, 1998], Seite 27).

Berücksichtigt man den Charakter, gibt es zwei unterschiedliche Aspekte zu berücksichtigen. Zum einen gibt es den Unterschied zwischen extrovertierten und introvertierten Menschen, zum anderen den zwischen Menschen, die sich auf ihr Gefühl verlassen und denen, die nur harten Fakten vertrauen. Die Extreme der beiden Paarungen haben jeweils andere Anforderungen an das Feedback, was bei der Gestaltung von Statusanzeigen zu berücksichtigen ist. Introvertierte Menschen beispielsweise möchten eher keine Aufmerksamkeit erregen und sind daher weniger bereit, an einem System mit akustischen Ausgaben zu arbeiten ([Tognazzini, 1992], Seiten 93 ff.). Für einen Menschen, der sich auf sein Gefühl verlässt, mag eine Grafik ausreichend sein, während der eher faktenorientierte Mensch eine numerische Darstellung vorzieht.

Rechner wurden kleiner, leistungsfähiger und preiswerter. Damit kommen immer mehr Menschen ohne Fachkenntnisse von Rechnern mit ihnen in direkten Kontakt, die sogenannten *Computerlaien*. Auf sie muss bei der Gestaltung der Systemoberfläche und damit auch von Statusanzeigen Rücksicht genommen werden. Ebenso spielen immer komplexere Anwendungen eine Rolle bei der Gestaltung der Systemoberfläche ([Herzer und Noll, 1990], Seiten 4-5). Berücksichtigt man die Erfahrung am System, so ist zu erkennen, dass Anfänger oder Gelegenheitsnutzer umfangreicheres und informativeres Feedback benötigen als Experten, die oft mit bescheidenem Feedback zufrieden sind ([Galitz, 1988], Seite 38). Speziell bei Systeminformationen kehrt sich das Bild aber um: Hier wünschen Experten mehr Informationen als Anfänger oder Gelegenheitsnutzer – letztere wären mit den Informationen überfordert, da sie sie nicht interpretieren können. Die Folgerung hieraus ist, dass den Experten auf Nachfrage ausführlichere Informationen zur Verfügung gestellt werden sollten, die im Normalfall nicht sichtbar sind.

## 4.5 Bedienelemente

Eine Fläche auf dem Bildschirm kann zum einen Inhalte darstellen, zum anderen kann auch auf sie gezeigt werden. Diese Kombination ermöglicht es, Bedienelemente

---

<sup>1</sup> Microsoft Bob, eine Alternative zu Windows, bei der eine Comic-Figur in einem gezeichneten Wohnzimmer die Programmoberfläche bildete, ist nicht zuletzt deshalb am Markt gescheitert, weil die Erwachsenen als Zielgruppe angesprochen wurden, das Feedback aber eher kindgerecht war.

auf dem Bildschirm realistisch zu gestalten ([Oberquelle, 1994], Seiten 111-112). Die meisten Bedienelemente auf dem Bildschirm zeigen auch die momentane Einstellung. Daher sieht beispielsweise [Cooper, 1995] (Seite 349) eine gedrückte Schaltfläche in einer Werkzeugleiste als eine Art Statusanzeige. Bedienelemente, die gleichzeitig einen Status anzeigen, erfüllen die von [Mullet und Sano, 1995] aufgestellte Forderung, dass Elemente für eine maximale Wirkung kombiniert werden sollen ([Mullet und Sano, 1995], Seite 37).

Virtuelle Bedienelemente sind naturgemäß optischer Natur. Damit gelten natürlich zunächst einmal alle in Kapitel 4.2, *Optische Ausgaben*, gemachten Aussagen. Nun ist bei einem Bedienelement jedoch nicht die Darstellung von Informationen das primäre Ziel, sondern die leichte Bedienbarkeit. Die Vermittlung von Statusinformationen ist bei Bedienelementen eher ein Abfallprodukt. Trotzdem sollte bei der Gestaltung von Bedienelementen nicht nur

die Bedienbarkeit, sondern auch die Darstellung des eingestellten Werts berücksichtigt werden. [Shneiderman, 1998] zeigt Designstudien für einen Ein-Aus-Schalter auf dem Bildschirm (Abbildung 18). Diese Studien können sowohl unter dem Aspekt der Bedienbarkeit als auch unter dem Aspekt der Erkennbarkeit des momentanen Status betrachtet werden. Problematisch bei allen sechs Varianten ist in meinen Augen die Beschriftung: Ein gutes Bedienelement sollte sowohl seine Funktionsweise als auch den Status der zugehörigen Funktion ohne Aufschriften vermitteln können.

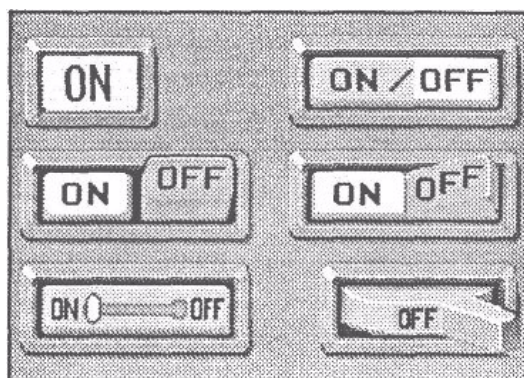


Abbildung 18: Verschiedene Visualisierungen eines Ein-Aus-Schalters (Design von Catherine Plaisant, in [Shneiderman, 1998], Seite 216)

Problematisch bei allen sechs Varianten ist in meinen Augen die Beschriftung: Ein gutes Bedienelement sollte sowohl seine Funktionsweise als auch den Status der zugehörigen Funktion ohne Aufschriften vermitteln können.

[Tognazzini, 1992] vertritt die Auffassung, dass komplexere Statusanzeigen möglichst separat von Bedienelementen gehalten werden sollten. Zwar soll das Bedienelement den momentanen Status der zugehörigen Funktion anzeigen. Darüber hinausgehende Statusinformationen sollten jedoch getrennt angezeigt werden. Er empfiehlt, einzelnen Objekten nicht zu viele oder mehrdeutige Bedeutungen zuzuordnen. Schalter und eine komplexere Statusanzeige sollten nicht in einem einzigen Kontrollelement kombiniert werden, es sollten zwei verschiedene Elemente genommen werden ([Tognazzini, 1992], Seiten 214-215).

Der Zustand der zugehörigen Programmfunktion ist jedoch nicht die einzige Information, die ein Bedienelement dem Benutzer geben kann. Eingabefelder beispielsweise können die Gültigkeit der Eingabe anzeigen, zum Beispiel durch simulierte

LEDs oder durch Änderung der Farbe. Das Eingabefeld kann auch anzeigen, ob eine Wertekontrolle stattfindet, beispielsweise durch einen andersfarbigen Rand ([Cooper, 1995], Seiten 398 ff., 457).

Insbesondere bei Funktionen, die Auswirkungen auf das Aussehen des Dokumentes haben, können von [Cooper, 1995] so genannte *visual gizmos* verwendet werden ([Cooper, 1995], Seiten 416 ff.). Diese Bedienelemente sind primär eine Vorschau auf das spätere Aussehen des Dokuments, mit der das Aussehen beeinflusst werden kann. Abbildung 19 zeigt im unteren linken Bereich ein solches Bedienelement. Die

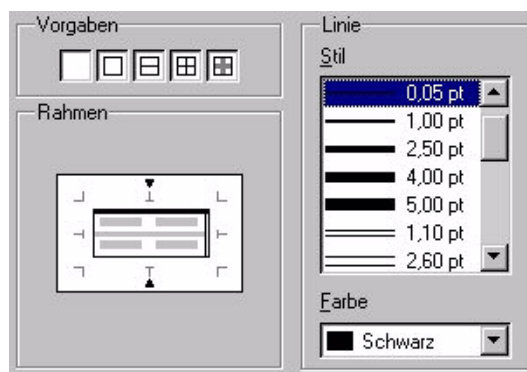


Abbildung 19: Einstellen der Umrandung von Tabellenzellen beim StarOffice Calc

Dreiecke zeigen an, welche der sechs Linien momentan gesetzt werden<sup>1</sup>. Die Linien zeigen den gewählten Stil und die gewählte Farbe, wobei die dicke graue Linie bedeutet, dass die momentane Einstellung nicht geändert wird. Meiner Meinung nach kann man hieran gut erkennen, dass die Vermittlung eines komplexeren Status die Bedienbarkeit beeinträchtigen kann: Im Gegensatz zu einer Schaltfläche ist die Bedienung hier nicht sonderlich intuitiv.

## 4.6 Visuelles Feedback außerhalb des Bildschirms

Ein Beispiel, wie und welche Informationen visuell außerhalb des Bildschirms vermittelt werden können, ist der ambientROOM ([Ishii, Wisneski, Brave, Dahley und Gorbet, 1998], Video). Mit Hilfe des ambientROOM wurde untersucht, wie mit Hilfe von Umwelteindrücken Informationen vermittelt werden können. Für diese Studie wurde ein spezieller Raum geschaffen, eben der ambientROOM.

Das Video beginnt mit der Feststellung, dass Menschen Informationen auf zwei verschiedenen Wegen erhalten:

- Sie widmen ihre Aufmerksamkeit einer Sache und erhalten Informationen hierüber.
- Sie nehmen Umwelteindrücke wie Licht, Geräusche, Temperatur oder Luftbewegungen wahr. Diese Eindrücke informieren sie über beispielsweise Wetter, Tageszeit oder die Anzahl anwesender Personen.

<sup>1</sup> Im Beispiel wird nur eine Linie gesetzt. Es ist aber auch das Setzen von mehreren Linien gleichzeitig möglich, indem eine der Optionen unter *Vorgaben* gewählt wird.

Die übliche Interaktion des Menschen mit dem Computer ist auf Methoden beschränkt, die seine Aufmerksamkeit erfordern. Mit dem ambientROOM jedoch können Umwelteindrücke genutzt werden, um Informationen zu vermitteln. Damit werden die kognitiven Fähigkeiten des Menschen voll ausgenutzt: Er widmet einer Sache seine Aufmerksamkeit, verarbeitet aber auch unbewusst Informationen. Umwelteindrücke eignen sich gut zur Beobachtung kontinuierlicher Prozesse. Folgende Beispiele zeigte das Video:

- Kleine Kräuselwellen, die an die Decke projiziert werden, zeigen die Aktivität eines Hamsters.
- Unregelmäßige Lichtflecke, die an die Seitenwand projiziert werden, zeigen den Publikumsverkehr im Eingangsbereich des Gebäudes.
- Verkehrslärm gibt Auskunft über den Verkehr in Netzwerken. Der Benutzer kann die Geräusche ein- und ausschalten, wenn er die zugehörige Information benötigt, oder falls die Geräusche ihn stören.
- Licht bewegt sich im Raum, analog zur Tageszeit.
- Das Geräusch von Regen oder Gewitter informiert über eintreffende Mails (wenig oder viel).

Das Vermitteln von Informationen über Umwelteindrücke nutzt die Fähigkeit des Gehirns zur Parallelverarbeitung und seine Fähigkeit zum Managen von Aufmerksamkeit.

## 5 Statusanzeigen in Literatur und Praxis

Statusanzeigen werden in der wissenschaftlichen Literatur nur selten behandelt. Auch kommerzielle Style Guides gehen erst in jüngster Zeit verstärkt auf Statusanzeigen ein. Trotzdem gibt es die verschiedensten Formen von Statusanzeigen in Software. Ich zeige zunächst verschiedene Beispiele, die ich anschließend in einem Überblick zusammenfasse.

Dieses Kapitel enthält noch keine Kritik an den gefundenen Beispielen aus der Praxis, es soll lediglich einen Überblick über die Verwendung von Statusanzeigen in heutigen Programmen geben. Die Kritik an den gefundenen Statusanzeigen erfolgt in Kapitel 6, *Kritische Betrachtung existierender Statusanzeigen*.

### 5.1 Statusanzeigen in der wissenschaftlichen Literatur

In der wissenschaftlichen Literatur werden überwiegend zwei Aspekte von Statusanzeigen untersucht: Fortschrittsindikatoren und Modusanzeigen durch geänderte Mauszeigerformen. Einige wenige Arbeiten haben sich mit der Möglichkeit zur Integration akustischer Signale befasst<sup>1</sup>, wobei jedoch der Schwerpunkt nicht auf Vermittlung von Statusinformationen lag.

#### Dynamische Zustände

Fortschrittsindikatoren werden in längeren Zeitabständen immer wieder untersucht. Ich führe exemplarisch drei Arbeiten an, die die wissenschaftliche Entwicklung in diesem Bereich zeigen. Fortschrittsindikatoren sind nach [Myers, 1985] seit etwa 1976 bekannt. In seiner Arbeit untersucht er, ob sie dem Benutzer einen Vorteil bringen. [Musil, 1993] untersucht die optische Gestaltung von Fortschrittsindikatoren: Wie müssen sie beschaffen sein, damit der Benutzer von ihnen profitieren kann? [Cooper, 1995] schließlich führt in seinem Standardwerk über Software-Ergonomie einige Punkte bezüglich Platzierung von Fortschrittsindikatoren sowie den notwendigen Eingriffsmöglichkeiten – insbesondere die Möglichkeit zum Abbruch des Prozesses – an. In den Jahren nach [Musil, 1993] sind Fortschrittsindikatoren nicht der Mittelpunkt wissenschaftlicher Arbeit gewesen, jedenfalls ist in den Tagungsbänden wichtiger Konferenzen wie beispielsweise der CHI nichts mehr zu diesem Thema zu lesen.

---

<sup>1</sup> Siehe hierzu den *SonicFinder* von Apple in Kapitel 4.3, ab Seite 74.



**Myers, 1985**

[Myers, 1985] beklagte sich, dass Fortschrittsindikatoren kaum eingesetzt werden, obwohl sie seit 1976 bekannt sind (Seite 13). Daher untersuchte er, ob Fortschrittsindikatoren dem Benutzer einen Vorteil bringen.

Fortschrittsindikatoren sind notwendig, weil der Benutzer zwischendurch immer wieder mit Zeiten konfrontiert wird, wo der Computer eine Aufgabe noch nicht erledigt hat. Fortschrittsindikatoren zeigen ihm, wie viel einer langen Aufgabe bereits fertig ist. Sie funktionieren wie ein riesiges Thermometer, das „gefüllt“ wird. Der Benutzer kann auf einen Blick sehen, wie viel erledigt ist und wann die Aufgabe vervollständigt ist. Ein Fortschrittsindikator erlaubt es dem Benutzer, einen Prozess während dessen Ausführung zu beobachten. Fortschrittsindikatoren können mit jeder Art von Programm benutzt werden. Typischerweise muss die Anwendung den Fortschrittsindikator explizit aktualisieren. Bei einem Crash kann der Benutzer also feststellen, dass das System nicht mehr läuft (Seiten 11-12).

Fortschrittsindikatoren sind in der Regel Grafiken, keine numerischen Werte. Benutzer können Grafiken schneller erfassen als Zahlen. Wenn keine genauen Werte nötig sind, ist eine Grafik also vorteilhaft. Außerdem impliziert eine Grafik, dass nur eine Schätzung der Restzeit möglich ist. Grafiken können außerdem in kleinen Bildschirmbereichen untergebracht werden, wo sie nicht stören. Fortschrittsindikatoren kann es in verschiedenen Formen geben. Wichtig ist aber, dass der bereits abgearbeitete Anteil sichtbar ist. Es sollte eine zentrale Routine geben, die von allen Programmen benutzt werden kann (Seite 12).

In Mehrprozessumgebungen können Fortschrittsindikatoren für jeden Prozess den Benutzer über die gesamte Systemumgebung informieren. Damit helfen sie Benutzern, mehrere Aufgaben simultan zu planen und zu beobachten (Seiten 11, 16).

Am einfachsten ist der Fortschritt bei Prozessen mit linearer Geschwindigkeit darzustellen. Falls der Fortschritt nicht berechnet werden kann, kann das System dem Benutzer durch einfaches Ausgeben von Punkten, ein sich zufällig bewegendes Objekt oder zufällig variierende Muster zeigen, dass der Prozess noch arbeitet (Seite 13).

Ein Benutzertest zeigte, dass eine große Mehrheit der Benutzer Fortschrittsindikatoren mag. Außerdem werden variable Antwortzeiten akzeptiert, wenn Fortschrittsindikatoren verwendet werden. Variable Antwortzeiten ohne Fortschrittsindikator sorgten für Unzufriedenheit. Ein Fortschrittsindikator bindet die Aufmerksamkeit des Benutzers an das Geschehen auf dem Bildschirm (Seiten 15-16).

Fortschrittsindikatoren sind besonders wichtig für Anfänger, die erwarten, dass alle Aufgaben mit dem Computer schnell erledigt sind und an einen Crash glauben, wenn längere Zeit nichts passiert. Experten, auch wenn sie oft ein Gefühl dafür haben, wie lange eine Aufgabe benötigt, dürften aber auch von Fortschrittsindikatoren profitieren: Fortschrittsindikatoren erlauben es ihnen, in der Wartezeit weitere Aufgaben zu planen und zu erledigen. Eine Alternative zu Fortschrittsindikatoren kann also auch eine numerische Anzeige der verbleibenden Zeit sein, falls diese berechnet werden kann (Seite 16).

### **Musil, 1993**

[Musil, 1993] (Seiten 329-332) führt zwei Arten von zwei Anzeigen auf, die den Benutzer über längere Wartezeiten informieren. Der von ihr sogenannte Performance-Indikator ist ein statisches Instrument zur Vermittlung von Information. Er ist permanent auf dem Bildschirm, während das System arbeitet. Der sogenannte Fortschrittsindikator ist ein dynamisches Instrument zur Vermittlung von Information. Er zeigt nicht nur an, dass das System arbeitet, sondern gibt auch Hinweise, wie lange es noch dauern wird oder wie viel der Aufgabe bereits erledigt ist. Performance-Indikatoren sind ihrer Ansicht nach eher für kurze Standardaufgaben geeignet, während Fortschrittsindikatoren für Aufgaben verwendet werden sollten, die länger als zwei Sekunden dauern – ansonsten kann der Benutzer kaum die Ereignisse auf dem Bildschirm wahrnehmen.

Für alphanumerische Monitore schlägt sie einen blinkenden Cursor oder statische Informationen in einer Statuszeile als Performance-Indikator vor. Ein Fortschrittsindikator könnte in Form einer Statuszeile mit sich ändernder Information, einem aus ASCII-Zeichen gebildeten Balken oder als numerischer Count-Down realisiert werden.

Bei grafischen Monitoren unterscheidet Musil drei verschiedene Klassen von Performance-Indikatoren:

- Symbole, die die vergehende Zeit repräsentieren (z.B. Uhren). Diese Symbole sind am häufigsten zu finden. Sie können für alle Aufgaben verwendet werden.
- Symbole, die zeigen, dass gearbeitet wird (z.B. sich drehende Zahnräder oder ein kehrender Besen, eine Denkblase mit dem Wort *Think*, eine Buddha-Statue). Viele dieser Symbole sind animiert und können für alle Aufgaben verwendet werden. Lediglich bei einer animierten Schaufel empfiehlt Musil, sie nur für Aufgaben zu verwenden, bei denen etwas transportiert wird, und die Buddha-Statue sollte nur

verwendet werden, wenn die Bedeutung in der Zielgruppe als bekannt vorausgesetzt werden kann.

- Symbole, die den Benutzer informieren, was er tun kann (z.B. Kaffee trinken)

Eine weitere Methode eines Performance-Indikators ist das Ändern der Farbe eines Objekts, wenn es beschäftigt ist. Diese Methode ist besonders für Systeme mit mehreren Fenstern geeignet.

Bei Fortschrittsindikatoren führt sie drei Typen an:

- Indikatoren, die sich vorwärts bewegen und so zeigen, wie viel bereits getan wurde (z.B. ein Balken)
- Indikatoren, die sich rückwärts bewegen und so zeigen, wie viel noch zu tun ist (z.B. eine Blume, die Blütenblätter verliert)
- Indikatoren, die sowohl zeigen, was bereits getan wurde als auch, wie viel noch zu tun ist (wie eine Sanduhr, bei der der Sand von der einen in die andere Hälfte fließt)

Balken sind die gebräuchlichste Form eines Fortschrittsindikators. Sie werden stets von links nach rechts gefüllt, und der wesentliche Unterschied zwischen ihnen ist nicht ihre Farbe oder ihr Muster, sondern welche Informationen zusätzlich gegeben werden: Eine Prozentzahl, der Name der Aktion oder anderes. Da Balken sehr abstrakt sind, können sie für alle Arten von Aufgaben verwendet werden.

Die Idee, ein Objekt zu füllen, führt zu anderen „Balken“: Ein sich füllendes Thermometer, eine Blume, die ihre Blütenblätter erhält, ein Drehzahlmesser, der von 0 bis zum Maximalwert geht, eine sich langsam füllende Flasche und nicht zuletzt eine Sanduhr, wo der Sand von der einen in die andere Hälfte fließt. Diese Alternativen sind gut geeignet für Anwendungen mit „Echte-Welt-Aussehen“, da sie nicht so abstrakt wie ein einfacher Balken sind.

Als alternative Fortschrittsindikatoren schlägt sie eine rückwärts laufende analoge oder digitale Stoppuhr vor, eine Person, die einen Fels auf einen Berg schiebt (wie Sisyphus, gleichzeitig die Befürchtung des Benutzers zeigend, dass die Aktion fehlschlägt und automatisch wieder von vorne beginnt), oder ein Sandhaufen, der Stück für Stück von einem Ort zu einem anderen transportiert wird. Computerspiele sind ihrer Meinung nach eine gute Quelle für alternative Performance-Indikatoren.

Auch auf die Verwendung akustischer Signale geht sie ein. Ein konstanter Ton oder eine Melodie könnte gespielt werden, wenn das System beschäftigt ist. Um Klänge als Fortschrittsanzeige zu verwenden, könnte beispielsweise eine Tonskala imple-

mentiert werden: Die Aufgabe wird in mehrere Teile zerlegt und jedem Teil wird eine Note zugeordnet. Eine andere Möglichkeit ist ein Geräusch, das die sich ändernde Restzeit anzeigt, beispielsweise ein sich änderndes Echo oder das Geräusch einer Flasche, die geleert wird. Musil weist auch auf die Probleme beim Einsatz akustischer Signale hin.

### **Cooper, 1995**

[Cooper, 1995] untersucht zunächst, ob Informationen zu einem Prozess in einem separaten Dialog oder im Hauptfenster des Anwendungsprogramms angezeigt werden sollen. Prozesse, die ein integraler Bestandteil des im Hauptfenster Angezeigten sind, sollten dort und nicht in einem separaten Dialog angezeigt werden (Seite 320).

Eine Anzeige des Prozessfortschritts muss durch zeitbezogene Änderungen dem Benutzer den Eindruck vermitteln, dass der Prozess normal abläuft. Sie sollte ihm auch eine Möglichkeit zum Abbruch bieten. Ein Fortschrittsindikator muss nicht unbedingt eine abstrakte Grafik sein, sondern kann gerne einen Bezug zur Aufgabe haben wie beispielsweise ein Hund, der zu einer Zeitung läuft und so den Fortschritt beim Laden einer Online-Zeitung anzeigt. Wird der Prozessfortschritt in einer Dialogbox angezeigt, darf diese nach Beendigung des Prozesses automatisch geschlossen werden (Seiten 316-317, 419, 425).

[Cooper, 1995] kritisiert zwei Punkte bei Statusanzeigen in Programmen. Zum einen hält er es für falsch, Statusanzeigen mit einer Dialogbox, die den Benutzer bei der Arbeit unterbricht, anzuzeigen (Seite 443, Figure 29-1). Zum anderen kritisiert er die insbesondere bei Installationsprogrammen häufig zu findende Unart, bei einem Fortschrittsindikator nicht anzugeben, was dieser anzeigt: Den Fortschritt für einen bestimmten Teilprozess oder den gesamten Prozess? Kopierte Bytes oder die benötigte Zeit? Fortschrittsindikatoren, die den Benutzer hierüber im Unklaren lassen, hält er für sinnlos, weil sie schnell ignoriert werden (Seite 529).

### **Statische Zustände**

In Bezug auf statische Zustände sind zwei Ideen vorherrschend: Zum einen das Ändern des Mauszeigers, um einen Modus anzuzeigen, zum anderen die Verwendung von animierten Icons, um unter anderem den Status von Daten zu zeigen.

### **Mauszeigerformen**

Bereits [Muller, 1988] machte sich Gedanken über Modifikationen des Mauszeigers, um den Systemmodus anzuzeigen. Er forderte, dass der Mauszeiger eindeutig modi-

fiziert werden soll, also nicht eine Form für mehrere Modi verwendet wird, um so die kognitive Belastung des Benutzers zu reduzieren (Seite 90).

[Cooper, 1995] empfiehlt eine Umkehr von der üblichen Praxis, bei der der Mauszeiger bei Drag&Drop-Aktionen anzeigt, wenn das gezogene Objekt momentan nicht fallen gelassen werden kann. Er fordert stattdessen, dass der Mauszeiger anzeigt, wenn das gezogene Objekt sich über einem möglichen Ziel befindet. Auch das potenzielle Ziel sollte in diesem Augenblick sein Aussehen ändern (Seite 245). [Cooper, 1995] sieht eine geänderte Form des Mauszeigers ausdrücklich als Statusanzeige an (Seiten 245-246).

### **Status von Daten**

Die Idee, animierte Icons zur Vermittlung von Status zu benutzen, ist unter anderem bei [Cooper, 1995] (Seite 541), [Tognazzini, 1996] (Seite 196) und [Shneiderman, 1998] (Seite 209) zu finden. In älteren Werken war hierüber noch nichts zu lesen, so dass es sich vermutlich um einen neueren Gedanken handelt. Um die Entwicklung zu illustrieren, habe ich die vorgestellten Style Guides chronologisch geordnet.

## **5.2 Statusanzeigen in kommerziellen Style Guides**

Jedes grafische Betriebssystem stellt dem Systementwickler eine Menge vordefinierter Bedienelemente, sogenannte *Widgets*, zur Verfügung. Eine Übersicht über die Widgets sowie eine Anleitung für ihren Gebrauch liefern die vom Hersteller des Betriebssystems erstellten Style Guides (vgl. [Herzer und Noll, 1990], Seiten 17-18). Style Guides geben zum einen dem Systementwickler ein durchdachtes Regelwerk zur Gestaltung der Benutzungsoberfläche an die Hand und ermöglichen es ihm dadurch, den Entwicklungsaufwand deutlich zu reduzieren. Zum anderen führt die Befolgung der Regeln in einem Style Guide zu der für den Benutzer erfreulichen Tatsache, dass die Bedienung verschiedener Anwendungsprogramme einander mehr oder weniger ähnelt; der Benutzer erhält also konsistente Programme. Konsistente Software führt potenziell zu einem ästhetischeren Interface. Style Guides haben jedoch nicht nur Vorteile. Ihre Entwicklung ist teuer und bringt zunächst keinen Profit. Es besteht die Gefahr, dass der kleinste gemeinsame Nenner festgeschrieben wird, eventuell sind sogar schlechte Regeln enthalten ([Nielsen, 1989], Seiten 3-5).

Konsistenz innerhalb eines Computersystems ist wie ein Versprechen dem Benutzer gegenüber, das nicht gebrochen werden sollte ([Nielsen, 1989], Seite 5). Daher sollten Style Guides meiner Ansicht nach so umfangreich wie nötig sein und bezüglich Statusanzeigen mindestens Fortschrittsindikatoren enthalten. Leider wird diese Ansicht nicht allgemein geteilt. In der von [Shneiderman, 1998] (Seiten 100-102)

aufgestellten Liste, was Style Guides enthalten sollen, werden Statusanzeigen nicht erwähnt. Wenn es jedoch um die Auswahl von User-Interface-Building-Programmen<sup>1</sup> geht, sind für ihn die unterstützten Widgets, zu denen auch Anzeigen wie Messuhren gehören sollten, eines der Kriterien (Seite 180).

Während die Statusvermittlung über den Mauszeiger von jeher Bestandteil von Style Guides war, ebenso wie Statusleisten – wenn auch mit unterschiedlichem Aussehen –, so sind Fortschrittsanzeigen erst in neuerer Zeit ein Bestandteil der Widgets. Ältere Style Guides haben höchstens einmal erwähnt, dass ein Prozessfortschritt grafisch dargestellt werden kann oder sollte, ohne dass jedoch entsprechende Widgets zur Verfügung standen.

### **IBM SAA-Common User Access (CUA)**

Die Informationen über den Style Guide von IBM SAA-CUA, dem ältesten der hier vorgestellten Style Guides, basieren auf der Beschreibung in [Eberleh, 1994] (Seite 164). Eberleh erwähnt, dass Icons durch verschiedene Darstellungen vermitteln, ob das zugehörige Objekt bereits in Gebrauch ist, ausgewählt wurde oder momentan nicht auswählbar ist. Abschaltbare akustische Signale sollen den Benutzer auf unzulässige Aktionen hinweisen. CUA sieht außerdem eine Informations- und eine Statuszeile für textuelle Informationen vor. Die Informationszeile am unteren Bildschirmrand enthält Hinweise zum Objekt oder der Aktion unter dem Mauszeiger. Hier können auch Informationen über den Ablauf eines Prozesses ausgegeben werden. In der Statuszeile unterhalb der Menüzeile wird der aktuelle Status des gewählten Objekts ausgegeben. Eberleh erwähnt außerdem, dass verschiedene Formen des Mauszeigers dem Benutzer Rückmeldungen geben.

### **OSF/Motif**

Der Style Guide von OSF/Motif ([OSF/Motif, 1991]) sieht das Vermitteln des momentanen Status als eine wichtige Komponente der Interaktion mit dem Benutzer. Status sollte mit Hilfe von Beschriftungen und Grafiken vermittelt werden. Das Anwendungsprogramm sollte den Status dynamisch anzeigen, zum Beispiel durch einen geänderten Mauszeiger, wenn spezielle Aktionen möglich sind (Seite 4-50). Die verschiedenen vom Betriebssystem zur Verfügung gestellten Mauszeiger werden bezüglich Form und Verwendung ausführlich vorgestellt (Seiten 2-9 bis 2-12).

---

<sup>1</sup> Stark vereinfacht formuliert ist ein User-Interface-Building-Programm ein in Software realisierter Style Guide: Dank der im User-Interface-Building-Programm implizit enthaltenen Regeln ist sichergestellt, dass damit erstellte Anwendungsprogramme konform zum Style Guide sind.

Bei lange andauernden Aktionen könnte der Benutzer fälschlicherweise glauben, dass das Anwendungsprogramm oder das System nicht mehr arbeitet. Solche Verzögerungen sollten mit einem speziellem Dialog, dem sogenannten WorkingDialog, angezeigt werden. Um die Arbeit des Benutzers nicht zu unterbrechen, sollte dieser Dialog nicht modal sein. Er sollte ein spezielles Symbol und eine Nachricht enthalten und falls möglich analog zum Fortschritt aktualisiert werden (Seiten 4-32 und 4-51).

Primärfenster eines Anwendungsprogramms haben am unteren Rand eine Mitteilungszeile. Diese könnte als Statuszeile verwendet werden, jedoch scheinen die Autoren des Style Guide sie hauptsächlich für flüchtige textuelle Mitteilungen vorgesehen zu haben (Seiten 4-13 bis 4-17).

Der Style Guide rät auch, wie die Aufmerksamkeit des Benutzers gezielt erregt werden kann. Dazu sollte die Masse der Elemente auf dem Bildschirm zurückhaltend gestaltet werden. Dann können Kontraste gezielt eingesetzt werden, um die Aufmerksamkeit des Benutzers zu erregen. Farbe sollte nur als redundantes Merkmal bei der Gestaltung eines Bildschirms eingesetzt werden, primär sollten Form und Größe zur Unterscheidung von Bildelementen verwendet werden.<sup>1</sup> Als Beispiel hierfür wird das Stoppschild aufgeführt, das in vielen Teilen der Welt ein rotes Achteck ist und aufgrund von Form und Farbe erkannt wird (Seite 1-7).

### **Apple Macintosh**

Im Style Guide für den Apple Macintosh ([Apple, 1992]) gibt es kein separates Kapitel über die Vermittlung von Statusinformationen. Jedoch erhält der Systementwickler einige Hinweise, die sich für die Anzeige von Statusinformationen nutzen lassen. So wird beispielsweise darauf hingewiesen, dass Fortschrittsinformationen vom Betriebssystem erst nach einer kurzen Wartezeit ausgegeben werden: Bei längeren Finder-Aktionen beispielsweise nach 4 s, beim Leeren des Mülleimers nach 8 s. Apple berücksichtigt hierbei die Erwartungen des Benutzers an die Aktion und geht bei diesen konkreten Beispielen davon aus, dass der Benutzer erwartet, dass das Leeren des Mülleimers einen Augenblick in Anspruch nehmen wird, und daher eine kurze Wartezeit ohne Meldung akzeptiert (Seite 57).

Informationen über den aktuellen Modus werden beim Macintosh häufig im Menü angezeigt. Häkchen und Striche neben einem Eintrag beispielsweise bedeuten, dass dieser Punkt für die komplette aktuelle Selektion im Dokument gilt oder nur für Teile

---

<sup>1</sup> 1991 konnte ein Farbbildschirm noch nicht als selbstverständlich vorausgesetzt werden. Interessant ist meiner Ansicht nach, dass der Style Guide hier durchaus vorausschauend Hinweise gibt.

davon (Seiten 64 ff); eine Raute im Application-Menü zeigt an, dass die entsprechende Applikation auf Eingabe wartet (Seiten 71 ff).

Für längere Vorgänge wie z.B. das Konvertieren von Dokumenten soll ein modaler Dialog wie in Abbildung 20 verwendet werden, der dem Benutzer grafisch den bereits erledigten Teil anzeigt und ihm eine Möglichkeit bietet, den Vorgang abubrechen (Seite 190).

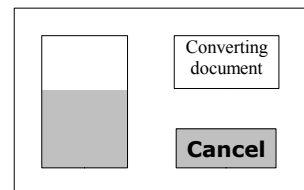


Abbildung 20: Konvertieren von Dokumenten (nach [Apple, 1992], Seite 190)

## NeXTSTEP

Im Gegensatz zum gleich alten Style Guide für den Apple Macintosh beschäftigt sich der Style Guide von NeXTSTEP ([NeXTSTEP, 1992]) ausführlich mit verschiedenen Aspekten von Statusanzeigen. Insbesondere die Verwendung modaler Werkzeuge sowie die Vermittlung von Statusinformation über Bedienelemente werden ausführlich erläutert. Auf einen Fortschrittsanzeiger geht der Style Guide jedoch nicht ein.

Anwendungen, die mit modalen Werkzeugen arbeiten, sollten sie dem Benutzer in einer Palette anbieten. Wird ein Werkzeug gewählt, soll der Mauszeiger eine andere Form annehmen, und das gewählte Werkzeug soll in der Palette markiert werden. Damit weiß der Benutzer stets, welches Werkzeug gerade gewählt ist. Jedes dieser Werkzeuge etabliert einen Modus, also einen Zeitraum, in dem die Aktionen des Benutzers auf eine besondere Art interpretiert werden. Auch wenn ein Modus die Möglichkeiten des Benutzers einschränkt und daher normalerweise nicht verwendet werden sollte, gibt es diverse mildernde Faktoren. Der Modus ist ein direktes Resultat einer Aktion des Benutzers. Er ist durch den veränderten Mauszeiger und das markierte Werkzeug in der Palette deutlich erkennbar, und es gibt einen einfachen Weg, den Modus zu verlassen. Außerdem ähnelt das Verhalten modaler Werkzeuge dem Verhalten von Werkzeugen in der realen Welt. Die Verwendung modaler Werkzeuge ist angemessen, wenn davon ausgegangen werden kann, dass der Benutzer ein Werkzeug über einen längeren Zeitraum nutzt, beispielsweise zum Zeichnen mehrerer Linien (Seite 19). Das Aussehen des Mauszeigers sollte einen Hinweis auf das gewählte Werkzeug geben. Beispielsweise kann der Mauszeiger beim Zeichnen von Linien wie ein Bleistift aussehen, oder beim Zeichnen breiterer Striche wie ein breiter Pinsel. Sind Aktionen mit der Maus nur in einem bestimmten Bereich auf dem Bildschirm möglich, sollte der Mauszeiger seine normale Form annehmen, wenn er diesen Bereich verlässt. Generell sollte der Mauszeiger jedoch nicht



zu oft geändert werden. Der normale Pfeil sollte verwendet werden, wann immer er sinnvoll ist (Seite 46).

Der NextSTEP Style Guide rät davon ab, Grafiken wie zum Beispiel Häkchen in Menüs zu verwenden, um Statusinformationen anzuzeigen. Meist gibt es eine bessere Möglichkeit, sie anzuzeigen – unter anderem Schaltflächen oder Checkboxen in Dialogfenstern. Inaktive Menüeinträge sollten entweder nicht gewählt werden können oder ein Dialogfenster mit einer Erklärung öffnen. Wenn ein inaktiver Eintrag mit einem Tastaturkommando gewählt wird, soll ein kurzer Ton den Benutzer über seine ungültige Eingabe informieren (Seite 110).

Auch die Titelzeile eines Fensters ist laut NeXTSTEP Style Guide kein guter Ort für Statusinformationen. Informationen wie die Auskunft darüber, was das Programm gerade tut, sollten im Hauptfenster oder in einem Dialogfenster angezeigt werden. Im Hauptfenster werden Statusinformationen meist in kleinem dunkelgrauen Text ausgegeben (Seite 69). Trotzdem sollte eine veränderte Schaltfläche für das Schließen eines Dokumentenfensters dem Benutzer anzeigen, dass Änderungen am Dokument noch nicht gespeichert wurden (Seite 55).

Status sollte ebenfalls nicht mit einer Schaltfläche mit variablem Text angezeigt werden, da der Benutzer die Aufschrift eher als Kommando und nicht als momentanen Status interpretieren wird. Daher sollte der Status mit anderen Mitteln angezeigt werden (Seite 138). Eine Ausnahme hiervon ist lediglich eine Schaltfläche, die während einer längeren Aktion vorübergehend zu einer STOP-Schaltfläche geändert wird und bei der diese Änderung bei beendeter oder abgebrochener Aktion rückgängig gemacht wird (Seite 142).

Der Benutzer kann Application Icons am rechten Rand andocken, um einen schnelleren Zugriff auf die entsprechenden Applikationen zu haben. Die Icons sind ständig sichtbar, egal, ob die Applikation gerade gestartet ist oder nicht. Drei zusätzliche graue Punkte in der unteren linken Ecke des Icons bedeuten, dass die Applikation momentan nicht gestartet ist. Ist das normalerweise grau hinterlegte Icon weiß hinterlegt, wird die entsprechende Applikation gerade gestartet (Seite 6).

Um Status anzuzeigen, können verschiedene Bedienelemente verwendet werden. Bedienelemente, die den Status anzeigen, sollten nicht gleichzeitig zum Aufruf von Funktionen verwendet werden (Seite 154).

NeXTSTEP empfiehlt, bei ungültigen Eingaben in einem Textfeld die komplette Eingabe zu selektieren und zu markieren (Seite 145).

## Microsoft Windows

Der Style Guide von Microsoft Windows enthält zunächst das Übliche wie einen Überblick über verschiedene Mauszeigerformen und ihren Einsatz. 16 gebräuchliche Mauszeiger-typen werden gezeigt, und ihr Einsatz be-schrieben ([MSDN, 2000/3]). Außerdem zeigt Microsoft noch verschiedene Over-lays<sup>1</sup> (Abbildung 21), die bei Drag&Drop-Aktionen rechts unten am Mauszeiger zu sehen sind ([MSDN, 2000/4]).



Abbildung 21: Mauszeiger-Overlays: Verschieben, Kopieren, Verknüpfung er-stellen ([MSDN, 2000/4], Figure 6.9)

Auch die Statusanzeige durch Menüeinträge wird erläutert. Microsoft Windows bietet zwei Möglichkeiten. Entweder kann ein einzelner Menüeintrag wie eine Checkbox genutzt werden oder eine Gruppe von Menüeinträgen wird wie Option-buttons verwendet. Einträge, die wie Checkboxes verwendet werden, sind bei ge-setztem Status durch einen vorangestellten Haken markiert. In einer Gruppe von Einträgen, die wie Optionbuttons verwendet werden, ist der momentan gewählte Eintrag durch ein vorangestelltes Bullet markiert ([MSDN, 2000/6]).

Neben den üblichen Checkboxes und Optionbuttons gibt es unter Windows noch Checkboxes, die einen dritten Zustand annehmen können. Dieser soll dann angezeigt werden, wenn der zugehörige Status für ei-nige, aber nicht für alle Elemente der Selektion gilt ([MSDN, 2000/7]).



Abbildung 22: Gemischte Selektion in einer Checkbox ([MSDN, 2000/7], Figure 8.16)

Schieberegler und ihre Verwendung werden ebenfalls erläutert ([MSDN, 2000/7]).

Microsoft Windows bietet auch einen Fortschrittsbalken an. Dieser besteht entweder aus einem einteiligen oder aus einem segmentierten Balken, der von links nach rechts „gefüllt“ wird. Die einteilige Darstellung ist die Voreinstellung und wird nach Aus-sage von Microsoft auch am häufigsten verwendet. Um den Zweck des Fortschritts-balkens zu vermitteln, sollten zusätzlich Texte oder andere Informationen verwendet werden. Fortschrittsbalken sollten für lange andauernde Aktionen oder für Hinter-groundprozesse verwendet werden. Abhängig von der Modalität des angezeigten Pro-zesses empfiehlt Microsoft, den Fortschrittsbalken entweder in einem separaten Dialog oder in der Statuszeile anzuzeigen ([MSDN, 2000/7]).

Die Statusleiste ist ein spezieller Bereich im Fenster, normalerweise am unteren Rand, wo Informationen über den Status vom Hauptfenster-Inhalt oder andere In-

<sup>1</sup> Ein kleiner Teil in der rechten unteren Ecke der quadratischen Grundfläche des Mauszeigers wird durch das Overlay ersetzt.

formationen mit Bezug hierauf, beispielsweise der Tastaturstatus, angezeigt werden. Auch wenn eine Statusleiste Bedienelemente enthalten kann, so enthält sie normalerweise nur Informationsanzeigen. Microsoft bietet Statusleisten als eigenes Bedienelement an. Bedienelemente in Statusleisten sollten immer nur als zusätzliche Möglichkeit angeboten werden. Bedienelemente in einer Statusleiste, die kein Textlabel haben, sollten einen ToolTip bekommen ([MSDN, 2000/8]). Zum Anzeigen von Informationen in der Statusleiste werden Statusfelder verwendet, die durch die Art der Umrandung gekennzeichnet sind ([MSDN, 2000/9]).

Nicht nur die Statusleiste vermittelt Statusinformationen, sondern auch der Mitteilungsbereich in der Windows Task-Leiste. Dieser soll nur für kurze temporäre Mitteilungen oder für Statusinformation zu einem laufenden Prozess oder Ereignis verwendet werden. Ein Icon in der Taskleiste soll nur dann persistent sein, wenn es für den Benutzer notwendig ist, beispielsweise bei einer DFÜ-Verbindung. Die Beachtung dieser Regeln hilft dem Benutzer, Änderungen in diesem Bereich zu bemerken ([MSDN, 2000/1]).

### **5.3 Beispiele für Statusanzeigen**

Statusanzeigen gibt es nicht nur bei Software. Auch die Hardware informiert den Benutzer über ihren Status.

#### **Hardware**

Statusinformationen der Hardware werden dem Benutzer überwiegend auf drei Arten vermittelt: Über LEDs, über Betriebsgeräusche und über Texte. Zwei wesentliche Unterschiede bestehen: Der offensichtliche Unterschied ist der sensorische Kanal, optisch für LEDs und Texte, akustisch für die Betriebsgeräusche. Ein weiterer wichtiger Unterschied ist aber, dass optische Signale über LEDs oder Texte bewusst gestaltet wurden, während Betriebsgeräusche ein nicht gestaltetes zufälliges Produkt sind.

LEDs treten sowohl einzeln als auch in Gruppen auf. Sie können, wie beispielsweise bei Monitoren häufig zu sehen, durch Farbwechsel verschiedene Informationen vermitteln. Zu finden sind ebenfalls LEDs, die sowohl konstant leuchten als auch mit verschiedenen Frequenzen blinken.

Auch unter inhaltlichen Aspekten betrachtet werden LEDs unterschiedlich verwendet. Einige LEDs informieren über einen (meist vom Benutzer gewählten) Modus, beispielsweise darüber, wie die Tasten des Ziffernblocks interpretiert

werden. Andere LEDs informieren über Ereignisse, beispielsweise über einen Zugriff auf die Festplatte.

Textuelle Statusinformation über die Hardware erhält ein Benutzer meist nach dem Einschalten des Systems. Ein typisches PC-System beispielsweise informiert über das verwendete BIOS, den vorhandenen Speicher, über eingebaute Laufwerke und weitere Hardware-Komponenten. Diese Informationen sind nur sehr kurz sichtbar.

Auch wenn Betriebsgeräusche sich der Gestaltung entziehen, so werden sie dennoch vom Benutzer verarbeitet. In der Regel handelt es sich um mechanische Geräusche, beispielsweise Lüftergeräusche oder Geräusche der Kopfpositionierung einer Festplatte. Zu beobachten sind gleichförmige Geräusche, die mitunter durch rhythmische Komponenten überlagert sind und unregelmäßige Geräusche. Auch Geräusche informieren den Benutzer sowohl über Modi wie zum Beispiel Betriebsbereitschaft als auch über Ereignisse, beispielsweise einen Zugriff auf die Festplatte. Meist informieren gleichförmige Geräusche über Modi und unregelmäßige Geräusche über Ereignisse. Durch Betriebsgeräusche werden also in der Regel statische Statusinformationen vermittelt, keine dynamischen.

Statusanzeigen gibt es jedoch nicht nur bei PCs. Auch Mobiltelefone und viele weitere technische Geräte haben Statusanzeigen. Es würde jedoch den Rahmen dieser Arbeit sprengen, auch diese Geräte noch zu untersuchen.

## Betriebssysteme und Hilfsprogramme

### Windows NT 4.0

Betriebssysteme mit grafischer Oberfläche arbeiten mit mehreren Fenstern. Das aktive Fenster, das Eingaben des Benutzers aufnimmt, unterscheidet sich lediglich durch die Farbe der Titelzeile von den inaktiven Fenstern.

In der Standardeinstellung hat

das aktive Fenster eine farbige Titelzeile, inaktive Fenster haben eine graue. Jedoch kann der Benutzer die Farben ändern, so dass diese intuitiv leicht zu erfassende Bedeutung bei anderen Farbkombinationen nicht zwangsläufig ebenfalls gegeben ist.

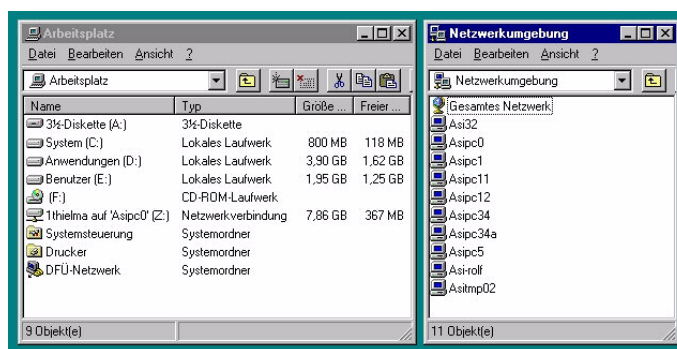


Abbildung 23: Aktives (rechts) und inaktives Fenster (links)

Das linke Fenster in Abbildung 23 zeigt eine Liste aller Laufwerke in einem System. Zu jedem Laufwerk werden der Typ (textuell), die gesamte Größe und der noch freie Speicherplatz (numerisch) angezeigt. Detailliertere Informationen zu einem bestimmten Laufwerk werden in einem gesonderten Dialog ausgegeben (Abbildung 24). Gesamter und freier Speicherplatz werden numerisch angezeigt. Eine Grafik (Tortendiagramm) zeigt zusätzlich den Anteil des freien Speichers am gesamten zur Verfügung stehenden Speicher. Einige weitere Informationen werden textuell angezeigt.

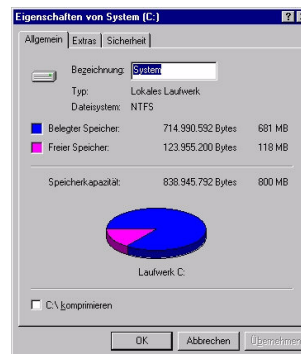


Abbildung 24: Laufwerkeigenschaften

Auf einer weiteren Registerseite des Eigenschaften-Dialogs von Laufwerken kann ein separater Dialog aufgerufen werden, wo der Benutzer Informationen über Verzeichnisberechtigungen für das Laufwerk erhält (Abbildung 25). Ein Icon sowie ein Text kennzeichnen eine Klasse von Benutzern, die erlaubten Aktionen sind in einer Listbox textuell dargestellt.

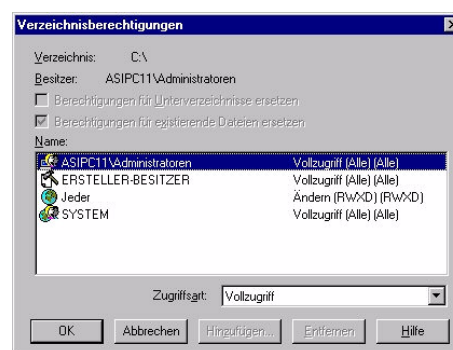


Abbildung 25: Verzeichnisberechtigungen

Grafische Betriebssysteme stellen bereits verschiedene Mauszeigerformen zur Verfügung, die von den Anwendungsprogrammen genutzt werden können. Das Aussehen des Mauszeigers informiert über den Modus des Anwendungsprogramms. Abbildung 26 zeigt die von Windows NT 4.0 zur Verfügung gestellten Mauszeigertypen, die vom Betriebssystem selbst genutzt werden, die aber auch allen Anwendungsprogrammen zur Verfügung stehen. Im wesentlichen handelt es sich um Mauszeigertypen für das Manipulieren von Fenstern, das Vermitteln der Systemauslastung sowie Auswahl-funktionen für grundlegende Dokumenttypen.

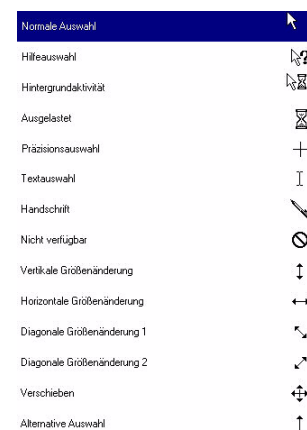


Abbildung 26: Verschiedene vom Betriebssystem zur Verfügung gestellte Mauszeiger

Während des Kopierens von Dateien zeigt Windows NT 4.0 einen separaten Dialog an. Eine Animation versucht, dem Benutzer das Warten zu verkürzen und den Vorgang zu visualisieren: Ein Blatt Papier bewegt sich



Abbildung 27: Kopieren mehrerer Dateien

in einer Endlosschleife vom linken zum rechten Ordner. Es werden der Name der aktuell gerade zu kopierenden Datei sowie die nicht vollständigen Namen von Quell- und Zielverzeichnis angezeigt. Ein segmentierter Fortschrittsbalken visualisiert den bereits kopierten Anteil der Datei. Dieser Dialog wird erst ab einer bestimmten Dateigröße angezeigt, sehr kleine Dateien werden ohne Rückmeldung kopiert. Bei sehr großen Dateien wird wie in Abbildung 27 eine Restzeit ausgegeben.

Windows NT 4.0 stellt dem Benutzer mit dem DFÜ-Netzwerk auch ein Hilfsprogramm zur Verfügung, um Verbindungen ins Internet herzustellen. Direkt nach dem Herstellen der Verbindung wird für einige Sekunden eine Meldung über den Status der Verbindung eingeblendet (siehe Abbildung 28 rechts).



Abbildung 28: Offline und Online (Farbe s. Anh.)

Anschließend ist diese Meldung nicht mehr aufzurufen. Mit dem Aufbau der Verbindung wird der DFÜ-Monitor gestartet, ein Hilfsprogramm zum Beobachten von DFÜ-Verbindungen. In der rechten unteren Bildschirmcke erscheint bei gestartetem DFÜ-Monitor ein Icon in Form eines Telefons. Die Farbe dieses Icons verrät dem Benutzer, ob er momentan online (Gelb) oder offline (Grau) ist. Wenn Daten übertragen werden, verfärbt sich der graue Hintergrund um das gelbe Telefon blau (siehe Abbildung 29 rechts). Da die Daten paketweise übertragen werden, entsteht ein unregelmäßiges blaues Blinken, das dem Benutzer einen Hinweis gibt, ob und mit welcher Geschwindigkeit Daten eintreffen.

## RVS-COM

RVS Com ist ein Hilfsprogramm, das virtuelle Modems an ISDN-Anschlüssen zur Verfügung stellt. Im rechten Teil der



Abbildung 29: Offline und online (Farbe s. Anh.)

Task-Leiste erscheint ein kreisförmiges Icon, dessen acht Segmente in den Farben Rot, Gelb und Blau gefärbt sind. Diese Segmente verfärben sich in Grün und Gelb, wenn ein von RVS-COM zur Verfügung gestelltes virtuelles Modem genutzt wird.

## ZoneAlarm

ZoneAlarm ist eine Firewall. Sie schützt einen am Internet hängenden PC vor unberechtigten Zugriffen von außen. Im rechten Teil der Task-Leiste erscheint



Abbildung 30: Datenübertragung (Farbe s. Anh.)

bei gestarteter Firewall ein weiteres Icon (Abbildung 30, drittes Icon von links). In der oberen Hälfte zeigt es eine Reihe simulierter roter LEDs, in der unteren Hälfte das selbe in Grün. Die obere Reihe zeigt ausgehende, die untere eingehende Daten

an. Je näher die „leuchtenden“ LEDs am Vollausschlag sind, desto mehr Datenverkehr besteht. ZoneAlarm zeigt mit diesem Icon interessanterweise weniger den eigenen Status, dafür aber sehr gut den Status der Verbindung zum Internet.

### Windows Commander 3.02

Der Windows Commander ist ein Hilfsprogramm für die Dateiverwaltung unter Windows und hilft im wesentlichen beim Kopieren und Verschieben von Dateien. Abbildung 31 zeigt den Windows Commander beim Kopieren mehrerer Dateien. Die zu kopierende Dateien sind farbig hervorgehoben. Informationen über den Kopiervorgang werden in einem separaten Dialog ausgegeben.

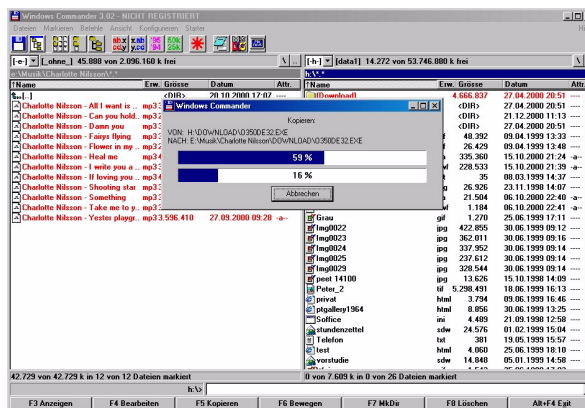


Abbildung 31: Kopieren mehrerer Dateien

Der komplette Name von Quelle und Ziel der aktuell kopierten Datei wird textuell ausgegeben. Je ein Fortschrittsbalken informieren über das Kopieren der aktuellen Datei sowie über den gesamten Prozess. In der Mitte der Grafik steht zusätzlich der entsprechende numerische Wert. Beim Kopieren einer einzelnen Datei wird nur ein Fortschrittsbalken angezeigt – Gesamtprozess und Teilprozess sind schließlich identisch, so dass bei getrennten Balken beide Balken das gleiche anzeigen würden.

Auch das normale Fenster des Windows Commander liefert Statusinformationen. Unter der Visualisierung eines Verzeichnisses wird textuell bzw. numerisch informiert, wie viele Dateien im Verzeichnis enthalten sind, wie groß diese Dateien zusammen sind, wie viele Dateien markiert sind und wie groß die markierten Dateien zusammen sind. Über der Visualisierung werden die Gesamtgröße des gewählten Laufwerks sowie der noch freie Anteil numerisch angezeigt.

### Windows Taskmanager

Der Windows Taskmanager von Windows NT 4.0 dient sowohl zum Verwalten laufender Prozesse als auch zur Information über die Systemauslastung. Die Informationen werden auf drei Registerseiten verteilt. Die erste Registerseite (ohne Abbildung) gibt eine Übersicht über die aktuellen Anwendungen und deren Status (Wird ausgeführt, Keine Rückmeldung). Der Status wird textuell dargestellt. Der Benutzer kann einen Prozess auswählen und einige Aktionen mit ihm ausführen, beispielsweise ihn beenden oder zum zugehörigen Prozess wechseln.



Die zweite Registerseite (ohne Abbildung) gibt eine Übersicht über die laufenden Prozesse. Zu jedem Prozess werden eine ID, die momentane CPU-Nutzung in %, die seit dem Start verbrauchte CPU-Zeit sowie der momentane Speicherbedarf angezeigt. Diese Informationen zu den einzelnen Prozessen werden numerisch präsentiert. Der Benutzer kann einen Prozess auswählen und ihn beenden.

Die dritte Registerseite (Abbildung 32) informiert über die Systemleistung. Angezeigt werden die CPU-Nutzung sowie der Speicherbedarf, sowohl die momentane Auslastung als auch der Verlauf über einen nicht weiter spezifizierten Zeitraum. Diese Informationen werden grafisch dargestellt. Die momentanen Auslastungen werden in Form einer simulierten LED-Aussteuerungsanzeige, wie sie bei Cassettenrecordern gefunden werden kann, angezeigt. Eine Skala gibt es nicht, jedoch wird der exakte Wert unter der Grafik numerisch angezeigt. Die Informationen über den Verlauf werden in einem simulierten Display eines wissenschaftlichen Messinstruments als Kurve angezeigt. Die CPU-Auslastung kann optional anzeigen, welcher Teil für Prozesse des Betriebssystemkerns benötigt wird. Der entsprechende Anteil wird mit roten statt grünen LEDs bzw. einer zusätzlichen roten Kurve angezeigt. Weitere Systemparameter werden in der unteren Hälfte numerisch präsentiert.

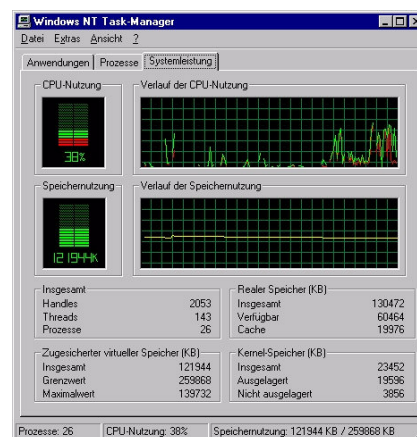


Abbildung 32: Systemleistung (Farbe s. Anh.)

## Windows 2000

Bei Windows 2000, dem Nachfolger von Windows NT 4.0, können im Hintergrund laufende Programme den Benutzer informieren, dass sie seine Aufmerksamkeit benötigen. Dazu wird ihre Schaltfläche in der Task-Leiste farbig hervorgehoben.

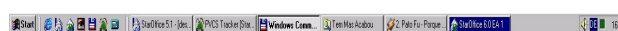


Abbildung 33: Hervorgehobener Task benötigt die Aufmerksamkeit des Benutzers

## Installationsprogramme

Installationsprogramme setzen einen längeren Prozess in Gang. Die Betrachtung von Installationsprogrammen zeigt daher verschiedene Möglichkeiten, wie ein Prozessfortschritt visualisiert wird.



## StarOffice 5.2

Die Installation eines umfangreichen Programmpakets wie StarOffice dauert auf einem Computer mittlerer Leistung etwa 15 Minuten. Zudem sind umfangreiche Einstellungen möglich, welche Programmteile installiert werden sollen. Das Installationsprogramm nimmt den kompletten Bildschirm in Beschlag. Die Informationen über den Fortschritt der Installation werden im Hauptfenster angezeigt. Lediglich für Eingriffe durch den Benutzer werden separate Dialogfenster geöffnet. Abbildung 34 zeigt den Installationsprozess noch recht weit am Anfang. Der Benutzer hat hier die Möglichkeit, die zu installierenden Programmteile auszuwählen. Die Optionen sind in einer sogenannten Tree View auswählbar, bei der semantisch zusammengehörende Optionen auf einer höheren Ebene zusammengefasst werden. Damit ergeben sich für einen Knoten im Baum drei Zustände: Zunächst einmal natürlich *Die Option ist gewählt* und *Die Option ist nicht gewählt* für die Blätter des Baums. Vollständig gewählte Optionen sind mit einem farbigen Icon gekennzeichnet, nicht gewählte Optionen sind durch ein Icon gekennzeichnet, das nur die Konturen zeigt. Für die Knoten, die Verzweigungen enthalten, gibt es noch die Möglichkeit *Einige Optionen der nächsten Ebene sind gewählt*. Diese Option hat ebenfalls ein farbiges Icon, jedoch mit geringer gesättigten Farben. Am linken Rand des Hauptfensters ist eine geordnete Liste der einzelnen Schritte der Installation. Der aktuelle Schritt ist durch hellere und fettere Schrift hervorgehoben.



Abbildung 34: Auswahl der zu installierenden Programmteile, kurz nach Beginn der Installation (Farbe s. Anh.)

Die bereits in Abbildung 34 sichtbare Liste aller Schritte der Installation wird im weiteren Verlauf durch einen nicht segmentierten Fortschrittsbalken ergänzt. In seiner Mitte wird der prozentuale Anteil zusätzlich numerisch angezeigt. Nach kurzer Wartezeit (wenige Sekunden) erscheint unter dem Fortschrittsbalken ein Icon in Form einer Uhr, neben der die ungefähr noch benötigte Zeit für die restliche Installation steht (siehe Abbildung 36).



Abbildung 35: Fortschrittsanzeige

Während des langdauernden Kopierens der Programmdateien wird der größte Teil des Hauptfensters für ein sogenanntes Installationskino genutzt. Wie bei einer Bildschirmpräsentation wird in regelmäßigen Abständen ein neues Bild gezeigt. Der Inhalt des Installationskinos ist zum größten Teil Werbung für das Produkt, der Kunde wird hier aber auch auf neue Features der aktuellen Version hingewiesen, die er gerade installiert.

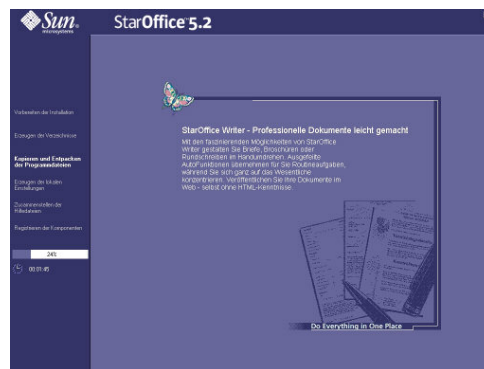


Abbildung 36: Restzeit und Installationskino

Am Ende der Installation zeigt der dann vollständig gefüllte Fortschrittsbalken kurzzeitig 100% an. Die Installation ist zu diesem Zeitpunkt jedoch noch nicht beendet. Für den weiteren Verlauf werden zunächst einmal das Uhr-Icon und die Zeit bis zum Ende der Installation nicht mehr angezeigt. Auch der einfarbige Fortschrittsbalken verschwindet. An seine Stelle tritt eine Animation, die scheinbar einen blau-schwarzen Farbverlauf von links nach rechts schiebt. Während dieser Zeit, die auf PCs mittlerer Leistung nahezu eine halbe Minute dauert, erhält der Benutzer keine weiteren Informationen über den Fortgang der Installation. Lediglich die Animation soll ihm vermitteln, dass der Installationsvorgang noch nicht beendet ist. Über den endgültigen Schluss der Installation informiert eine Meldung.

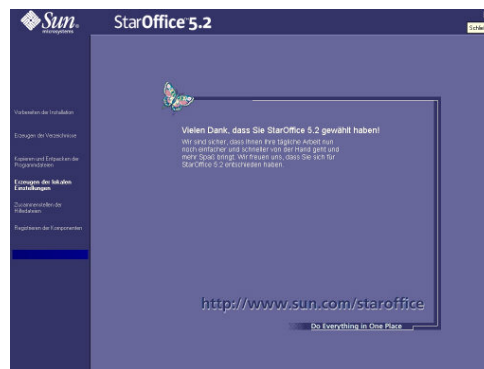


Abbildung 37: Kurz vor dem Ende der Installation (Farbe s. Anh.)

### dBASE 5.0

Bei der Installation von dBASE 5.0 wurde sehr viel Augenmerk auf ein optisch imposantes Installationskino gelegt. Wohl um dem Benutzer das Gefühl von Kontrolle zu geben, wurde der Installationsbildschirm als Blick vom Fahrersitz eines Autos gestaltet. Im Blickfeld sind eine Straße sowie eine Reklametafel am rechten Straßenrand, auf der während der Installation



Abbildung 38: Installationsbildschirm von dBase 5.0

neue Programmfunktionen vorgestellt werden, außerdem ein Tachometer und einige Texte auf dem Armaturenbrett. Die Tachonadel zeigt den bereits installierten Teil an, die Texte rechts daneben die Diskette sowie die gerade installierte Datei. Sämtliche Informationen werden im Hauptfenster angezeigt. Lediglich die Aufforderung an den Benutzer zum Einlegen einer neuen Diskette erscheint in einem separaten Dialogfenster.

### Thumbs Plus 4.01

Auch wenn das Installationsprogramm den kompletten Bildschirm in Anspruch nimmt, wird der Installationsfortschritt dennoch in einem separaten Dialog angezeigt, und der restliche Bildschirm bleibt ungenutzt. Der Installationsfortschritt wird auf mehrere Arten vermittelt. Zunächst versucht eine Animation, den Installationsvorgang zu visualisieren. Diese Animation erinnert auf den ersten Blick an die von Windows beim Kopieren, jedoch bewegt sich das Blatt Papier nicht von Ordner zu Ordner, sondern aus einem Pappkarton vor einem PC heraus. Je ein segmentierter Fortschrittsbalken zeigen den bereits erledigten Anteil für eine aktuelle Datei sowie für die gesamte Installation an. Über dem oberen Fortschrittsbalken steht der Name der Datei, die gerade installiert wird. Über dem unteren Fortschrittsbalken steht die für die restliche Installation benötigte Zeit. Die Grafik am linken Rand hat lediglich dekorativen Wert.

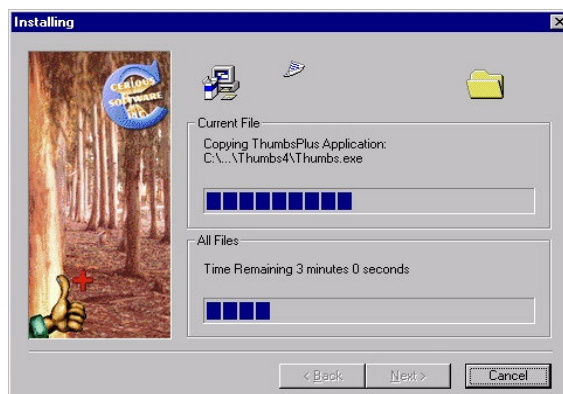


Abbildung 39: Installation von Thumbs Plus 4.01

### Sonique Player 1.6

Die Installation von Sonique erfordert zuerst die Eingabe einiger Parameter durch den Benutzer. Diese Eingabe geschieht in mehreren aufeinander folgenden Dialogfenstern. Nachdem alle für die Installation notwendigen Parameter eingegeben wurden, werden sie komplett auf einer Seite als Text ausgegeben. Der Benutzer hat hier die Möglichkeit, die Parameter noch einmal zu ändern.

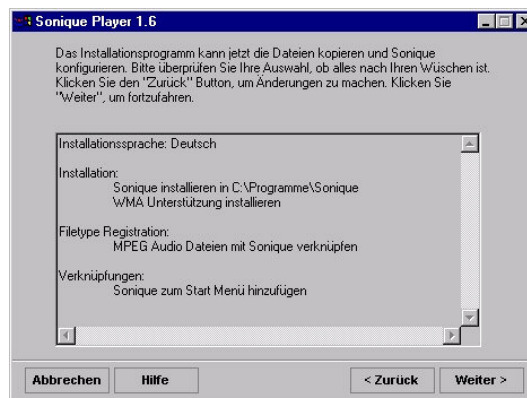


Abbildung 40: Installationsparameter bestätigen

Nach Bestätigung der Installationsparameter werden die Programmdateien kopiert. Sonique zeigt den Fortschritt bei der Installation durch zwei Fortschrittsbalken. Der obere Balken informiert über den gesamten Vorgang, der untere über die momentan kopierte Datei, deren Name darüber angezeigt wird. Da die zu installierenden Dateien verhältnismäßig klein sind, ist der untere Teil der Anzeige nicht lesbar, der Benutzer sieht lediglich ein Flackern. Numerische Angaben oder eine Information über die noch benötigte Zeit gibt es nicht.

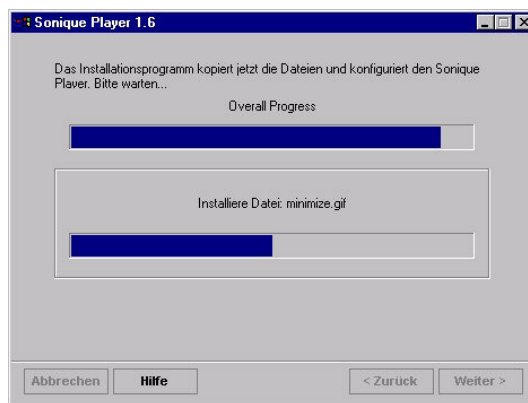


Abbildung 41: Kopieren der Dateien bei der Installation

### Microsoft Flugsimulator 2000

Der Microsoft Flugsimulator zeigt den Fortschritt während der Installation sowohl numerisch als auch mit Hilfe einer kreisförmigen Grafik an. Im Vorgriff auf das Spielgeschehen ist die Anzeige einem Flugzeuginstrument nachempfunden.



Abbildung 42: Installationsfortschritt

## Anwendungsprogramme

### Allgemeine Bedienelemente

In Anwendungsprogrammen vermitteln nicht nur spezielle Anzeigen Statusinformationen. Sehr viele Statusinformationen werden über Bedienelemente vermittelt.

Ein *Scrollbar* ist primär ein Bedienelement zum Navigieren im Dokument. Er liefert jedoch auch Informationen über die Position im Dokument und dessen ungefähre absolute Größe.

Abbildung 43 zeigt zwei Scrollbars von ver-



Abbildung 43: Scrollbar

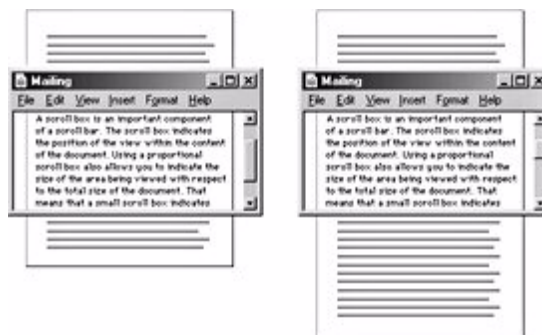


Abbildung 44: Scrollbar und Dokumentengröße ([MSDN, 2000/5], Figure 7.6)

chiedenen großen Dokumenten. Der linke Scrollbar gehört zu einem sehr langen Dokument. Der gezeigte

Ausschnitt, visualisiert durch die Größe des „Griffs“, ist weniger als 3% der Gesamtgröße<sup>1</sup>. Gezeigt wird ein Ausschnitt, der etwa am Ende des ersten Fünftels liegt, was durch die relative Position des „Griffs“ im Verschieberegion zu sehen ist. Der rechte Scrollbar gehört zu einem kleinen Dokument, das etwa dreimal so lang wie der gezeigte Ausschnitt ist. Abbildung 44 illustriert den Zusammenhang von Dokumentengröße und Scrollbar.

Menüs dienen zur Auswahl von Programmfunktionen. Sie zeigen aber auch die momentane Verfügbarkeit der jeweiligen Funktion an. Die ersten vier Punkte des in Abbildung 45 gezeigten Menüs sind inaktiv, die entsprechenden Funktionen stehen nicht zur Verfügung.

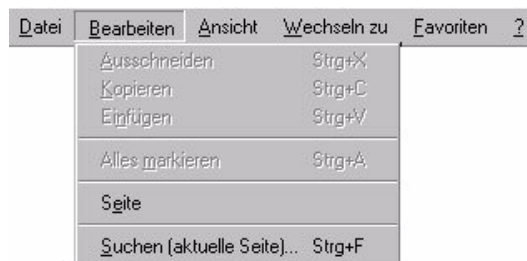


Abbildung 45: Aktive und inaktive Menüeinträge

Menüs können auch den Status von Objekten anzeigen. In Abbildung 46 bedeutet ein vorgestellter Haken, dass das zugehörige Element bereits auf dem Bildschirm zu sehen ist, ein erneuter Aufruf der Funktion die zugehörige Anzeige also ausschalten würde.



Abbildung 46: Menüeinträge informieren über den Status eines Objekts

Auch der aktuelle Zustand einer Checkbox oder einer Gruppe von Optionbuttons liefert Information über den Status der zugehörigen Funktion. In Abbildung 47, einem Dialog aus dem StarOffice 5.2, ist beispielsweise abzulesen, welche Einstellungen momentan für die ausgewählte Absatzvorlage gelten.

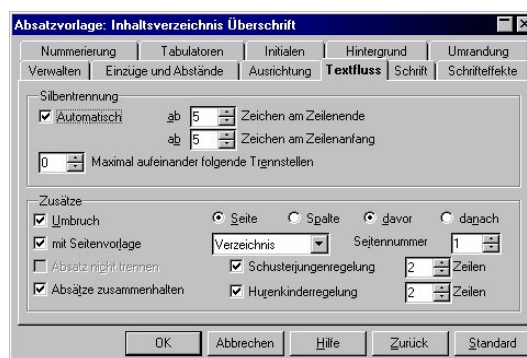


Abbildung 47: Checkboxes und Radiobuttons

<sup>1</sup> Aus praktischen Gründen wird der „Griff“ nicht beliebig klein gemacht, so dass ab einem bestimmten Verhältnis von gezeigtem Ausschnitt zu gesamtem Dokument nicht mehr sicher auf die Größe des gesamten Dokuments geschlossen werden kann. Diese minimale Größe des „Griffs“ ist hier erreicht, so dass nur noch eine Obergrenze für den gezeigten Ausschnitt bestimmt werden kann.



## Napster

Napster ist eine Musik-Tauschbörse im Internet. Benutzer bieten einander Musiktitel zum Download an. Je Benutzer sind also zwei Richtungen möglich: Download (obere Hälfte von Abbildung 48) und Upload (untere Hälfte). Downloads und Uploads unterscheiden sich nicht nur durch die Farbe des Fortschrittsbalkens (Blau für Downloads, Gelb für Uploads), sondern auch durch die Position auf dem Bildschirm. In einer separaten Spalte stehen die genauen numerischen Angaben zur Größe der Datei und dem bereits transferierten Teil. In der unteren linken Ecke des Bildschirms steht, ob der Benutzer mit dem Napster-Netzwerk verbunden ist oder nicht.

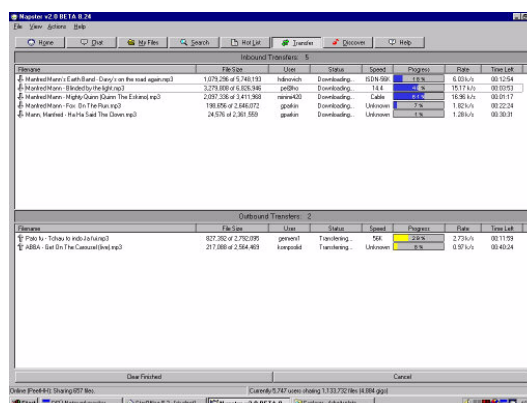


Abbildung 48: Napster (Farbe s. Anh.)

## CanoScan

CanoScan ist ein Treiberprogramm, das mit Flachbett-Scannern von Canon ausgeliefert wird. Das Scannen geschieht in zwei Stufen: Zuerst wird eine Vorschau eingescannt. Dieser Vorgang wird von Abbildung 49 gezeigt. Anschließend kann der Benutzer den genauen Ausschnitt wählen und mit den gewählten Einstellungen scannen. Der Fortschrittsbalken sowie die vom Benutzer für den Scan gewählten Einstellungen (weißer Kasten links) sind bei beiden Schritten identisch. Der Fortschritt bei der Vorschau wird jedoch „live“ angezeigt: Das Bild baut sich beim Einlesen sukzessive auf. Der Benutzer hat hier also im weitesten Sinne zwei Fortschrittsbalken, nämlich den herkömmlichen im separaten Dialog sowie die Vorschau selbst.

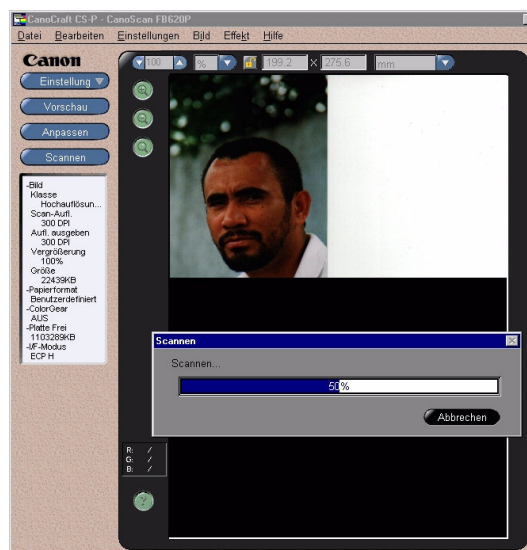


Abbildung 49: Vorschau beim Scannen

## Photoshop 5.0 LE

Photoshop ist ein Programmpaket zur Bildbearbeitung. Entsprechend viele modale Werkzeuge gibt es. Abbildung 50 zeigt die Werkzeugpalette. Das Icon in der Werkzeugpalette wird auch als Mauszeiger verwendet (vgl. Abbildung 51). Der Benutzer hat eine doppelte Rückmeldung darüber, welches Werkzeug verwendet wird: Zum einen die Form des Mauszeigers, zum anderen die gedrückte Schaltfläche in der Werkzeugleiste.



Abbildung 50:  
Werkzeugpalette



Abbildung 51:  
Verschiedene  
Mauszeiger  
(Pipette, Pinsel)

Photoshop arbeitet mit mehreren Bildebenen, die zu dem Gesamtbild zusammengesetzt werden. Abbildung 51 zeigt die Ebenenverwaltung. Photoshop verwendet auch hier grafische Symbole für die Sichtbarkeit von Ebenen (die mit dem Auge), die momentane Arbeitsebene (die mit dem Pinsel) oder miteinander verbundene Ebenen (die mit der Kette und die darunter liegende). Der Schieberegler stellt die Transparenz der markierten Ebene ein, die Listbox die Kombinationsmethode der Ebene mit den darunter liegenden.



Abbildung 52: Ebenenverwaltung

## StarOffice 5.2

StarOffice 5.2 ist ein integriertes Büropaket. Es enthält unter anderem Textverarbeitung, Tabellenkalkulation, Präsentation, Pixel- und Vektorgrafik sowie einen E-Mail-Client.

Viele Statusinformationen



Abbildung 53: StarOffice Statusleiste

werden in einer speziell dafür vorgesehenen Statusleiste angezeigt. Abbildung 53 zeigt die Statusleiste für die Textverarbeitung. Sie enthält die aktuelle Seitennummer und die gesamte Seitenzahl, den Namen der verwendeten Seitenvorlage, den Zoomfaktor, den Modus der Texteingabe (Einfügen oder Überschreiben), den Selektionsmodus, den Status von Hyperlink ausführen/bearbeiten. Jede Statusinformation ist

gleichzeitig ein verstecktes Bedienelement. Ein Doppelklick auf die Seitenzahlen beispielsweise öffnet oder schließt den Navigator (ein separates Fenster für die Navigation im Dokument), ein einfacher Klick wechselt zwischen Einfügen und Überschreiben. Die Statusleisten für die anderen Module von StarOffice sind ähnlich aufgebaut.

StarOffice macht sehr viel Gebrauch von sogenannten Werkzeugleisten, permanent sichtbaren Sammlungen grafischer Bedienelemente. Eine Werkzeugleiste enthält in der Regel Schaltflächen für den schnellen Zugriff auf häufig genutzte Funktionen. Diese Schaltflächen zeigen an, ob die zugehörige Funktion verfügbar ist oder nicht: Eine momentan nicht verfügbare Funktion wird grau, eine verfügbare Funktion farbig dargestellt (s. Abbildung 54). Schaltflächen können noch auf drei weitere Arten genutzt werden:



Abbildung 54: Aktive und inaktive Buttons

- Als *grafische Checkboxes*. Ein Klick schaltet einen Modus ein- oder aus. Ein eingeschalteter



Abbildung 55: Eine Gruppe grafischer Checkboxes

Modus wird durch eine gedrückte Schaltfläche visualisiert. Abbildung 55 zeigt eine Gruppe von drei grafischen Checkboxes.

- Als *grafische Optionbuttons*. Mehrere Schaltflächen gehören zur selben Funktion.



Abbildung 56: Grafische Optionbuttons

Die gewählte Schaltfläche ist gedrückt, die anderen nicht. Abbildung 56 zeigt vier grafische Optionbuttons mit jeweils einem anderen gewählten Modus.

- Zum Aufruf einer Funktion, die einen Modus setzt, wobei die Schaltfläche den gewählten Modus anzeigt. Abbildung 57 zeigt zwei Schaltflächen, die einen Dialog zum Auswählen der Schriftfarbe bzw. der Farbe für die Hervorhebung aufrufen. Das Aussehen der Schaltfläche ändert sich und zeigt die aktuell gewählte Farbe an.



Abbildung 57: Buttons für Farbe von Schrift und Hintergrund

Auch mit Hilfe von Listboxen kann Status dargestellt werden. In Abbildung 58 werden die momentan gewählte Formatvorlage sowie die Schrift und ihre Größe angezeigt. Zum Ändern kann der Benutzer einen Eintrag aus der Liste wählen; dabei erhält er auch gleich eine Übersicht der Alternativen.

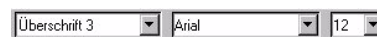


Abbildung 58: Schriftattribute in Listboxen



StarOffice zeigt während des Navigierens mit dem Scrollbar eine Tipphilfe an, um dem Benutzer eine genaue Information über die Position im Dokument zu geben. Bei der Textverarbeitung beispielsweise werden Seitennummer, Seitenzahl sowie Nummer und Name des Kapitels angezeigt.

Seite 8 13 / 149 · 2.2 Definition von Status und Statusanzeigen

Abbildung 59: Scrollbar mit zusätzlicher textueller Informationen über die Position im Dokument

Ein Posteingangskorb zeigt die Anzahl der enthaltenen E-Mails an. Falls der Eingangskorb ungelesene Mails enthält, wird er durch fette Schrift hervorgehoben, und die Anzahl der noch nicht gelesenen E-Mails wird angezeigt.

Mail (ivx) (2/138)  
Mail (Student) (1)

Abbildung 60: Verschiedene Posteingangskörbe

Ein geöffneter Posteingangskorb zeigt eine Liste der enthaltenen E-Mails. Ein Dreieck

Autor	Titel			Erstellungsdatum	Priorität
Peter Thielmann	Markierte Mail			15.01.01, 15:18:06	3 (Normal)
<b>Peter Thielmann</b>	<b>Ungelesene Mail</b>			<b>15.01.01, 15:1...</b>	<b>3 (Normal)</b>
Peter Thielmann	Gelesene Mail			15.01.01, 15:17:39	3 (Normal)

Abbildung 61: Inhalt eines Posteingangskorbs (Farbe s. Anh.)

in einem Spaltenkopf zeigt die Sortierung an. Für gelesene und nicht gelesene sowie für markierte und nicht markierte E-Mails gibt es jeweils eigene Icons, die zu Beginn der Zeile stehen. Nicht gelesene E-Mails sind zusätzlich durch fette Schrift hervorgehoben. In je einer separaten Spalte befinden sich Icons für ungelesene bzw. für markierte E-Mails. Die momentan angezeigte E-Mail ist durch einen gestrichelten Rahmen gekennzeichnet.

StarOffice bietet einen Fortschrittsbalken für die Makroprogrammierung an. Es handelt sich um einen segmentierten, von links nach rechts laufenden Balken, der stets die komplette Statuszeile in Anspruch nimmt. Der Makroprogrammierer kann im linken Teil einen Text ausgeben lassen, der Balken wird entsprechend verkürzt. Der Text kann zwischendurch geändert werden.



Abbildung 62: Fortschrittsbalken mit und ohne Text

Abbildung 63 zeigt eine Gruppe von bildhaft angeordneten Optionbuttons. Ihre Anordnung entspricht der jeweiligen Ausrichtung der Textverankerung. Der Benutzer muss keine langen Texte lesen, sondern sieht auf einen Blick, wie der Text verankert ist. Das Mapping unterstützt den Benutzer auch beim Bilden eines mentalen Modells der Textverankerung.



Abbildung 63: Optionbuttons mit bildhaftem Mapping

Neben Standardbedienelementen wie beispielsweise Optionbuttons oder Checkboxes hat StarOffice auch einige innovative grafische Bedienelemente, beispielsweise den

Kreis zum Einstellen der Schreibrichtung. Der Benutzer kann entweder auf herkömmliche Art einen numerischen Wert in das Eingabefeld schreiben oder er klickt auf den Rand des Kreises. In einem Raster von 15° wird damit die Schreibrichtung eingestellt. Das Ergebnis wird durch die Position des Kreises angezeigt, der alle 45° Gelb, ansonsten Dunkelblau ist.



Abbildung 64: Grafisches Bedienelement (Farbe s. Anh.)

### WS\_FTP95LE

WS\_FTP95LE ist ein Programm zum Übertragen von Dateien auf oder von FTP-Servern. Während der Übertragung von Dateien wird in einem separaten Dialog der Fortschritt angezeigt. Der prozentuale Fortschritt wird durch einen nicht segmentierten Balken angezeigt. Vier numerische Informationen ergänzen die Grafik: Der bereits übertragene Prozentsatz, die Größe der zu übertragenden Datei, die Menge der bereits übertragenen Daten sowie die durchschnittliche Geschwindigkeit der Datenübertragung.

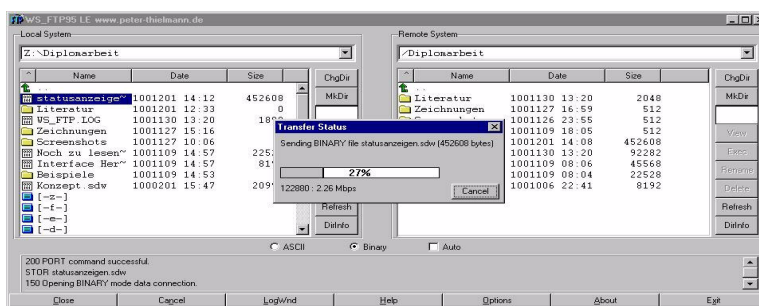


Abbildung 65: WS\_FTP95LE beim Übertragen von Dateien

Das Ende der Übertragung wird zusätzlich durch ein aus etwa einem Dutzend Tönen bestehendes Signal angezeigt. WS\_FTP95LE zeigt außerdem in den drei Zeilen oberhalb der Schaltflächen ein Log, das auch Statusinformationen enthält. Zu sehen sind immer nur die letzten drei Einträge, jedoch kann der Benutzer hochscrollen und so ältere Einträge lesen.

Die Anzeige des Übertragungsfortschritts durch einen separaten Dialog kann in den Optionen ausgeschaltet werden. Der Benutzer wird dann lediglich durch einen numerischen Wert über den Fortschritt informiert, der unterhalb des Quellfensters hochgezählt wird. Abbildung 66 zeigt die Übertragung einer Datei vom Server auf den lokalen Rechner.

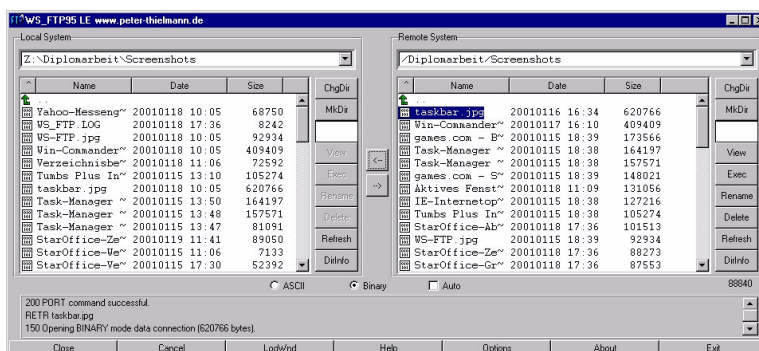


Abbildung 66: Übertragen von Dateien ohne separate Anzeige

**Fetch**

Fetch ist ein Programm auf dem Macintosh zum Übertragen von Dateien auf oder von FTP-Servern. Der Status der Datenübertragung wird zunächst einmal durch eine kreisförmige Fortschrittsanzeige angezeigt. Der Kreis wird kontinuierlich im Uhrzeigersinn gefüllt. Zusätzlich werden die Größe der zu übertragenden Datei sowie der bereits übertragene Teil, die Download-Geschwindigkeit sowie die Restzeit numerisch angezeigt. Auch der Mauszeiger wird für die Vermittlung von Statusinformationen genutzt: Zunächst einmal nimmt er – wohl als Assoziation zum Apportieren – die Form eines Hundes an (links neben dem Kreis). Während eines Downloads bewegt sich der Hund von rechts nach links, während eines Uploads von links nach rechts.

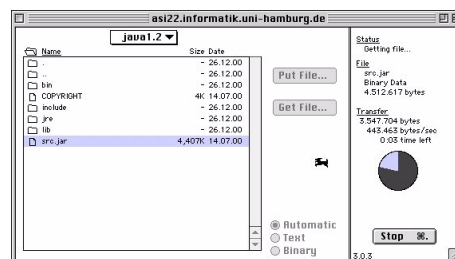


Abbildung 67: Fetch beim Download einer Datei

**Internet Explorer 5.5**

Viele Statusinformationen werden in einer speziell dafür vorgesehenen Statusleiste am unteren Fensterrand angezeigt. Interessant ist der linke Bereich, der kontextabhängig verschiedene Informationen enthalten kann:

- Die URL der gerade zu ladenden Seite (Abbildung 68 a) sowie einen kleinen, nicht segmentierten Fortschrittsbalken,
- die Anzahl der noch fehlenden Objekte (z.B. noch nicht geladene Grafiken, Abbildung 68 b) und einen Fortschrittsbalken,
- die Information, dass die angezeigte Seite komplett geladen ist (Abbildung 68 c),
- ein potenzielles Sprungziel, falls der Mauszeiger über einen Link bewegt wird (Abbildung 68 d).

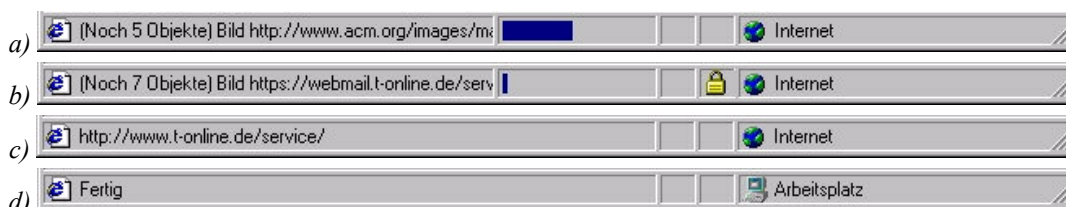


Abbildung 68: Statusleiste zu verschiedenen Zeitpunkten

Auch der Internet Explorer hat Symbolleisten. Abbildung 69 zeigt die Symbolleiste für Standardschaltflächen.



Abbildung 69: Werkzeugleiste des Internet Explorers

Microsoft hat hier drei verschiedene Darstellungen implementiert: Graues,

eingraviertes Bild mit grauer, eingravierter Schrift für inaktive Buttons sowie graues Bild oder farbiges Bild mit schwarzer Schrift für aktive Buttons. Während im Normalfall das Bild grau ist, wird es farbig, wenn der Mauszeiger sich über ihm befindet. Gleichzeitig wird die Schaltfläche durch dreidimensionale Darstellung erkennbar.

## Opera 5.02

Opera ist ein weiterer Browser. Mit größerem Abstand nach den Marktführern Microsoft Internet Explorer und Netscape Navigator ist er der am dritthäufigsten verbreitete Browser. Während sich der Internet Explorer und der Navigator nur in kleinen Details unterscheiden, geht Opera bei der Gestaltung der Benutzeroberfläche eigene Wege. Das macht sich auch bei den Statusanzeigen bemerkbar. Opera hat die Statusleiste oberhalb der eigentlichen Daten. Auch hier gibt es einen Fortschrittsbalken, der allerdings auch einen numerischen Wert anzeigt. Es folgen die Anzahl der geladenen Bilder sowie aller Bilder im Dokument. Im nächsten Feld bewegt sich die blaue Markierung in einer Schleife von links nach rechts. Die Tipphilfe beschreibt das Feld mit „Zeigt den Ladefortschritt und wieviel bereits geladen wurde“. Es folgen die Ladegeschwindigkeit sowie die bereits verstrichene Zeit.

Ein weiterer Unterschied zu den Platzhirschen wird beim Download von Dateien sichtbar. Während der Internet Explorer und der Navigator für jeden Download einen separaten Dialog öffnen, erstellt Opera eine neue Seite, auf der sämtliche Downloads gelistet sind. Die Liste enthält für jeden Eintrag den Dateinamen, die Größe der Datei, Informationen über den Fortschritt, die noch benötigte Zeit, die Übertragungsgeschwindigkeit und den Typ der Datei. Der Fortschritt wird für laufende Downloads als segmentierter Balken mit numerischem Wert angezeigt. Beendete oder angehaltene Downloads haben keinen Fortschrittsbalken, hier wird eine textuelle Statusmeldung ausgegeben. In einem Feld unterhalb der Liste werden für einen ausgewählten Eintrag der vollständige Quellpfad, der vollständige Zielpfad sowie die Dateigröße und der bereits geladene Anteil angezeigt. Die Rest-

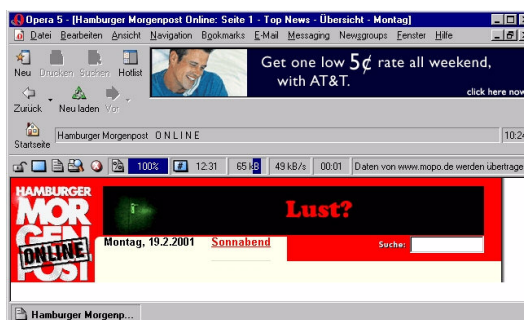


Abbildung 70: Opera beim Laden einer Seite

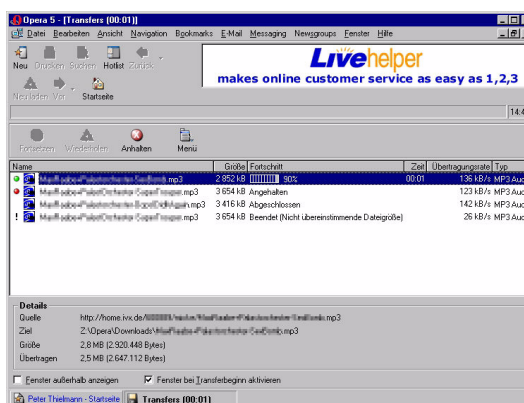


Abbildung 71: Download von Dateien (Farbe s. Anh.)

zeit wird außerdem in der Titelzeile des Hauptfensters sowie in einer Schaltfläche zum Wechseln zwischen verschiedenen Fenstern im Browser angezeigt. Unter Windows wird sie auch in der Schaltfläche der Task-Leiste angezeigt, jedoch ist der dort zur Verfügung stehende Platz zu knapp, um die Information vollständig lesen zu können.

### Noris Bank Homebanking-Software

Homebanking-Software erfordert besondere Sicherheitsmaßnahmen. Die Homebanking-Software der Noris Bank schließt sich automatisch, wenn der Benutzer sie längere Zeit nicht benutzt hat. Um dem Benutzer mitzuteilen, wie lange er schon keine Eingabe mehr getätigt hat, wird während inaktiver Zeiten eine kreisförmige Grafik in zwölf Schritten gefüllt. Die ersten neun Segmente sind grün, die letzten drei gelb. Eine numerische Angabe fehlt auf dem Bildschirm, so dass eine Schätzung der Restzeit nur schwer bis gar nicht möglich ist.

Der Benutzer erfährt nur aus schriftlichen Unterlagen zum Konto oder aus der Hilfe, nach welcher Zeit sich die Anwendung automatisch schließt. Auch der grobe Kontostand ist auf einen Blick zu erkennen: Haben wird in Dunkelblau, Soll in Hellrot dargestellt.



Abbildung 72: Hinweis auf Zeit ohne Benutzereingaben (Farbe s. Anh.)

### Sonique

Sonique ist ein Audio-Player, der sich offensichtlich an eine jüngere Zielgruppe wendet. Der Player kann in drei verschiedenen Größen mit jeweils anderem Funktionsumfang angezeigt werden. Ein Merkmal, das Sonique mit anderen vergleichbaren Programmen teilt, ist die Möglichkeit, das Aussehen des

Programms mit Hilfe sogenannter Skins an eigene Vorlieben anzupassen. Abbildung 73 zeigt die mittlere Größe von Sonique mit zwei verschiedenen Skins; der linke Skin ist dabei der installierte Standard. Auch wenn der erste Eindruck es nicht nahe



Abbildung 73: Sonique mit verschiedenen Skins (Farbe s. Anh.)



legt, haben die beiden Versionen die selben Bedienelemente und Anzeigen. Alphanumerisch angezeigt werden der Name des gerade gespielten Titels, seine Position in der Playlist<sup>1</sup> sowie die bereits verstrichene Zeit. Eine Grafik in Form einer Mischung aus Lichtorgel und Oszilloskop zeigt, dass momentan Musik gespielt wird. Interessant sind zwei den Status anzeigende Bedienelemente. Am einfachsten als Bedienelement zu erkennen ist der graue Drehregler in der linken Grafik. Mit ihm wechselt man die Titel. Die Position der „Griffmulde“ zeigt die Position des gerade gespielten Titels in der Playlist an und gibt damit quasi einen Überblick über den gesamten Vorgang. Der Kreis farbiger Punkte um die alphanumerische Anzeige ist verhältnismäßig leicht als Fortschrittsanzeige für den gerade laufenden Titel erkennbar. Ein Klick auf eine beliebige Position setzt die Wiedergabe an der entsprechenden Stelle fort. In der rechten Grafik sind diese beiden Elemente in den beiden Quadranten in der Mitte realisiert. Das schwarze Segment im äußeren Quadrat zeigt den Fortschritt für den Gesamtvorgang, das rote im inneren Quadrat den bereits gespielten Teil des aktuellen Titels.

## WinAmp

WinAmp ist ein weiterer Audio-Player. Er kann in verschiedenen Größen dargestellt werden, um entweder ständig geöffnet bleiben zu können, ohne viel Bildschirmfläche in Anspruch zu nehmen, oder sämtliche Informationen anzuzeigen. In der größten Darstellung gibt es unter anderem einen Lautstärkereglер, einen Balanceregler sowie einige zusätzliche Informationen und Bedienelemente. Der Lautstärkereglер – direkt rechts neben dem Display – weist eine Besonderheit auf: Je nach gewählter Lautstärke verfärbt sich die Bahn, auf der er geschoben wird. Bei niedriger Lautstärke ist sie Grün, bei mittleren Lautstärken wechselt die Farbe nach Gelb, und bei hohen Lautstärken nach Rot – anscheinend wird versucht, so vor ungesunder hoher Lautstärke zu warnen. Der Übergang zwischen den Farben ist fließend. Die Playlist zeigt den momentan gespielten Titel mit weißer fetter Schrift, die übrigen Titel sind mit grüner Schrift eingetragen. Zusätzlich kann ein beliebiger Titel markiert werden, um beispielsweise dessen Eigenschaften zu ändern. Der markierte Titel ist mit einem blauen Balken hinterlegt.



Abbildung 74: WinAmp mit Playlist  
(Farbe s. Anh.)

<sup>1</sup> Die Playlist ist eine geordnete Liste der abzuspielenden Titel. Sie kann mit der vom Benutzer programmierten Auswahl von Titeln einer CD verglichen werden.

## WinOnCD

WinOnCD ist ein Programm zum Beschreiben von CD-ROMs. Abbildung 75 zeigt einen Dialog, der beim Schreiben von Daten auf die CD-ROM erscheint. Falls wie in diesem Fall keine beschreibbare CD eingelegt ist, wird diese Tatsache dem Benutzer als Status mitgeteilt. Der Dialog schließt sich selbsttätig nach 30 s. Ein Countdown (direkt über den Schaltflächen) zeigt die bis dahin noch verbleibende Zeit an, getreu der Forderung von [Tognazzini, 1992] (Seite 217).

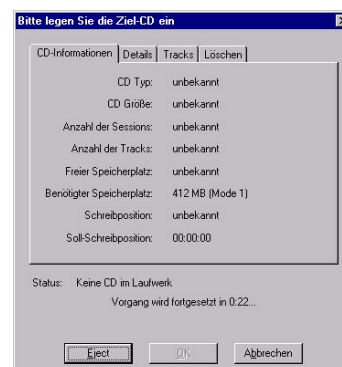


Abbildung 75: Selbsttätig schließender Dialog

## AOL 5.0

AOL gibt einige Statusmeldungen als gesprochene Texte aus („Sie haben Post“).

## Yahoo! Messenger

Der Yahoo! Messenger ist ein Programm zur Kommunikation über das Internet. Der Benutzer kann mit anderen Yahoo!-Teilnehmern kommunizieren, falls diese gerade online sind. Das Hauptfenster (Abbildung 76) zeigt die Liste aller Freunde des Benutzers sowie ihren momentanen Status, ebenso seinen eigenen Status. Der exakte Status von Freunden wird textuell wiedergegeben (siehe *berend\_cornelius* oder *k\_steele*). Icons informieren jedoch zusätzlich über den groben Status (offline, online oder online, aber momentan nicht erreichbar). Der eigene Status wird überwiegend textuell dargestellt („Ich bin verfügbar“, „Verbunden“). Über den Status der Verbindung informiert jedoch zusätzlich noch eine simulierte LED, die entweder rot oder grün leuchtet.



Abbildung 76: Yahoo! Messenger (Farbe s. Anh.)

## Spiele

Von Computerspielen kann eine Menge über die Gestaltung der Benutzungsschnittstelle gelernt werden, insbesondere, weil bei ihnen höhere Anforderungen an die Benutzungsschnittstelle gestellt werden als bei anderen Anwendungsprogrammen. Bei einer Textverarbeitung beispielsweise wird die Benutzungsschnittstelle nicht absolut betrachtet, sondern im Vergleich mit anderen Textverarbeitungen. Der Benutzer

einer Textverarbeitung hat die Wahl, entweder mit einem Programm zu arbeiten oder Papier und Bleistift zu nehmen. Er ist zufrieden, wenn die Textverarbeitung seine Arbeit deutlich erleichtert. Ein Computerspieler hingegen muss keine Aufgabe erledigen, sondern benutzt das Programm zum Vergnügen. Wenn die Benutzungsschnittstelle umständlich oder verwirrend ist, wird das Spiel wieder eingepackt. Die Benutzungsschnittstelle muss hier also einen absoluten Test bestehen. Sie muss nicht nur funktional sein, sondern zusätzlich noch Spaß machen. Computerspiele waren von jeher diejenigen, die neue akademische Konzepte der Benutzungsschnittstelle zuerst in kommerziellen Produkten eingesetzt haben ([Crawford, 1991], Seite 103). Vielleicht ist also das Fehlen wirklich innovativer Konzepte für Statusvermittlung bei Computerspielen ein Indiz dafür, dass auf diesem Gebiet zur Zeit keine bahnbrechenden Neuerungen zu erwarten sind. Insgesamt bleibt festzustellen, dass Feedback und damit auch Statusvermittlung bei Spielen weniger abstrakt, sondern eher liebevoll bildlich mit Bezug zum Thema gestaltet wird.

### Online-Spiel *Battleship* bei games.com

Das Online-Spiel *Battleship* ist eine elektronische Variante des Klassikers *Schiffe versenken*. Zwei Spieler platzieren ihre Flotte auf einem 10x10 Kästchen großen Spielfeld und versuchen zu erraten, wo der Gegenspieler seine Schiffe versteckt hat. In Abbildung 77 fällt zu-

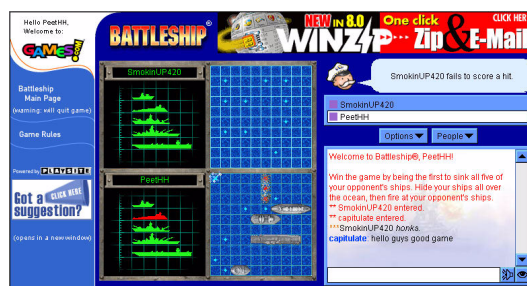


Abbildung 77: Online-Spiel Battleship (Farbe s. Anh.)

nächst einmal der Bereich auf, wo die Flotten der beiden Spieler zu sehen sind. Jeder Spieler hat fünf Schiffe. Noch nicht entdeckte, getroffene und versenkte Schiffe sind durch unterschiedliche Farben dargestellt (Grün, Orange, Rot). Rechts daneben sind die Spielfelder. Hier werden die erfolglosen Rateversuche (blaue Punkte) und eventuelle Treffer (orangerote Punkte) notiert. Der aktuelle Schuss wird zunächst animiert dargestellt, im unteren Spielfeld in der rechten oberen Ecke ist noch der Rest der Animation zu sehen. Zusätzlich wird ein Geräusch abgespielt, das entfernt an einen ins Wasser fallenden Gegenstand erinnert. Welcher Spieler an der Reihe ist, erzählt einem zum einen der Seebär, zum anderen ist der entsprechende Name markiert. Außerdem ist das aktuelle „Kampfgebiet“ durch ein betontes Gitter hervorgehoben. Die Farbe des Quadrats links neben dem Namen des Spielers zeigt seine Spielstärke. Wird der Mauszeiger über den Namen des Spielers bewegt, zeigt eine Tipphilfe den genauen Wert.



## Minesweeper

Minesweeper ist eines der in Windows enthaltenen Spiele. Der Spieler muss eine bestimmte Anzahl von Minen finden. Dazu bekommt er Hinweise, wie viele der benachbarten Felder Minen enthalten. Diese Hinweise werden durch verschiedenfarbige Zahlen gegeben. Felder mit Minen zeigen durch ein Symbol ihren Status an. Die Informationen über dem Spielfeld zeigen die noch zu findenden Minen und die bereits verstrichene Zeit an.



Abbildung 78: Minesweeper

## Microsoft Flugsimulator 2000

Die Statusinformationen, die der Flugsimulator vermittelt, gehören allesamt zum Spielgeschehen. Daher sind sie konsequent auch passend dazu als Simulation eines Flugzeugcockpits gestaltet.



Abbildung 79: Cockpit des Flugsimulators

## Fritz 3.52, das sprechende Schachprogramm

Fritz ist nach eigener Werbung eines der stärksten Schachprogramme auf dem Markt. Vermutlich versucht es deshalb, mit den vielfältigen technischen Informationen in der rechten unteren Ecke Eindruck zu schinden. Ein Coach hilft dem Spieler durch Hinweise in kritischen Situationen. Diese sind sowohl optischer Natur wie in Abbildung 80 als auch gesprochene Texte: Fritz informiert den Spieler mit humorigen Texten über den Stand des Spiels sowie kritische Situationen.



Abbildung 80: Fritz

## Online

Moderne Multimedia-Techniken wie Flash ermöglichen es, immer umfangreichere Projekte im Internet zu realisieren. Zeichentrickfilme von mehreren Minuten Länge sind keine Seltenheit mehr, grafisch immer aufwändiger gestaltete Menüsysteme sind noch häufiger zu finden. Immer umfangreichere Inhalte bedingen zwangsläufig jedoch immer längere Ladezeiten. Glücklicherweise kann mit den Techniken, deren Einsatz die lange Ladezeit verursacht, auch eine Ladeanzeige gestaltet werden.

Ebenso wie das Spielen, so ist auch der Besuch vieler Web-Auftritte eine freiwillige Sache mit dem alleinigen Wunsch, unterhalten zu werden. Web-Designer müssen daher ein verstärktes Augenmerk darauf legen, den Benutzer zu unterhalten – wenn er sich langweilt, ist er schnell wieder verschwunden. Insbesondere die teilweise sehr lange Ladezeit bei umfangreicheren Multimedia-Angeboten im Web ist ein Problem und könnte interessante Lösungen für Fortschrittsanzeigen hervorbringen. In Monatszeitschriften für Grafikdesigner wie beispielsweise in der PAGE ([Gerdes, 2001], Seiten 32-35) werden bereits verschiedene Methoden vorgestellt, wie der Web-Designer den Benutzer während der Ladezeiten beschäftigen kann. Gezeigt werden verschiedene Animationen wie beispielsweise trommelnde Hände, „gestylte“ herkömmliche Fortschrittsbalken, Fortschrittsanzeiger mit Bezug zur Website wie beispielsweise eine sich füllende Weinflasche bei einem Weingut oder auch Informationen über die Website, beispielsweise Bedienhinweise oder eine Sitemap.

### Icebox

Icebox ([Icebox, 2000]) bietet diverse mit Flash realisierte Zeichentrickfilme an. Die Fortschrittsanzeigen sind teilweise sehr schlicht wie bei *Zombie College* (Abbildung 81), teilweise sind sie grafisch etwas aufwändiger gestaltet und passen zum Inhalt des Films

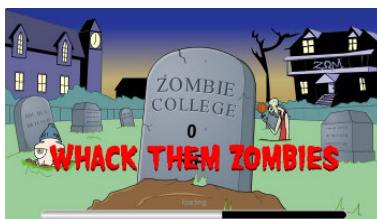


Abbildung 81: Ladevorgang von *Zombie College* (Farbe s. Anh.)



Abbildung 82: Ladevorgang von *Queer Duck* (Farbe s. Anh.)



Abbildung 83: Ladevorgang von *Poker Night* (Farbe s. Anh.)

wie beispielsweise bei *Poker Night* (Abbildung 83), wo eine abbrennende Zigarre den Fortschritt anzeigt. Eine inhaltliche umfangreichere Fortschrittsanzeige gibt es bei *Queer Duck* (Abbildung 82). Hier wird der Fortschritt zusätzlich in Zehnerschritten numerisch angezeigt, außerdem zeigt eine Endlosanimation<sup>1</sup> dem Benutzer, dass das System nicht untätig ist. Zusätzlich ertönt bei bestimmten Prozentwerten ein Gong, und kurz vor Schluss ertönen drei Gongs schnell hintereinander, ähnlich wie im Theater, wo zum Ende der Pause geläutet wird. Interessant, wenn auch für diese

<sup>1</sup> Beim Wort *LOADING* wird der Reihe nach ein Buchstabe vergrößert gezeigt. In der Abbildung ist es das *I*.

Arbeit unwichtig, ist, dass nach dem Laden von etwa 30% eines jeden Films ein Spiel gespielt werden kann, um die subjektive Wartezeit zu verkürzen. Die 0 auf dem Grabstein in Abbildung 81 beispielsweise zeigt den Spielstand an.

### Jacob's Creek

Jacob's Creek ist ein australisches Weingut. Die Ladezeit der Website ist dank ausgiebiger Nutzung von Multimediatechniken recht lang. Während der Wartezeit wird eine Weinflasche analog zum Fortschritt beim Laden gefüllt. Gleichzeitig dreht sich das Wort *Loading* um die Flasche herum.



Abbildung 84: Fortschrittsanzeige in Form einer sich füllenden Weinflasche für die Website eines Weinguts (Farbe s. Anh.)

### Palastorchester

Der Web-Auftritt des deutschen „Palastorchesters“ und seinem Sänger Max Raabe ([Palastorchester, 2001]) enthält auch einige Hörproben. Im Gegensatz zu den sonst üblichen einfachen Links, die einen externen Player starten, ist hier mit Flash eine spezielle Anwendung realisiert. Ein dem berühmten Nipper ähnlicher Hund fängt die gewählte Schallplatte, legt sie auf das Grammophon und läuft zum rechten Rand, während die Musik spielt. Pünktlich zum Ende der Musik ist er dort angekommen.

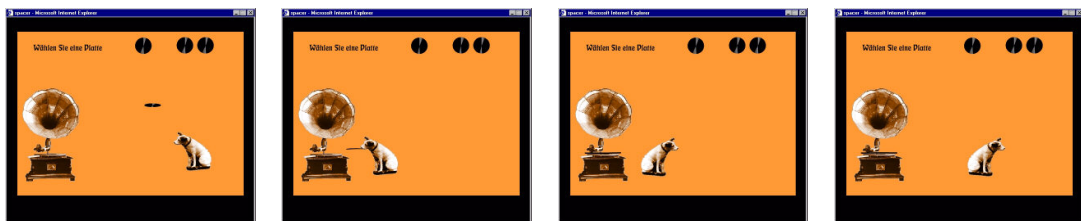


Abbildung 85: Abspielen einer Hörprobe vom Palastorchester

## 5.4 Auswertung der gezeigten Beispiele

Folgende spontane Fragen können zur Einteilung von Statusanzeigen gestellt werden:

- Sind sie abstrakt oder bildlich?
- Handelt es sich um ein Bedienelement, das über den Status der einzustellenden Funktion informiert, handelt es sich um eine Anzeige mit der Möglichkeit, Bedienelemente aufzurufen, oder handelt es sich um eine reine Anzeige?
- Werden die Informationen numerisch, textuell oder grafisch präsentiert?

- Werden diskrete Zustände oder ein kontinuierlicher Prozess angezeigt?
- Wie schnell ändert sich die präsentierte Information?
- Sind die präsentierten Informationen permanent oder vergänglich?
- Ist die Statusanzeige ständig präsent, nur nach Anforderung oder nur nach dem Eintreten von nicht durch den Benutzer veranlassten Ereignissen?
- An welcher Position befindet sich die Statusanzeige?

Diese Fragen behandeln zwei Aspekte: *Was* wird dargestellt, und *wie* wird es dargestellt. Für den Benutzer ist zuerst einmal der Inhalt, also die vermittelte Information, relevant. Ich beginne die Auswertung der Beispiele daher unter inhaltlichen Gesichtspunkten und werte anschließend aus, mit welchen Mitteln und wo die Informationen dargestellt werden.

Gestaltete Statusanzeigen bei der Hardware sind fast ausschließlich LEDs, eine Auswertung würde keine wesentlichen Erkenntnisse bringen. Daher untersuche ich nur die Statusanzeigen in der Software.

### **Auswertung unter inhaltlichen Aspekten**

Unter inhaltlichen Aspekten können die angezeigten Informationen erst einmal, analog zur Definition von Status in Kapitel 2.2, *Definition von Status und Statusanzeigen*, in statische Informationen und dynamische Informationen unterteilt werden.

Eine Betrachtung unter inhaltlichen Gesichtspunkten kann mit oder ohne Berücksichtigung der Bedeutung der Information im Rahmen der Anwendung erfolgen. Wird die Bedeutung im Rahmen der Anwendung nicht berücksichtigt, spreche ich von einer Betrachtung unter syntaktischen Gesichtspunkten. Hier interessiert lediglich der dargestellte Wert, beispielsweise eine Zahl. Die Betrachtung mit Berücksichtigung der Bedeutung, beispielsweise die Nutzung einer Ressource, ist eine Betrachtung unter semantischen Gesichtspunkten. „Low-Level-Semantik“, insbesondere das Interpretieren eines numerischen Werts als physikalische Größe wie beispielsweise Geschwindigkeit, gehört für mich noch nicht zur Bedeutung im Rahmen der Anwendung. Der Einfachheit halber werde ich dieses zusammen mit der Syntax betrachten.

### **Statische Informationen**

Unter Berücksichtigung semantischer Aspekte können die in den Beispielen gefundenen statischen Statusanzeigen einer der folgenden Gruppen zugeordnet werden:

- Eigenschaften von Objekten
- Position in Dokumenten
- Modi eines Programms
- Parameter von Programmfunktionen, inkl. Eintreten eines Ereignisses
- Parameter von Prozessen
- Ressourcennutzung

Bei Betrachtung syntaktischer Aspekte ohne Berücksichtigung der Semantik können die in den Beispielen gefundenen statischen Statusinformationen einer der folgenden Gruppen zugeordnet werden:

- Genau einer aus einer endlichen (ungeordneten) Menge von Zuständen (mit dem Spezialfall *binäre Zustände*, wo die Zustandsmenge zwei Elemente enthält)
- Numerische Werte<sup>1</sup>
  - Absolute oder relative Zahlen
  - Diskret oder kontinuierlich
  - Interpretation als Menge, Größe, Zeit, Geschwindigkeit
- Beliebige Texte
- Räumliche Information

Semantische und syntaktische Aspekte sind nicht unabhängig voneinander. Die folgende Tabelle zeigt, welche Kombinationen bei den Beispielen zu finden sind:

Semantik	Syntax	Menge von Zuständen	Numerisch	Beliebiger Text	Räumlich
Eigenschaften von Objekten		X	X		
Position in Dokumenten			X	X	X
Modi eines Programms		X			
Parameter von Funktionen		X	X	X	
Parameter von Prozessen			X	X	
Ressourcennutzung		X	X	X	

Tabelle 5: Gefundene Kombinationen von Syntax und Semantik

<sup>1</sup> Aus Gründen der Übersichtlichkeit habe ich nicht alle Alternativen ausformuliert, sondern Unter-aspekte gebildet. Die vollständige Liste entsteht durch Kombination der Parameter aller Unter-aspekte.

### **Dynamische Informationen**

Die Anzeigen dynamischer Statusinformationen können unter semantischen Aspekten einer von zwei Gruppen zugeordnet werden:

- Prozessverlauf
- Verlauf der Nutzung von Ressourcen

Unter syntaktischen Aspekten können dynamische Statusanzeigen einer der folgenden Gruppen zugeordnet werden:

- Numerische Werte mit den gleichen Unterteilungen wie bei statischen Statusanzeigen (siehe Seite 122)
- Geordnete Liste diskreter Schritte
  - mit Gesamtüberblick
  - nur der aktuelle Schritt, ohne Gesamtüberblick
- Beliebige Texte

Während numerische Werte sowohl für die Anzeige eines Prozessverlaufs als auch für die Anzeige des Verlaufs der Nutzung von Ressourcen verwendet werden, wird die geordnete Liste diskreter Schritte lediglich für die Anzeige eines Prozessverlaufs verwendet.

Beliebige Texte werden für ein Log benutzt, d.h. für ein Protokoll der Systemreaktionen auf Anforderungen des Benutzers.

### **Auswertung unter formalen Aspekten**

#### **Art der Darstellung**

Statusinformationen können zunächst einmal entweder optisch oder akustisch präsentiert werden. Akustische Ausgaben können entweder ein Dauerton, ein kurzes akustisches Phänomen wie ein einzelner Ton oder eine kurze Melodie oder ein gesprochenener Text sein. Optische Ausgaben können numerisch, textuell oder grafisch präsentiert werden.

Numerische Informationen können mit erklärendem Text, ohne Erklärung oder als genauer Wert zur Präzisierung einer Grafik verwendet werden.

Texte können entweder ausführlich oder Abkürzungen sein.

Die größte Vielfalt gibt es bei der grafischen Präsentation von Statusinformationen. Torten- und Balkendiagramme beispielsweise zeigen einen relativen Anteil. Kurven können den zeitlichen Verlauf von Werten zeigen. Animationen visualisieren Vorgänge. Verschiedene Icons können Zustände visualisieren. Nicht zuletzt die Statusvermittlung durch Bedienelemente ist sehr häufig grafisch realisiert, nur wenige Bedienelemente präsentieren den Status auf nicht-grafischem Weg.

### **Form der Darstellung**

Optische Statusanzeigen treten in drei verschiedenen Formen auf:

- Als reine Anzeige,
- als Anzeige ohne Hinweis darauf, dass sie gleichzeitig zum Aufruf von Bedienelementen genutzt werden kann, und
- als Bedienelement, das zusätzlich den Status der zugehörigen Funktion anzeigt.

Während bei den ersten beiden Formen der Schwerpunkt bei ihrer Gestaltung auf der Vermittlung von Statusinformation liegt, so liegt er bei den Bedienelementen auf der Bedienbarkeit. Die Vermittlung von Statusinformationen ist mehr oder weniger ein Abfallprodukt.

Akustische Statusanzeigen sind stets reine Anzeigen.

### **Gestaltung der Darstellung**

Je nach gewählter Art der Darstellung gibt es verschiedene Mittel zur Gestaltung. Ein universelles Mittel, das unabhängig von der gewählten optischen Art der Darstellung verwendet werden kann, ist Farbe. Farbe wird unter anderem verwendet, um verschiedene Zustände eines Objekts zu visualisieren. Insbesondere das Fehlen von Farbe als Kennzeichnung für inaktive Objekte wird häufig verwendet (diese Objekte wirken dann Grau). Sättigung oder Helligkeit einer Farbe können verändert werden, um grobe quantitative Aussagen zu illustrieren („Einige, aber nicht alle.“). Mehrere verschiedene Farben werden zur Unterscheidung von Objekten verwendet.

Bei numerischer oder textueller Ausgabe von Statusinformationen werden Schriftattribute wie beispielsweise Fettschrift verwendet, um bestimmte Teile hervorzuheben.

Bei einer sich ändernden grafischen Darstellung kann zwischen unmerklichen und deutlich wahrnehmbaren Änderungen unterschieden werden. Balkendiagramme bei-

spielsweise haben kontinuierliche<sup>1</sup> oder segmentierte Balken. Die Änderung geschieht also nahezu unmerklich stufenlos, oder in größeren, deutlich wahrnehmbaren Schritten.

Balkendiagramme gibt es sowohl in abstrakter Form als auch bildlich, dann oft als simulierte Anzeigen technischer Geräte.

Akustische Ausgaben können in Parametern wie beispielsweise Lautstärke, Klangfarbe oder Tonhöhe variiert werden.

### **Ort der Darstellung**

Optische Statusanzeigen werden an vier verschiedenen Orten angezeigt:

- in einer speziell dafür vorgesehenen Statusleiste, die sich in der Regel am unteren Rand des Fensters befindet,
- in einem separaten Dialog, der entweder immer oder nur bei längeren Aufgaben offen ist,
- im Hauptfenster der Anwendung, oder
- als geänderter Mauszeiger.

In Statusleisten keine eindeutig als solche zu erkennenden Bedienelemente vorhanden. Anzeigen, die ohne optischen Hinweis zum Aufruf von Bedienelementen genutzt werden können, sind hier jedoch ausschließlich zu finden. Statusleisten enthalten alle Arten der Darstellung.

In separaten Dialogen sowie im Hauptfenster der Anwendung sind reine Anzeigen und Bedienelemente vorhanden. Auch hier gibt es alle Arten der Darstellung.

Mauszeiger sind stets eine reine Anzeige und werden nur grafisch dargestellt.

Akustische Statusanzeigen sind im Raum präsent und nicht an einen bestimmten Ort gebunden.

### **Lebensdauer der angezeigten Information**

Hier kann zunächst einmal unterschieden werden zwischen flüchtigen und nicht-flüchtigen Statusanzeigen. Grundsätzlich sind alle akustischen Statusanzeigen flüchtig. Flüchtige optische Statusanzeigen gibt es überwiegend bei Anzeigen von dynamischen Informationen. Sie können in allen drei Arten auftreten. Abbildung 66

---

<sup>1</sup> Aufgrund der diskreten Bildschirmauflösung sind Balkendiagramme in Wahrheit natürlich nie richtig kontinuierlich. Aufgrund der geringen Größe eines Bildschirmpunkts jedoch wird trotzdem der Eindruck eines kontinuierlich, also nicht in Stufen, wachsenden Balkens erzeugt.



(Seite 111) beispielsweise zeigt eine flüchtige numerische Anzeige, die Balkendiagramme in Abbildung 32 (Seite 101) sind eine flüchtige grafische Anzeige.

Aber auch Informationen in nichtflüchtigen Statusanzeigen haben unterschiedlich lange Lebensdauern. Die Kurven in Abbildung 32 informieren über einen festen begrenzten Zeitraum. Ist dieser Zeitraum vorbei, sind ältere Informationen nicht mehr verfügbar. Andere Informationen hängen von Ereignissen ab, beispielsweise ändert sich die Form des Mauszeigers, wenn der Benutzer den Modus des Programms ändert, oder wenn der Mauszeiger über einen anderen Bildschirmbereich geschoben wird. Auch nicht vom Benutzer beeinflussbare Ereignisse wie eine eintreffende E-Mail können einen Zustand und die zugehörige Statusanzeige ändern.

### Präsenz

Für optische Statusanzeigen kann zunächst einmal unterschieden werden, ob die Statusanzeige ständig präsent ist, nach Aufruf durch den Benutzer wie bei den in Abbildung 24 (Seite 98) gezeigten Laufwerkseigenschaften oder vom System initiiert wird, beispielsweise beim Eintreffen einer E-Mail.

### Kombination von Inhalt und Darstellung

Ein Blick auf Tabelle 6 zeigt, dass der Inhalt und die Form relativ unabhängig voneinander sind. Lediglich geordnete Listen gibt es nur als Anzeige.

Form	Inhalt	Ungeordnete Menge von Zuständen	Numerisch	Beliebiger Text	Räumlich	Geordnete Liste
Reine Anzeige		X	X	X	X	X
Anzeige mit Bedienmöglichkeit		X	X	X	X	
Bedienelement		X	X	X	X	

Tabelle 6: Kombinationen von Inhalt und Form der Darstellung

Anders sieht es bei der in Tabelle 7 gezeigten Kombination von Inhalt und Art aus:

Art	Inhalt	Ungeordnete Menge von Zuständen	Numerisch	Beliebiger Text	Räumlich	Geordnete Liste
Langer Ton		X				
Kurzer Ton		X	X			
Gesprochener Text		X		X		
Numerisch			X			
Text		X		X	X	X
Grafisch		X	X		X	

Tabelle 7: Kombinationen von Inhalt und Art der Darstellung

Töne werden außer bei Zuständen nur noch für numerische Werte eingesetzt. Eine numerische Darstellung wird nur für numerische Inhalte verwendet. Geordnete Listen und beliebige Texte werden ausschließlich textuell dargestellt.

Tabelle 8 zeigt eine Zusammenfassung der Kombinationen von dargestelltem Inhalt und Ort der Darstellung. Zustände können an allen Orten präsentiert werden, sie sind der einzige Inhalt, der vom Mauszeiger dargestellt werden kann. Räumliche Inhalte und geordnete Listen sind nicht in der Statusleiste zu finden, was vermutlich nicht zuletzt am begrenzt verfügbaren Platz liegt.

Ort	Inhalt	Ungeordnete Menge von Zuständen	Numerisch	Beliebiger Text	Räumlich	Geordnete Liste
Statusleiste		X	X	X		
Dialog		X	X	X	X	X
Hauptfenster		X	X	X	X	X
Mauszeiger		X				

Tabelle 8: Kombinationen von Inhalt und Ort der Darstellung

Als Fazit stelle ich fest, dass die Wahl von der Form und dem Ort der Darstellung offensichtlich nicht oder nur in geringem Maß von dem darzustellenden Inhalt abhängt, da in den Beispielen nahezu alle möglichen Kombinationen gefunden wurden. Hier wird vom Systementwickler offensichtlich eine Entscheidung nach funktionalen Kriterien getroffen. Anders sieht es bei der Kombination von Inhalt und Art der Darstellung aus, wo nur ein Teil der möglichen Kombinationen in den Beispielen gefunden wurde. Bestimmte Inhalte werden also stets auf eine oder sehr wenige Arten dargestellt. Offensichtlich sind einige Arten der Darstellung besser für die Anzeige bestimmter Inhalte geeignet als andere.

### Visualisierungen von Prozessfortschritt

Prozessfortschritt wird überwiegend in Form eines Balkens visualisiert. Die überwältigende Mehrheit der Balken wächst von links nach rechts. Eine eindeutige Präferenz für segmentierte oder nicht segmentierte Balken ist nicht erkennbar. Tendenziell kann festgestellt werden, dass für Software zum Arbeiten eher abstrakte Balken verwendet werden, während Fortschrittsanzeigen bei Software zur Unterhaltung eher bildlich und mit Bezug zum Thema gestaltet wird. Falls ein Balken nicht von links nach rechts wächst, so wächst er von unten nach oben – diese Richtung liegt aufgrund der Analogie zum Wachsen von Pflanzen oder Lebewesen wahrscheinlich nahe.

Beim Scannertreiber (Abbildung 49, Seite 107) wird zwar auf der einen Seite mit einem herkömmlichen abstrakten Balken der Prozessfortschritt visualisiert. Interessant ist aber, dass der Prozess selbst seinen eigenen Fortschritt anzeigt, indem das eingescannte Bild sukzessive angezeigt wird. Die Vorschau kann hier als ein von oben nach unten wachsender Fortschrittsbalken betrachtet werden und gibt dem Benutzer nicht nur eine abstrakte Vorstellung über den Prozessfortschritt, sondern zeigt ihn aufs Pixel genau.

Eine Ausnahme von den dominierenden Balken bilden die kreisförmigen Anzeigen von Fetch und von der Noris Bank Homebanking-Software. Bei letzterer wird außerdem noch mit einer abweichenden Farbe für das letzte Viertel ein zusätzlicher Hinweis gegeben, während Fortschrittsbalken stets einfarbig sind.

Eine weitere Möglichkeit, Prozessfortschritt darzustellen, ist das Abarbeiten einer geordneten Liste. Dieses geschieht grundsätzlich von oben nach unten, so wie alle anderen Listen auch gelesen werden.

### **Bedienelemente**

Status wird mit verschiedenen Bedienelementen visualisiert. Eine einzelne Option wird mit Hilfe einer Checkbox dargestellt. Eine Auswahl von genau einer aus mehreren Alternativen wird mit einer Gruppe von Optionbuttons, mit Hilfe einer Liste, aus der genau ein Element gewählt werden kann, oder mit einem Schieberegler dargestellt. Eine Auswahl von beliebig vielen aus mehreren Möglichkeiten wird mit einer Gruppe von Checkboxes oder mit einer Liste, die die Wahl mehrerer Einträge ermöglicht, ausgewählt. Der Schieberegler bei WinAmp (Abbildung 74, Seite 115) zeigt die Lautstärke nicht nur durch die Position, sondern gibt zusätzlich auch durch die Farbe einen Hinweis.

Eine Position wird meist mit Scrollbars dargestellt, auch wenn Sonique (Abbildung 73, Seite 114) hier ein anderes Element vorstellt.

## 6 Kritische Betrachtung existierender Statusanzeigen

Die Qualität einer Statusanzeige kann anhand verschiedener Kriterien beurteilt werden:

- *Umfang der Anzeige:* Welche Informationen können bei konkreten Arbeitsschritten angezeigt werden? Welche Informationen werden dem Benutzer tatsächlich vermittelt? Wo können zusätzliche Informationen hilfreich sein? Wird unter Umständen sogar zu viel Information vermittelt?
- *Wahrnehmbarkeit:* Wo werden die Informationen angezeigt? Mit welchen darstellerischen Mitteln werden die Informationen angezeigt? Wird auf physiologische und psychologische Eigenschaften des Menschen Rücksicht genommen?
- *Verständlichkeit:* Ist die Bedeutung der Anzeige klar? Sind die Informationen konsistent zum mentalen Modell des Benutzers?
- *Nützlichkeit und Lästigkeit:* Unterbrechen Statusinformationen den Arbeitsfluss? Sind sie relevant für die Arbeit?

Diese Liste ist logisch geordnet: Zunächst einmal muss überhaupt eine Anzeige vorhanden sein. Diese muss wahrgenommen werden können, damit sie verstanden werden kann. Wurde sie verstanden, kann beurteilt werden, ob sie nützlich ist. Die im weiteren Verlauf dieses Kapitels vorgestellten Fehler habe ich in vier Unterkapitel gegliedert, analog zu den Punkten der Liste. Zur Illustration habe ich Programme gewählt, die typische Fehler zeigen und zum Zeitpunkt dieser Arbeit nach Möglichkeit entweder weit verbreitet oder leicht zu beschaffen waren, damit bei Interesse der Fehler im Kontext angesehen werden kann. Ich werde nach Möglichkeit nicht nur den Fehler zeigen, sondern auch gleich vorschlagen, wie er zu vermeiden ist.

Mit einem Abschnitt über das Testen von Statusanzeigen in der wissenschaftlichen Literatur und einem über Statusanzeigen bei Hardware endet dieses Kapitel.

### 6.1 Informationsumfang

Bezüglich des Informationsumfangs sind vier Fehler zu finden:

- Ein Modus wird nicht angezeigt,
- Informationen über einen Prozessfortschritt werden nicht angezeigt und
- es wird nur über einen Teilprozess informiert, nicht aber über den Gesamtprozess.
- Sehr selten zu finden ist hingegen ein Übermaß an Information.

## Nicht angezeigter Modus

Wordpad, eine einfache Textverarbeitung unter Windows, zeigt den momentanen Modus bezüglich Einfügen bzw. Überschreiben weder in Form eines geänderten Textcursors, noch in irgendeiner anderen Form wie beispielsweise einem Eintrag in der Statusleiste an. Versehentlicher Datenverlust durch Überschreiben ist insbesondere bei 10-Finger-Blind-Schreibern sehr leicht möglich.

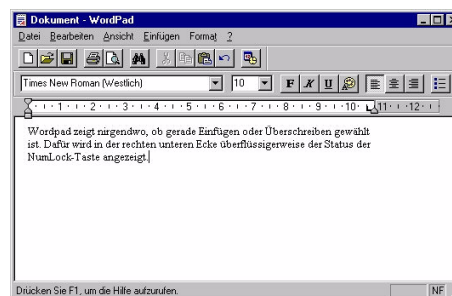


Abbildung 86: Keine Anzeige für Einfügen / Überschreiben bei Wordpad

## Fehlende Information zum Prozessfortschritt

Im StarOffice 5.2 gibt es verschiedene längere Aktionen, während derer der Benutzer nicht über den Fortgang informiert wird:

- Im Modul zur Textverarbeitung gibt es eine Funktion zum gleichzeitigen Aktualisieren aller Verknüpfungen, aller Verzeichnisse und der Seitenformatierung. Bei längeren Dokumenten mit vielen verknüpften Grafiken, so auch bei dieser Diplomarbeit, kann der Vorgang durchaus eine Minute dauern. Während dieser Zeit reagiert StarOffice auf keine Aktion des Benutzers.
- Beim Download von Mails mit StarOffice 5.2 gibt es keine Fortschrittsinformation. Insbesondere mit einer langsamen Internetverbindung kann das Abrufen einer längeren Mail mehrere Minuten dauern; währenddessen erhält der Benutzer keine Information, ob überhaupt etwas passiert. Erstaunlicherweise war das bei der Vorgängerversion 5.1 noch besser gelöst, dort wurde die Anzahl der bereits empfangenen Bytes in der Statusleiste hochgezählt.

## Fehlender Gesamtüberblick

Der Copy-Dialog in Windows 95 zeigt zwar durch die Animation mit dem fliegenden Papier, dass etwas passiert. Außerdem wird ein Fortschrittsbalken für den Fortschritt beim Kopieren einer einzelnen Datei gezeigt. Beim Kopieren mehrerer Dateien fehlt jedoch eine Information über den Fortschritt des kompletten Prozesses ([Cooper, 1995], Seite 317). Selbst bei neuesten Windows-Versionen hat sich hieran nichts geändert.



Abbildung 87: Kein Überblick über den Gesamtprozess

## Übermaß an Information

Beim Kopieren mehrerer Dateien unter Windows wird der Dialog mit dem Fortschrittsbalken auch für kleinere Dateien angezeigt. Dadurch entsteht eine unruhige Anzeige.

## 6.2 Wahrnehmbarkeit

Das größte Problem bei der Wahrnehmbarkeit sind unvollständig angezeigte Informationen, bei denen nach Murphy's Law mit Sicherheit gerade der wichtigste Teil nicht gesehen werden kann. Die menschliche Physiologie muss auf jeden Fall berücksichtigt werden, insbesondere bei der Verwendung von Farbe. Kontrastarme oder gar kontrastlose Anzeigen erschweren die Wahrnehmbarkeit ebenso wie die Wahl der falschen Darstellung, beispielsweise Text statt einer Grafik. Sind die Anzeigen ungünstig platziert, leidet die Wahrnehmung ebenfalls.

### Unvollständige Anzeige

Ist der in der Statusleiste des Internet Explorers angezeigte



Abbildung 88: Unvollständige Dateinamen beim Download

Pfad zu lang, um vollständig angezeigt zu werden, wird rechts abgeschnitten. In der Regel jedoch stammen alle zu einer Seite gehörenden Objekte von der selben Stelle, beispielsweise dem selben lokalen Verzeichnis oder dem selben Server im Internet. Folglich werden hier die eigentlich relevanten Informationen abgeschnitten und nur der meist für alle Objekte gleiche Teil wird angezeigt. Besser wäre es, einen Teil aus der Mitte durch Auslassungszeichen zu ersetzen, damit die relevanten Informationen (Internet-Server bzw. lokales Laufwerk und erstes Verzeichnis sowie der Dateiname) ständig sichtbar sind. Abbildung 88 legt aber auch eine andere Verbesserungsmöglichkeit nahe, nämlich eine Neuverteilung des Platzes in der Statusleiste. Insbesondere der Bereich rechts zur Anzeige für die Zone, aus der die momentan angezeigte Datei stammt, ist viel größer als notwendig. Eine Verkleinerung zugunsten des linken Bereichs bringt eine weitere Verbesserung.

WS\_FTP95LE führt ein Log, in dem unter anderem festgehalten wird, ob eine Datei erfolgreich übertragen wurde. Jedoch stehen für die Anzeige des Logs nur drei Zeilen zur Verfügung (ohne Möglichkeit, mehr Zeilen zur Verfügung zu stellen). Beim Übertragen mehrerer Dateien kann nur durch Scrollen herausgefunden werden, ob die Übertragung einzelner Dateien fehlgeschlagen ist.

## Ignorieren der menschlichen Physiologie

Ein in Abbildung 89 zu erkennendes Problem ist ein Farbwechsel von Rot nach Grün. Da viele Männer rot-grünblind sind (siehe Seite 17), kann ein großer Teil der Nutzergruppe diese Anzeige nicht oder nur sehr schwer wahrnehmen.



Abbildung 89: Offline oder Online nur durch unterschiedliche Farben zu erkennen (Farbe s. Anh.)

Ein weiteres Problem, das in Abbildung 89 zu sehen ist, ist eine in Bezug auf die menschliche Physiologie ungünstige Position auf dem Bildschirm. Der Screenshot stammt aus der rechten unteren Bildschirmecke, also aus dem peripheren Sichtfeld. Im peripheren Sichtbereich können Farben nicht sonderlich gut wahrgenommen werden (vgl. Kapitel 3.2, *Optische Wahrnehmung (Sehen)*). Beiden für die Anzeige des Online-Betriebs zuständigen Icons ist gemein, dass sie den Status lediglich durch verschiedene Farben anzeigen. Der Benutzer kann also nicht „mal eben aus dem Augenwinkel“ kontrollieren, ob er online oder offline ist, sondern muss seine ungeteilte Aufmerksamkeit der Anzeige widmen. Das unterbricht ihn bei jeder anderen Tätigkeit, die er gerade ausführt.

## Farbe als einziges Mittel zur Differenzierung

Die beiden rechten Icons in Abbildung 89 zeigen dem Benutzer, ob er online oder offline ist. Beide Zustände unterscheiden sich nur durch ihre Farbe. Die geringe Größe der Icons (ein Quadrat mit etwa 5 mm Kantenlänge) erschwert das Erkennen seiner aktuellen Farbe zusätzlich (vergleiche *Assoziative Dimension*, Seite 51).

## Zu geringer Kontrast

Die einzige Entschuldigung für die gewählten Farben der in Abbildung 90 gezeigten Fortschrittsanzeige ist der Wunsch des Designers, ein naturgetreues Abbild zu schaffen. Für eine gute Wahrnehmbarkeit haben sie jedoch zu wenig Kontrast.



Abbildung 90: Kontrastarme Farben (Farbe s. Anh.)

Fehlender Kontrast erschwert auch in Abbildung 91 die Wahrnehmung des gewählten Bildtyps. Zwar haben die Designer gleich zwei Hinweise eingebaut – die gewählte Option hat einen blassgelben Rahmen, nicht gewählte Optionen einen schwarzen Schatten –, jedoch ist



Abbildung 91: Fehlender Kontrast bei der Umrandung (Farbe s. Anh.)

der Kontrast zwischen einer nicht gewählten und der gewählten Option trotzdem zu gering.

Sind Grafiken, die verschiedene Zustände visualisieren, einander zu ähnlich, fällt die Wahrnehmung der einzelnen Zustände sehr schwer, wie ein Blick in den Ausgangskorb von StarOffice zeigt: Der Status einer Nachricht ist nur durch genaues Hinsehen zu erkennen<sup>1</sup>.




Empfänger	Titel
 pth...	Nachricht wird gerade versendet
 pth...	Versandte Nachricht
 pth...	Nicht versandte Nachricht

Abbildung 92: Zu ähnliche Icons

## Fehlende Hervorhebung

StarOffice zeigt im Explorer eine Übersicht aller Postkörbe. Während bei den Eingangskörben ein Zustand, der die Aufmerksamkeit des Benutzers erfordert („Es sind ungelesene E-Mails enthalten“), durch fette Schrift gekennzeichnet wird, muss der Benutzer beim Ausgangskorb erst die numerischen Werte lesen und interpretieren, um zu erfahren, ob einige E-Mails nicht versandt werden konnten, da dieser Zustand, der ebenfalls die Aufmerksamkeit des Benutzers erfordert, nicht durch fette Schrift hervorgehoben wird.

 Ausgangskorb (1/3)
 <b>Mail (iva) [2/138]</b>
 Mail (Student) (1)

Abbildung 93: Fehlende Hervorhebung für den Ausgangskorb

## Falsche Darstellung (Text statt Grafik)

Eine gute Grafik fasst viele numerische Werte auf kleinem Raum zusammen und kann die zugrunde liegenden Daten besser visualisieren als eine numerische Darstellung ([Tuft, 1984], Seite 13). So ist eine grafische Anzeige von Geschwindigkeiten besser wahrnehmbar als eine numerische, da eine Grafik die den Benutzer interessierende relative Geschwindigkeit sofort zeigt, während die absoluten numerischen Informationen erst noch verarbeitet werden müssen (vgl. auch *Zu detaillierte Information*, Seite 139).

## Falscher Ort zum Anzeigen eines Modus

Viele Programme zum Eingeben von Texten haben zwei verschiedene Modi, einen Einfügemodus und einen Überschreibenmodus. StarOffice (sowie unter anderem auch der Marktführer, Microsoft Word) benutzt für beide Modi den selben Textcursor. Der Modus wird als Abkürzung in der Statusleiste am unteren Bildschirmrand angezeigt.

Beim Einfügen ist der Cursor ein schmaler Strich

Beim Überschreiben ist der Cursor ebenfalls ein schmaler Strich

Abbildung 94: Einfügen und Überschreiben bei StarOffice

<sup>1</sup> Die Texte neben dem Icon sind die Betreffzeile der jeweiligen Nachricht und gehören nicht zur Statusanzeige.



Dass es besser geht, zeigt 4NT, ein Kommandointerpreter. Hier wird der Einfügemodus durch



Abbildung 95: Einfügen und Überschreiben mit 4NT

einen Unterstrich als Textcursor, der Überschreibenmodus durch einen Block dargestellt. Der momentane Modus ist also direkt im Aufmerksamkeitsbereich des Benutzers zu sehen (vgl. [Brodbeck und Rupietta, 1994], Seite 206). Diese Lösung kann nicht unbedingt für eine WYSIWYG-Textverarbeitung übernommen werden, da die Breite des Cursors bei Proportionalschriften stark schwanken wird. Jedoch sollte auf jeden Fall der Modus durch den Cursor angezeigt werden.

Ein gutes Beispiel ist auch Photoshop 5.0 LE, wo der Mauszeiger für jeden Modus die Form des verwendeten Werkzeugs annimmt. Der Benutzer sieht bei seiner Arbeit sofort, mit welchem modalen Werkzeug er gerade arbeitet. Zusätzlich wird der Modus durch eine gedrückte Schaltfläche in der Werkzeugpalette angezeigt.



Abbildung 96: Veränderter Mauszeiger zeigt den Modus an

## Verteilte Darstellung zusammengehörender Informationen

Der Scrollbar informiert auf einen Blick visuell über die relative Position im Dokument sowie die Größe des Dokuments in Relation zum sichtbaren Ausschnitt. Diese Informationen sind meist am rechten Fensterrand. Viele Anwendungsprogramme informieren auch numerisch über die aktuelle Seite und die gesamte Seitenzahl, so auch StarOffice 5.2. Diese Information ist hier links unter dem Dokument und damit weit entfernt vom Scrollbar. Zusammengehörende Informationen sind also auseinander gerissen. Der Screenshot in Abbildung 97 zeigt ein für illustratorische Zwecke stark verkleinertes Fenster, keine Vollbilddarstellung. Bei einer Vollbilddarstellung ist die räumliche Distanz zwischen der numerischen Anzeige der Seitenzahl und der Visualisierung durch den Scrollbar sogar noch größer. Hier hilft ein einfaches Umordnen der Informationen, so dass die Seitenzahl ganz rechts steht.

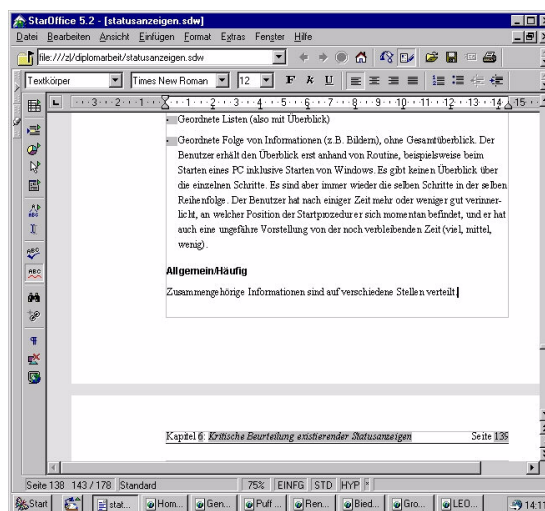


Abbildung 97: Zusammengehörende Informationen an auseinander liegenden Positionen

## Verbesserung durch Konsistenz

StarOffice bietet dem Makroprogrammierer Zugriff auf den Fortschrittsbalken in der Statusleiste. Ein Fortschrittsbalken im StarOffice sitzt also stets an der selben Stelle und fördert dadurch die Bildung von Wahrnehmungsgewohnheiten, die bei der schnellen und sicheren Orientierung helfen (vgl. [Oberquelle, 1994], Seite 113).

## Nutzungsart des Programms nicht berücksichtigt

Der Windows CD-Player, eine kostenlose Beigabe zum Betriebssystem, kann entweder normal angezeigt oder minimiert werden. Im nicht minimierten Zustand (Abbildung 98) zeigt das Programm in etwa das, was ein „normaler“ CD-Player ebenfalls zeigt. Im minimierten Zustand werden die Informationen aus dem Display in der Schaltfläche der Task-Leiste angezeigt. Der CD-Player ist nach der in Kapitel 4.4 im Abschnitt *Der Einfluss des Programms* (Seite 79) vorgestellten Einteilung ein dämonisches Programm, das meistens im Hintergrund läuft und keine Interaktion mit dem Benutzer benötigt. Wenn der Benutzer vergisst, den CD-Player zu minimieren, während er seinen eigentlichen Aufgaben nachgeht, erhält er keine Informationen mehr, da in der Regel das souveräne Programm das Fenster vom CD-Player verdeckt. Hier wäre es besser, die im minimierten Zustand sichtbaren Informationen in der Taskleiste ständig anzuzeigen.

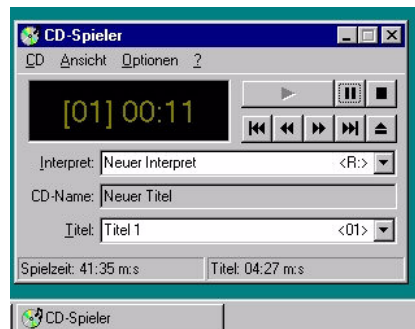


Abbildung 98: Windows CD-Player

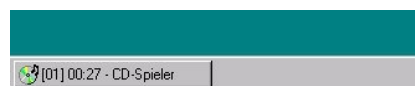


Abbildung 99: Windows CD-Player minimiert

## Verbesserung durch Redundanz

Der Lautstärkeregler von WinAmp zeigt die Lautstärke zunächst einmal durch seine Position. Zusätzlich jedoch verändert sich die Farbe der Bahn, auf der der Regler geschoben wird: Grün bei niedrigen Lautstärken, Gelb bei mittleren und Rot bei hohen. Die Übergänge zwischen den Farben sind fließend.



Abbildung 100: Lautstärkeregler zusätzlich mit veränderlicher Farbe

## 6.3 Verständlichkeit

Eine unverständliche Anzeige kann zunächst einmal durch eine ungeschickt gewählte Metapher entstehen (vgl. *Metaphern und konzeptuelle Modelle*, Seite 30). Aber auch

eine zu abstrakte oder eine zu verspielte Darstellung mindert die Verständlichkeit. Unverständliche numerische Anzeigen können oft durch Erläuterungen verständlich gemacht werden. Auch Konsistenz fördert das Verständnis. Bei textuellen Anzeigen können fremdsprachige Texte, ungenaue Formulierungen oder zu technische Details die Verständlichkeit der Anzeige beeinträchtigen.

### Wahl der Metapher

Bei der Installation von dBase 5.0 wird der bereits installierte Teil mit einem Tacho angezeigt. Ein Tacho zeigt in der Regel eine Geschwindigkeit, also eine Menge je Zeit. Der bereits installierte Teil ist jedoch keine Menge je Zeit, sondern eine reine Menge. Auch erfahrene Benutzer erfassen nicht auf Anhieb, was visualisiert wird, sondern müssen das Gesehene erst einmal verarbeiten (vgl. auch die Befragung von *Benutzer 2*, im Anhang auf Seite 168).



Abbildung 101: Falsche Metapher

Ein positives Beispiel für die Wahl der Metapher liefert Photoshop. Die in der Palette wählbaren Werkzeuge entsprechen Vorbildern aus der realen Welt. Durch eine entsprechende Darstellung ist die Funktion auf einen Blick zu erkennen.



Abbildung 102: Werkzeugkasten von Photoshop

### Zu abstrakte Darstellung

Abbildung 103 zeigt verschiedene Schaltflächen aus einer Werkzeugleiste des StarOffice-Diagrammmoduls. Von oben nach unten bedeuten sie: Diagrammtyp bearbeiten, Autoformat, Diagrammdaten bearbeiten, Daten in Reihen, Daten in Spalten, Textskalierung, Diagramm neu anordnen. Die Icons versuchen also, abstrakte Konzepte zu visualisieren. Die resultierende Darstellung ist dadurch sehr abstrakt, und der Benutzer kann die Bedeutung der Schaltflächen nicht mehr erkennen (vgl. Seiten 65 f.).



Abbildung 103: Schaltflächen für das Bearbeiten eines Diagramms im StarOffice

## Zu verspielte Darstellung

Nicht nur eine zu abstrakte, auch eine zu verspielte Darstellung kann die Verständlichkeit einer Anzeige reduzieren. Um bei der Website des Palastorchesters ([Palastorchester, 2001]) zu erkennen, dass der nach rechts rutschende Hund den Fortschritt beim Abspielen einer Hörprobe visualisiert, muss der Benutzer die Anzeige eine längere Zeit beobachten, um dann aus der Bewegung seine Schlüsse ziehen zu können.

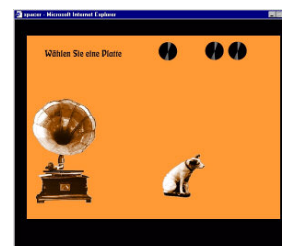


Abbildung 104: Zu verspielte Anzeige

## Fehlende Erläuterung

WS\_FTP95LE zeigt in einem separaten Dialog verschiedene Informationen zur aktuellen Dateiübertragung. Ohne Erläuterung kann der Benutzer jedoch nicht verstehen, was die einzelnen Zahlen bedeuten, auch wenn die ersten beiden recht leicht zu entschlüsseln und die letzten beiden als Zeit zu erkennen sind.

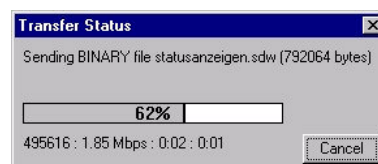


Abbildung 105: Dateitransfer mit WS\_FTP95LE

Eine Lösung für dieses Problem bietet Opera. Hier kann der Benutzer sich für unverständliche Anzeigen einen Tooltip anzeigen lassen, der die Bedeutung des angezeigten Wertes erläutert.

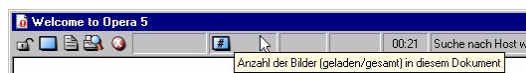


Abbildung 106: Tooltip zur besseren Verständlichkeit der angezeigten Werte

Informationen können ohne weitere Erläuterung jedoch nicht nur unverständlich, sondern auch missverständlich sein: StarOffice 5.2 zeigt bei der Installation die restliche Zeit an. Dank fehlender Erläuterung ist jedoch nicht zu erkennen, dass dieser Balken nur den Fortschritt eines einzelnen Schritts aus der Liste, nämlich *Kopieren und Entpacken der Programmdateien*, anzeigt. Ein Benutzer, der sich auf die angezeigte Restzeit verlässt, muss anschließend noch fast eine weitere Minute warten, bis der Installationsvorgang tatsächlich vorbei ist. Die beste Lösung der konkreten Situation wäre natürlich ein Fortschrittsbalken, der den kompletten Installationsvorgang visualisiert. Ansonsten sollte der Balken eine Erläuterung haben, aus der hervorgeht, dass er nur den Fortschritt eines Teilprozesses darstellt.



Abbildung 107: Fehlende Erläuterung zum Fortschrittsbalken

## Fehlende Differenzierung

Während die beiden linken vertikalen Balken in Abbildung 108 einen Fortschritt anzeigen, zeigt der rechte vertikale Balken den freien Speicherplatz auf dem Installationslaufwerk an. Erst bei genauem Hinsehen wird dieses deutlich. Eine abweichende Darstellung würde die unterschiedliche Bedeutung besser vermitteln. Auch eine andere Positionierung hilft, die beiden unterschiedlichen Anzeigen zu differenzieren.

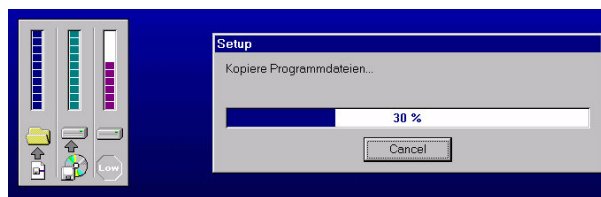


Abbildung 108: Installationsfortschritt bei WinOnCD

## Verbesserung durch bildhafte Anordnung

Eine bildhafte Anordnung von Optionbuttons, wo möglich, erspart umständliche textuelle Beschreibungen. Der Benutzer erfasst mit einem Blick, was gemeint ist. Gut gemachte Bilder oder bildhafte Anordnungen von Bedienelementen können also die Verständlichkeit erhöhen.



Abbildung 109: Bildhafte Anordnung von Optionbuttons

## Probleme bei Bedienelementen

Beim Entwurf von Bedienelementen muss zunächst einmal darauf geachtet werden, dass die Bedienbarkeit offensichtlich ist – [Norman, 1989] (Seite 21) nennt das die Gebrauchseigenschaften eines Gegenstands. Soll ein Bedienelement zusätzlich zur Vermittlung von Status verwendet werden, sollte dieses danach berücksichtigt werden. Der Standard-Skin vom Audio-Player Sonique (siehe Seite 114) zeigt besonders gut, dass dieses zwei konkurrierende Ziele sind.



Abbildung 110: Positionierung in der Playlist bzw. im aktuellen Titel (Farbe s. Anh.)

So ist zwar der große runde Knopf mit der Griffmulde leicht als Bedienelement zu identifizieren. Dass die Position der Griffmulde die Position in der Playlist visualisiert, ist jedoch nur nach intensiver längerer Beobachtung zu erkennen. Genau entgegengesetzt verhält es sich mit dem Kreis aus simulierten LEDs um die Titelinformationen, der recht schnell als Anzeige der Position zu erkennen ist. Dass er aber

auch zur Positionierung genutzt werden kann, vermittelt er nicht. Standard-Schieberegler bieten einen besseren Kompromiss als diese individuellen Bedienelemente.

## 6.4 Nützlichkeit

Statusanzeigen sind wenig nützlich, wenn sie den Benutzer durch zu viele Details verwirren oder wenn sie zu ungenau sind und ihn damit verunsichern. Auch objektiv falsche Anzeigen sind selbstverständlich nicht nützlich. Wird die Datenbasis ungünstig gewählt, können selbst richtige Anzeigen weniger nützlich sein. Redundante Anzeigen erhöhen die Nützlichkeit ebenfalls nicht. Statusanzeigen sollten den Arbeitsfluss des Benutzers nicht unterbrechen.

### Zu detaillierte Information

Der Internet Explorer zeigt die Download-Geschwindigkeit in absoluten Werten an (siehe Abbildung 113, Seite 140). Diese wird mit einer sehr großen Präzision angegeben und suggeriert damit eine höhere Genauigkeit, als tatsächlich gegeben ist. Der Benutzer hat einen erhöhten Leseaufwand, der sich nicht in einem höheren Nutzen niederschlägt.

### Zu ungenaue Information

Bei der Installation vom Windows NT 4.0 Service Pack 5 werden zwar mit je einem Fortschrittsbalken der Status des Gesamtprozesses sowie der der aktuell installierten Datei angezeigt. Das Service Pack hat einen Umfang von etwa 40 MB, die Installation nimmt also einige Zeit in Anspruch. Wie viel Zeit, muss der Benutzer allerdings schätzen, da Microsoft keine Berechnung der Restzeit vorgesehen hat. Der Benutzer muss auf seine ungenaue Schätzung zurückgreifen.

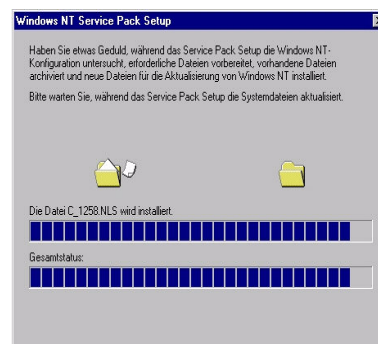


Abbildung 111: Fehlende Restzeit bei der Installation

Die Homebanking-Software der Noris Bank schließt sich aus Sicherheitsgründen automatisch, wenn der Benutzer längere Zeit keine Eingabe getätigt hat. Durch eine Grafik in der linken unteren Ecke wird angezeigt, wie lange der Benutzer schon keine Eingabe mehr getätigt hat. Ein Blick auf diese Grafik, die sich alle paar Sekunden ändert, verrät aber nicht, wie lange es noch



Abbildung 112: Wie lange noch bis zum automatischen Schließen?



bis zum automatischen Schließen dauert. Auch hier ist der Benutzer auf eine ungenaue Schätzung angewiesen, nachdem er die Anzeige einige Zeit beobachtet hat.

StarOffice 5.2 informiert nur mit einem kleinen Balken in der Statusleiste über den Fortschritt beim Laden einer HTML-Seite. Insbesondere bei langsamen Internetverbindungen und großen Seiten kann der einfarbige, nicht segmentierte Balken nicht mehr anzeigen, ob noch Daten eintreffen – die Änderungen erfolgen zu selten und sind zu klein. Um in solchen ungünstigen Situationen Auskunft über eintreffende Daten geben zu können, sollte der Fortschrittsbalken mit zusätzlichen Informationen ergänzt werden. Denkbar ist eine numerische Anzeige der bereits übertragenen Bytes, eventuell mit Anzahl der gesamten Bytes und Angabe der Übertragungsgeschwindigkeit. Alternativ wäre auch eine animierte Grafik wie beispielsweise ein Propeller denkbar: Beim Eintreffen eines Datenpaketes wird animiert. Die Drehgeschwindigkeit des Propellers visualisiert in diesem Fall die Übertragungsgeschwindigkeit. Jede Lösung hat ihre Vor- und Nachteile. Die numerischen Informationen sind genauer, benötigen aber mehr Platz und sind nicht „mal eben auf einen Blick“ zu erfassen. Eine animierte Grafik gibt „auf den ersten Blick“ einen Überblick über die Übertragung und nimmt wenig Platz ein. Dafür aber ist die Information ungenau. Daher sollte eine Kombination gewählt werden: Die animierte Grafik wird angezeigt, und in der Tipphilfe von Fortschrittsbalken und Grafik werden genauere numerische Informationen zum Ladevorgang gegeben.

### Falsche Angaben

Der Internet Explorer hat manchmal Schwierigkeiten beim Berechnen der Restzeit. Abbildung 113 zeigt eine Restzeit von 0 Sekunden, obwohl erst drei Viertel erledigt sind. Im nächsten Moment hat der Internet Explorer erneut 5 Sekunden Restzeit angezeigt. Insgesamt wurde bei diesem Download fünfmal eine Restzeit von 0 Sekunden angezeigt. Der Fortschrittsbalken hat übrigens stets korrekt informiert.



Abbildung 113: Falsche Restzeit

### Ungünstige Datenbasis

Viele Programme berechnen eine durchschnittliche Geschwindigkeit und eine Restzeit. Jedoch verfolgen nicht alle Programme die selbe Strategie. Der Internet Explorer beispielsweise nimmt die seit Beginn eines Downloads eingetroffenen Daten und die seitdem vergangene Zeit als Basis für die Berechnung der Durchschnittsgeschwindigkeit und der Restzeit. Selbst wenn also längere Zeit keine Daten mehr ein-

treffen, wird dem Benutzer eine deutlich über 0 liegende Geschwindigkeit angezeigt und so suggeriert, dass keine Probleme auftreten. Zwar könnte der Benutzer sich die Zahl der bereits eingetroffenen Bytes merken und später mit der aktuellen Anzeige vergleichen, jedoch erhöht das seine kognitive Belastung. Eine andere und in meinen Augen bessere Strategie verfolgt Napster. Hier wird als Referenz nicht der Beginn eines Downloads genommen, sondern lediglich ein nach eigenen Beobachtungen vermutlich nur wenige Sekunden zurückliegender Zeitpunkt. Dadurch erfährt der Benutzer relativ schnell, dass keine Daten mehr eintreffen, weil die durchschnittliche Geschwindigkeit auf 0 gesunken ist, ohne sich numerische Werte merken zu müssen. Ein Vorteil der größeren Datenbasis ist jedoch, dass die Berechnung der Restzeit unempfindlicher gegen momentane Schwankungen wird.

### Redundante Information

CDs können mehrere Tracks enthalten. Beispielsweise gibt es Mixed-Mode-CDs zu kaufen, deren erster Track Audiodaten enthält und die in einem zweiten Track z.B. einen Videoclip enthalten. Die meisten CDs enthalten jedoch nur einen einzigen Track. WinOnCD, ein Programm zum Brennen von CDs, zeigt während des eigentlichen Brennvorgangs stets den Fortschritt für den aktuellen sowie für die gesamten Tracks. Wird wie in Abbildung 114 nur ein



Abbildung 114: Redundante Anzeige des Gesamtprozesses

Track geschrieben, haben beide Fortschrittsbalken selbstverständlich immer die selbe Länge, einer der beiden Balken ist also überflüssig. Dass es auch besser gelöst werden kann, zeigt der Windows Commander, der beim Kopieren einer Datei einen einzelnen, beim Kopieren mehrerer Dateien zwei Fortschrittsbalken anzeigt (siehe Seite 100).

### Fehlende Bezugsgröße

Beim Download einer Datei mit dem Internet Explorer informiert ein numerischer Wert über die mittlere Geschwindigkeit (siehe Abbildung 113, Seite 140). Der numerische Wert sagt aber nur dem erfahrenen Benutzer etwas, der die maximale Geschwindigkeit seiner Internetverbindung kennt oder wenigstens schon häufiger einen Download durchgeführt hat und die momentane Geschwindigkeit in Relation zur Geschwindigkeit anderer Downloads setzen kann. Erst mit einer Bezugsgröße weiß der Benutzer, ob der Download schnell oder langsam ist. Trotz einiger Problematiken bei der Ermittlung der maximal möglichen Geschwindigkeit<sup>1</sup> sollte versucht werden,

<sup>1</sup> Berücksichtigt werden muss eine eventuelle Datenkompression, dank der die Daten schneller übertragen werden, als es die physikalische Verbindung erlaubt. Und bei einer T-DSL-Verbindung



die momentane Geschwindigkeit in Relation zu der maximal möglichen zu setzen und beispielsweise durch einen Tacho anzuzeigen.

Beim Booten von Windows NT erscheint zwischendurch ein Bildschirm, auf dem eine wachsende Reihe von Punkten einen



Abbildung 115: Starten von Windows NT

Prozessfortschritt anzeigt. Ein Ende dieser Reihe wird nicht markiert. Nach meiner Beobachtung ist die maximale Anzahl der Punkte variabel und könnte vielleicht von der Anzahl der zu ladenden Treiber abhängen, auch der erfahrene Benutzer kann also nicht mit Sicherheit sagen, wo die Reihe aufhört. Damit fehlt jede Grundlage, die noch benötigte Zeit zu schätzen, und die Nützlichkeit der Anzeige sinkt.

### Höhere Nützlichkeit durch zusätzliche Information

Das Modul zur Textverarbeitung von StarOffice 5.2 zeigt dem Benutzer beim Scrollen in einer Tipphilfe weitere Informationen über die Position im Dokument an. Das Suchen einer bestimmten Stelle wird dadurch erleichtert (vgl. auch [Cooper, 1995], Seite 407).

Seite 8 / 13 / 149 - 2.2 Definition von Status und Statusanzeigen

Abbildung 116: Zusätzliche Informationen beim Scrollen

### Unterbrechung des Arbeitsflusses

Die freie und die gesamte Größe eines Laufwerks sind unter Windows in einer Liste zu sehen, die nach Doppelklick auf das Arbeitsplatz-Symbol erscheint. Um detailliertere Informationen zu einem ausgewählten Laufwerk zu erhalten, wird ein separater Dialog geöffnet. Dieser verdeckt das ursprüngliche Fenster<sup>1</sup>, und durch den Wechsel in ein anderes Fenster wird der Arbeitsfluss unterbrochen (vgl. [Cooper, 1995], Seite 140).

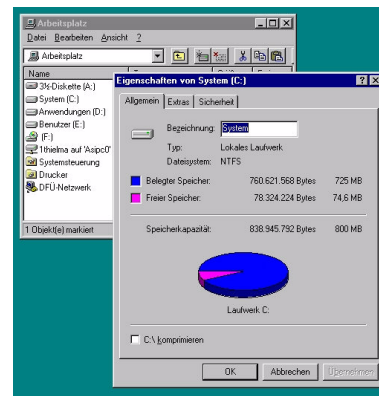


Abbildung 117: Unterbrechung des Arbeitsflusses durch einen separaten Dialog

Eine bessere Lösung zeigt Opera 5.02. Sämtliche Downloads werden zentral in einer Tabelle angezeigt. In den einzelnen Spalten der Liste werden grundlegende Informa-

wird die maximale Geschwindigkeit unter Windows NT 4.0 mit 10.000.000 Bit/s angegeben, obwohl durch Einschränkungen in der Vermittlungsstelle nur 768.000 Bit/s möglich sind.

<sup>1</sup> In Abbildung 117 habe ich den Dialog zur Illustration ein wenig zur Seite geschoben, damit das darunter liegende Fenster zu sehen ist. Ursprünglich wurde es vollkommen verdeckt.

tionen über den Download ausgegeben. Durch die begrenzte Menge an Informationen zu den Downloads bleibt die Liste verhältnismäßig übersichtlich. Der Benutzer kann einen Download auswählen und erhält zusätzliche detaillierte Informationen in einem speziellen Bereich unterhalb der Liste.

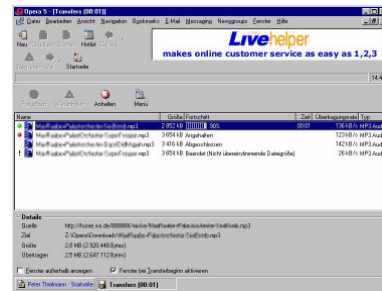


Abbildung 118: Detaillierte Informationen ohne Unterbrechung des Arbeitsflusses

## 6.5 Die schwerwiegendsten Fehler

Wie die vorigen Abschnitte dieses Kapitels zeigen, sind zu geringer Umfang und schlechte Wahrnehmbarkeit die häufigsten Probleme bei Statusanzeigen. Die folgenden Punkte sind meiner Ansicht nach die schwerwiegendsten Fehler und sollten daher bei der Gestaltung von Statusanzeigen unbedingt beachtet werden:

- Jeder Modus sollte angezeigt werden. Die beste Möglichkeit ist eine Änderung des Mauszeigers.
- Jeder Prozess ab ca. 2 s Dauer sollte anzeigen, dass das System arbeitet. Längere Prozesse sollten außerdem darüber informieren, wie lange es insgesamt noch dauert.
- Eine Farbänderung darf nicht die einzige Maßnahme zur Darstellung verschiedener Zustände sein.
- Wichtiges sollte kontrastreich dargestellt werden, um die Aufmerksamkeit des Benutzers zu erregen.
- Insbesondere bei Ausgabe von Texten, aber nicht nur dort, sollte ausreichend Platz zur Darstellung aller wichtigen Informationen vorhanden sein.

## 6.6 Testen von Statusanzeigen

[Galitz, 1988] (Seite 23) misst die Einfachheit der Benutzung unter anderem an folgenden Faktoren:

- Anzahl der Fehler in Bezug auf die Zeit oder die Anzahl der getätigten Operationen.
- Ausrufe der Benutzer, besonders die negativen („Verdammt!“).
- Die Anzahl der Benutzer, die mit dem System arbeiten wollen.

- Maximale Arbeitszeit, ohne dass der Benutzer Ermüdungserscheinungen zeigt.

[Shneiderman, 1998] gibt an, dass die Fehlerrate bei besseren Fehlermeldungen sinkt und die subjektive Zufriedenheit der Benutzer steigt. Dieses haben Studien gezeigt, bei denen Usability-Tests durchgeführt wurden (Seiten 377-378).

Unter Berücksichtigung dieser beiden Aussagen sollte die Effektivität von Statusanzeigen mit Hilfe von Usability-Tests bewertet werden. Dabei sind insbesondere zwei Punkte zu untersuchen: Die Anzahl der (objektiv feststellbaren) Fehler des Benutzers sowie seine subjektive Zufriedenheit. Die Ergebnisse eines solchen Tests sollten folgende Fragen beantworten:

- Wird der Benutzer genügend informiert, oder fehlen Informationen?
- Sind die angebotenen Informationen wahrnehmbar?
- Sind die angebotenen Informationen verständlich?

Den letzten Punkt kann bereits der Entwickler selbst testen. Für die anderen beiden Punkte sollte ein Benutzertest durchgeführt werden.

### **Test eines Bildschirms auf Wahrnehmbarkeit**

Die Qualität eines Bildschirms bezüglich Wahrnehmbarkeit kann mit dem in [Mullet und Sano, 1995] beschriebenen Squint-Test<sup>1</sup> beurteilt werden. Für diesen Test wird der Bildschirm mit einem ganz und einem halb geschlossenen Auge angesehen, eventuell aus einer leichten Distanz. Nur das, was jetzt noch sichtbar ist, kann vom Benutzer auf den ersten Blick wahrgenommen werden ([Mullet und Sano, 1995], Seite 75).

### **Beispiel eines Benutzertests**

[Musil, 1993] (Seiten 332-336) hat in einem Benutzertest verschiedene Performance- und Fortschrittsindikatoren getestet. Ausgewertet wurden subjektive Eindrücke des Benutzers: Die Benutzer sollten die Länge der Aktion schätzen und angeben, wie ungeduldig sie während des Wartens wurden. Außerdem sollten sie ihre Eindrücke und Vorlieben schildern und angeben, ob sie mit einer Methode arbeiten wollen, und für welche Art Anwendung.

Insgesamt wurden sechs Alternativen gewählt: Drei Performance-Indikatoren (eine Taschenuhr, eine Denkblase mit dem Wort *Think* und ein kehrender Besen), zwei sichtbare Fortschrittsindikatoren (ein Thermometer sowie eine Blume, die nach und

---

<sup>1</sup> Auf deutsch etwa *Test mit zusammengekniffenen Augen*

nach Blütenblätter erhält) und ein akustischer Fortschrittsindikator (eine Tonskala). Thermometer und Taschenuhr wurden zum Zeitpunkt des Tests bereits in kommerzieller Software verwendet, Besen und Blume wurden speziell für Anfänger gewählt, Denkblase und Tonskala wurden als neue und unkonventionelle Alternativen aufgenommen.

Die getesteten Benutzer waren mit egal welcher Anzeige weniger ungeduldig während des Wartens als ganz ohne Rückmeldung, fühlten sich besser informiert und hatten weniger Angst, einen Fehler gemacht zu haben.

Von den Performance-Indikatoren wurden Uhr und Denkblase für kurze Aufgaben akzeptiert, der Besen sowohl für lange als auch für kurze. Auch wenn die getesteten Benutzer Performance-Indikatoren mochten, wollten trotzdem nur wenige Benutzer mit ihnen arbeiten. Mit dem Besen wollten die wenigsten Leute arbeiten, mit der Denkblase die meisten.

Von den Fortschrittsindikatoren wurden Thermometer und Blume sowohl für lange als auch für kurze Aufgaben positiv bewertet, während die Tonskala recht deutlich für lange Aufgaben abgelehnt und nur für kurze Aufgaben akzeptiert wird.

Der Test zeigt, dass es Sinn macht, über das Design von Performance- und Fortschrittsindikatoren nachzudenken. Es ist auch vernünftig, beide Klassen von Indikatoren in einer Applikation zu verwenden – eine für kürzere, die andere für längere Aktionen.

Der Test zeigt nach Meinung von Musil außerdem, dass es mehr als nur Uhren und Balken gibt, auch wenn nicht alles gleich gut für jede Anwendung geeignet ist. Aber der Entwickler sollte sich Gedanken über Performance- und Fortschrittsindikatoren machen und nicht einfach nur die gebräuchlichen Lösungen übernehmen.

## **6.7 Hardware**

Statusanzeigen in der Hardware bestehen entweder aus Betriebsgeräuschen, aus LEDs oder aus schnell durchlaufenden Texten (vergleiche Seite 96).

Auch wenn die Betriebsgeräusche nicht gestaltet wurden, so sind sie dennoch vorhanden und werden vom Benutzer verarbeitet. Dabei irritieren fehlende Geräusche, aber auch plötzlich auftretende Geräusche, die nicht durch Aktionen des Benutzers initiiert wurden. Während ein Benutzer sich jedoch an fehlendes akustisches Feedback gewöhnen kann und statt dessen auf andere Signale vertraut, sind das für mich unmotiviert, da nicht durch seine Aktionen initiierte Geräusche auch auf Dauer irritierend. Betriebsgeräusche liefern im günstigsten Fall lediglich redundante Informa-

tionen, im schlimmsten Fall irritieren und verunsichern sie den Benutzer, der zudem keine Möglichkeit hat, sie zu unterdrücken oder zu beeinflussen. Glücklicherweise werden PC-Systeme nicht zuletzt dank gesetzlicher Regelungen immer leiser, die Betriebsgeräusche sind immer weniger hörbar. Das bietet einem Hardware-Entwickler die Möglichkeit, die „fehlenden“ Geräusche nachzubilden, sie also aktiv zu gestalten.

Die textuellen Statusinformationen gleich nach dem Einschalten des Geräts sind meist nach sehr wenigen Sekunden wieder verschwunden. Hier fehlt eine Möglichkeit, die Informationen länger auf dem Bildschirm zu halten. Wie bereits auf Seite 97 erwähnt, werden diese Informationen jedoch auch nur selten benötigt, da sie für Computerlaien unverständlich sind und von Experten nur im Fehlerfall benötigt werden. Die Empfehlung würde daher heißen, diese Information im Regelfall gar nicht zu zeigen und sie dem Experten nur auf Anforderung zugänglich zu machen. Da die einzelnen Texte jedoch nacheinander erscheinen, informieren sie dadurch über den Fortschritt des Startvorgangs. Diese Information würde fehlen, wenn auf die Anzeige im Normalfall verzichtet werden würde. Eine gute Lösung dieses Dilemmas ist ein Loslösen der technischen Information von den Informationen über den Fortgang des Startvorgangs. In einfach verständlicher Form wird der Fortschritt des Startvorgangs angezeigt. Erfahrene Benutzer erhalten die technischen Details der Statusinformationen nur auf ausdrückliche Anforderung, beispielsweise als zusätzliche Information beim BIOS-Setup.

Statusinformationen der Hardware im laufenden Betrieb werden ansonsten nur über LEDs vermittelt. Diese leuchten oder blinken, einzeln oder in Gruppen, eventuell auch noch in verschiedenen Farben. Während einzelne LEDs meist noch eine Erklärung haben (z.B. durch nebenstehende Worte wie „Power“ oder ein tonnenförmiges Symbol, das für fortgeschrittene Benutzer mit einer Festplatte in Verbindung gebracht werden kann), so ist der Benutzer ohne Handbuch völlig ahnungslos, wenn LEDs die Farbe wechseln oder manchmal leuchten, zu anderen Zeiten aber blinken oder wenn Informationen durch eine Gruppe von LEDs vermittelt werden. Hier ist auf jeden Fall eine Möglichkeit für Verbesserungen gegeben. Generell sollte je anzuzeigender Information genau eine LED verwendet werden. Die Bedeutung der LED sollte außerdem in ihrer Nähe durch ein kurzes Wort oder ein prägnantes Symbol erklärt werden. Statusinformationen der Hardware werden in der Regel nur selten benötigt. Sie müssen daher einfach aufgebaut sein und ohne Hilfe durch Handbücher zu verstehen sein. Da in der Regel nur Benutzer mit Computerwissen auf diese Informationen zugreifen, können die Erläuterungen der Anzeigen auf das Wissen dieser Benutzergruppe zugeschnitten werden, es darf also etwas knapper er-

läutert werden. Die Geräte stehen meist nicht im direkten Blickfeld des Benutzers und sind manchmal vom Arbeitsplatz aus gar nicht sichtbar. Daher dürfen direkt am Gerät nur Informationen angezeigt werden, die für den normalen Betrieb nicht essenziell notwendig sind.

Probleme mit Statusinformationen gibt es aber nicht nur am PC, sondern auch bei anderen Geräten wie zum Beispiel Mobiltelefonen. Eine beliebte Funktion, besonders bei Teenagern, ist das Versenden von Textnachrichten, sogenannter SMS. Nach unendlich umständlicher und dadurch langwieriger Eingabe von Nachricht und Empfänger wird die SMS verschickt, was bis zu 15 Sekunden dauern kann. Der Benutzer erhält eine Nachricht auf dem Display, ob die SMS versandt wurde oder nicht. Diese Meldung verschwindet beispielsweise beim Siemens M35i oder dem Trium Astral nach weniger als fünf Sekunden von selbst. Auch wenn Mobiltelefone ansonsten zu allen möglichen und unmöglichen Gelegenheiten piepen, wird diese Meldung bei beiden Modellen nicht von einem akustischen Signal unterstützt. Wer also nicht die ganze Zeit konzentriert auf das winzige Display schaut, wird unter Umständen nie erfahren, ob seine SMS versandt wurde. Entweder könnten zusätzliche akustische Hinweise gegeben werden, oder die Nachricht wird erst nach Bestätigung vom Benutzer gelöscht.

## 7 Eigene Beispiele

Statusanzeigen gibt es für viele Zwecke. Viele davon sind für sehr spezielle Aufgaben entworfen. Fortschrittsanzeigen jedoch werden in sehr vielen Programmen verwendet. Daher habe ich mich entschieden zu versuchen, neue Arten von Fortschrittsanzeigen zu finden und diese – auch im Vergleich mit bisherigen Anzeigen – zu bewerten.

Stellvertretend für alle Prozesse, die eine gleichartige Aktion auf mehrere Objekte anwenden, werde ich außerdem zeigen, wie die Statusanzeige für das Kopieren von mehreren Dateien aussehen sollte.

Alle Beispiele sind animierte Dummys ohne wirkliche Funktion, die lediglich zur Illustration der Ideen dienen. Durch die Realisierung mit Macromedia Flash ist sichergestellt, dass sie verhältnismäßig unabhängig vom verwendeten Betriebssystem sind und insbesondere unter Windows und auf dem Macintosh laufen.

### 7.1 Fortschrittsanzeige

Eine Fortschrittsanzeige ist in der Regel ein von links nach rechts wachsender Balken. Andere Darstellungen sind die Ausnahme. Eine erste Überlegung ist also, welche Darstellungsformen überhaupt möglich sind.

Generell sollten Anzeigen redundant gestaltet werden. Eine zweite Überlegung ist daher, welche Formen von Redundanz möglich sind.

#### Mögliche Darstellungsformen

Außer Balken und Kreisen sind in den Beispielen keine weiteren Formen von Fortschrittsanzeigen zu finden. Kreise und Balken liegen zunächst auch nahe, weil Größe, Orientierung und Position quantitative Dimensionen sind und Unterschiede sich daher gut schätzen lassen (vgl. Seite 50). Hieraus ergibt sich auch gleich die erste Idee:

- Ein größer werdendes Objekt könnte den Fortschritt anzeigen.

Hier handelt es sich im Prinzip aber um nichts anderes als um einen in zwei Dimensionen wachsenden Fortschrittsbalken, wenn auch in beliebiger anderer Form. Die zweidimensionale Darstellung einer eindimensionalen Änderung ist jedoch problematisch (vgl. Seite 67).

Auch eine andere naheliegende Idee, ein sich bewegendes Objekt als Fortschrittsanzeige zu nehmen, kann als Variante eines Balkens aufgefasst werden. Die Website

des Palastorchester beispielsweise verwendet einen sich bewegenden Hund (siehe Seite 120). Diese Lösung ist zudem oft nicht optimal gestaltet (vgl. Seite 141).

Menschen können einfache Formen verhältnismäßig gut erkennen (vgl. die Gestaltungsgesetze, Seite 17). Daher habe ich auch die Schlussfolgerung gezogen, dass Änderungen der Form gut für Statusanzeigen geeignet sind (siehe Seite 27). Für grafische Darstellungen sollten möglichst einfache Formen verwendet werden (siehe Seite 64), so dass eine weitere Idee lautet:

- Ein Objekt mit einer sich ändernden Form, eventuell unterstützt von sich ändernder Farbe, könnte den Fortschritt anzeigen.

Ansonsten kann natürlich auch versucht werden, Balken und Kreise zu verbessern. Ein Ansatzpunkt hierfür ist:

- Einsatz von Redundanz, zum Beispiel durch Verwendung von Farbe oder akustischen Signalen.
- Verwendung von Markierungen für bestimmte Werte, beispielsweise alle 10%.

Balken können segmentiert oder kontinuierlich sein, horizontal oder vertikal dargestellt werden, zunehmen oder abnehmen.

Kreise können vollständig gefüllt, ein Ring oder nur Zeiger sein, mit oder gegen den Uhrzeigersinn drehen, zunehmen oder abnehmen. Die Zunahme kann kontinuierlich, in 10 oder in 12 Schritten erfolgen.

### **Redundante Darstellung**

Das Ansprechen mehrerer Reize vereinfacht die Informationsaufnahme (vgl. Seite 47). Erste Wahl unter diesem Aspekt sollte also sein, optische Ausgaben durch akustische Ausgaben zu ergänzen. Aber auch bei optischer Ausgabe alleine kann Redundanz eingesetzt werden. Eine Änderung der Farbe zusätzlich zur Änderung von Form oder Größe ist eine Möglichkeit. Menschen sind es aus der Natur gewohnt, dass Farbe einen Hinweis auf Zeit gibt (Seite 53). Es liegt also nahe, Farbverläufe aus der Natur zu wählen:

- *Jahreszeiten*: Frühling kann durch ein zartes Grün, Sommer durch ein kräftiges Grün, Herbst durch Braun oder Orange und Winter durch Weiß oder ein helles Grau dargestellt werden. Jahreszeiten werden aber nicht in Ländern nahe dem Äquator verstanden, wo es anders als bei uns keine Unterschiede zwischen den Jahreszeiten gibt.



- *Sonnenaufgang*: Zuerst schwarz, dann Rot, Orange, Gelb und zum Schluss ein sehr helles, fast weißes Gelb.

Aber auch andere Farbkombinationen sind möglich:

- *Ampel*: Der Wechsel von Rot über Gelb nach Grün ist eine Konvention, die im gesamten Computer verwendenden Kulturkreis verstanden wird.
- *Näher kommen*: Entfernte Objekte haben eine weniger gesättigte Farbe. Durch zunehmende Sättigung könnte das näher kommende Ende angekündigt werden.

Bei Verwendung von Farbe kann sich sowohl die Farbe der kompletten Grafik ändern als auch einzelne Teile unterschiedlich eingefärbt werden.

Auch akustische Signale können unterstützend wirken. Sich kontinuierlich ändernde Geräusche, beispielsweise Einschenken einer Flüssigkeit in ein Gefäß, können einen Hinweis auf den fortschreitenden Prozess geben. Aber auch zu bestimmten Zeitpunkten, z.B. alle 10%, alle 25% oder alle 50%, können einzelne, eventuell unterschiedliche, Geräusche ausgegeben werden. Insbesondere bei einer kreisförmigen Grafik mit der naheliegenden Assoziation zu einer Uhr können entsprechende akustische Signale gewählt werden, entweder ein Stundenschlag oder ein Viertelstundenschlag. Denkbar ist auch eine Anlehnung an Digitalwecker mit Schlummertaste, die beim ersten Mal ein einzelnes Piep, beim zweiten Weckversuch ein Doppelpiep usw. von sich geben.

## Die Beispiele

Ziel der Beispiele ist es, die in den beiden vorigen Abschnitten erwähnten Darstellungsformen sowie die Ideen zur redundanten Darstellung zu illustrieren. Als Rahmenhandlung habe ich einen Dateidownload gewählt. Alle Beispiele zeigen als Startbildschirm einen Dialog zum Speichern eines Downloads. Ein Klick auf die Speichern-Schaltfläche startet den fiktiven, fünf Sekunden dauernden Download. Die Abbrechen-Schaltfläche kann jederzeit betätigt werden und stoppt die Anzeige. In dem Dialog für den Fortschritt des Downloads ist nur das für die Demonstration wesentliche, nämlich die Anzeige des Prozessfortschritts, enthalten. Für einen Einsatz in realen Anwendungen müssen natürlich noch weitere Informationen hinzukommen, die aber für die Untersuchung der Grafik nicht relevant sind. Ich habe folgende, in fünf Gruppen eingeteilte Beispiele erstellt:

### 1. Verschiedene Formen von Kreisen und Balken

- *Segmentierter Balken*: In 20 Sprüngen zunehmender einfarbiger Balken mit 20 unterscheidbaren Segmenten.

- *Kontinuierlicher Balken*: Kontinuierlich zunehmender einfarbiger Balken.
- *Segmentierter Kreis mit Trennung zwischen den Segmenten*: In zwölf Sprüngen zunehmender einfarbiger Kreis mit zwölf unterscheidbaren Segmenten.
- *Segmentierter Kreis ohne Trennung zwischen den Segmenten*: In zwölf Sprüngen zunehmender einfarbiger Kreis ohne unterscheidbare Segmente.
- *Kontinuierlicher Kreis*: Kontinuierlich<sup>1</sup> zunehmender einfarbiger Kreis ohne Markierungen.
- *Ring*: Ring aus zwölf erkennbaren Segmenten, die der Reihe nach im Uhrzeigersinn eingefärbt werden.
- *Tacho*: Ein Zeiger bewegt sich in zehn Sprüngen von 7 Uhr nach 5 Uhr. Der Bereich zwischen 5 Uhr und 7 Uhr hat eine andere Farbe.
- *Zeiger, rechtsdrehend*: Ein Zeiger bewegt sich in zwölf Sprüngen im Uhrzeigersinn, beginnend bei 12 Uhr.
- *Zeiger, linksdrehend*: Ein Zeiger bewegt sich in zwölf Sprüngen im Uhrzeigersinn, beginnend bei 12 Uhr.

## 2. Sonstige Formen

- *Formänderung*: Ein rotes Achteck ändert sich in einem kontinuierlichen Übergang zunächst zu einem gelben Quadrat, das sich dann zu einem grünen Kreis ändert.
- *Smiley*: Bei einem gelben Smiley schieben sich in einer kontinuierlichen Bewegung die zu Beginn nach unten gebogenen Mundwinkel immer weiter nach oben.
- *Zwei größer werdende Kreise*, einer mit und einer ohne Markierung bei 25%, 50% und 75%

## 3. Balken mit Verwendung von Farbe

- *Sonnenaufgang*: Kontinuierlich zunehmender Balken mit einem kontinuierlichen Farbwechsel von Schwarz über Orange zu einem immer heller werdenden Gelb.
- *Jahreszeiten*: Kontinuierlich zunehmender Balken mit kontinuierlicher Farbänderung von Hellgrün über Grün und Braun-Orange auf Grau.

---

<sup>1</sup> Als Näherung für eine wirkliche Kontinuität habe ich 60 Sprünge implementiert. Bei der eingestellten Bildfrequenz von 12 Hz entsteht aber ein genügend kontinuierlicher Eindruck.

- *Näher kommend*: Kontinuierlich zunehmender Balken mit kontinuierlicher Zunahme der Sättigung.
- *Ampel, drei Segmente*: Kontinuierlich zunehmender Balken. Das erste Drittel ist Rot, das zweite Gelb, das dritte Grün.
- *Ampel, drei verschiedenfarbige Phasen*: Kontinuierlich zunehmender Balken. Nach einem Drittel wechselt die Farbe des gesamten Balkens von Rot auf Gelb, nach zwei Dritteln auf Grün.
- *Ampel, kontinuierlicher Farbwechsel*: Kontinuierlich zunehmender Balken mit einem kontinuierlichen Farbwechsel in der ersten Hälfte von Rot auf Gelb, in der zweiten Hälfte von Gelb auf Grün.

#### 4. Kreise mit Verwendung von Farbe

- *Zweifarbzig*: Zunahme im Uhrzeigersinn in zwölf Sprüngen. Das letzte Viertel wird heller dargestellt.
- *Vierfarbig*: Zunahme im Uhrzeigersinn in zwölf Sprüngen. Jedes Viertel wird etwas heller als das vorige dargestellt.
- *Vier verschiedenfarbige Phasen*: Zunahme im Uhrzeigersinn in zwölf Sprüngen. Nach jedem Viertel wird die komplette Grafik in einem etwas helleren Farbton dargestellt.
- *Countdown, linksdrehend*: Ein vollständig gefüllter grüner Kreis wird in zwölf Sprüngen gegen den Uhrzeigersinn geleert. Ist er zur Hälfte leer, ändert sich die Farbe auf Gelb, bei 25% auf Rot.
- *Countdown, rechtsdrehend*: Ein vollständig gefüllter grüner Kreis wird in zwölf Sprüngen im Uhrzeigersinn geleert. Bei 9 Uhr wird der Rest in einem helleren Grün angezeigt, bei 10 Uhr in Gelb und bei 11 Uhr in Rot.

#### 5. Balken und Kreise mit akustischen Signalen

- In 20 Sprüngen zunehmender *Balken mit drei Tönen am Ende*.
- In 20 Sprüngen zunehmender *Balken mit einem Ton bei jedem Viertel und einer Fanfare am Ende*.
- Im Uhrzeigersinn kontinuierlich zunehmender *Kreis mit einer Fanfare am Ende*.
- Im Uhrzeigersinn kontinuierlich zunehmender *Kreis mit einem Ton bei jedem Viertel und einer Fanfare am Ende*.

- Im Uhrzeigersinn kontinuierlich zunehmender *Kreis mit einem Ton bei 25%, zwei bei 50%, drei bei 75% und vier bei 100%* in Anlehnung an eine Uhr, die die Viertelstunden schlägt.

### **Benutzerbefragung**

Ziel der Befragung war es herauszufinden,

- mit welchen Formen der Prozessfortschritt am besten geschätzt werden kann,
- ob der Einsatz von akustischen Signalen das Schätzen erleichtert,
- ob der Einsatz von Farbe das Schätzen erleichtert.

Da im Rahmen dieser Arbeit nur eine kleine Gruppe von Benutzern befragt werden konnte, sind die Ergebnisse natürlich nicht statistisch signifikant. Zum Erkennen eines Trends ist die Befragung jedoch geeignet, nicht zuletzt dank der Heterogenität der Gruppe der befragten Benutzer.

Sämtliche Benutzer meinten übereinstimmend, dass sich Kreise besser für eine Schätzung eignen als Balken. Die Benutzer ziehen bei Kreisen den kontinuierlichen Zuwachs einem Zuwachs in Sprüngen vor. Ein sich füllender Kreis wird oft mit einer Uhr assoziiert; diese Assoziation hilft beim Herstellen einer gedanklichen Verbindung zwischen Anzeige und Laufzeit des Prozesses. Dazu passt die Forderung einer Benutzerin, die Drehrichtung solle immer der Uhr entsprechen, weil der Mensch daran gewöhnt sei. Redundante Merkmale bei Kreisen wie Farbe oder Markierungen werden von der Mehrheit der Benutzer als überflüssig abgelehnt, da sie beim Schätzen nicht helfen, sondern schlimmstenfalls sogar irritieren. Im Gegensatz zum Balken spielt die Größe beim Kreis keine Rolle, ein kleiner Kreis kann ebenso gut geschätzt werden wie ein großer.

Ein Balken ist – besonders wenn er kontinuierlich ist – schlechter zu schätzen als ein Kreis. Farbe als redundantes Merkmal bringt hier nur eine geringe Verbesserung. Wenn Farbe eingesetzt wird, dann soll sie eine sofort erkennbare Verbindung zum täglichen Leben haben. Hilfreicher als der Einsatz von Farbe ist eine grobe Segmentierung des Balkens – eine Benutzerin hat konkret nach einer Vierteilung gefragt.

Eine von der Norm abweichende Anzeige sollte andere, spezielle Vorzüge haben. So ist der Smiley witzig und mag bei einzelnen Anwendungsbereichen durchaus seine Berechtigung finden. Eine schlechtere Schätzbarkeit wird dann hingenommen. Nicht intuitiv zu verstehende Anzeigen wie die sich ändernde Form jedoch, aber auch ein größer werdender Kreis oder ein sich drehender Zeiger, die keine speziellen Vorzüge

haben, werden von den Benutzern abgelehnt. Abgelehnt werden auch Formen wie der Ring, die keine Assoziationen hervorrufen.

Zusätzliche akustische Signale wurden von den Benutzern ziemlich deutlich abgelehnt. Ein Vorteil akustischer Signale ist allerdings, dass sie die Aufmerksamkeit wieder auf den Prozess lenken. Daher findet ein Klang am Ende des Prozesses vorsichtige Zustimmung, Klänge in seiner Mitte jedoch nicht. Wenn trotzdem Klänge an markanten Positionen des Prozesses eingesetzt werden, sollten sie voneinander unterscheidbar sein. Einige Benutzer fürchteten bei akustischen Signalen eine Belästigung anderer oder durch andere, insbesondere in Großraumbüros.

Insgesamt standen die befragten Benutzern Kreisen positiver gegenüber als Balken. Ihr Wunsch war eine sich möglichst sanft ändernde Anzeige, die als angenehm zu betrachten empfunden wurde.

### **Interpretation**

Kreise erlauben das einfache Erkennen von Vierteln durch waagerechte und senkrechte Begrenzungen (vgl. Seite 65) und ermöglichen dadurch ein leichtes Schätzen des bereits erledigten Anteils.<sup>1</sup> Bei Kreisen ist es aufgrund der Abweichungen von der Horizontalen bzw. der Vertikalen möglich zu erkennen, ob es beinahe ein Viertel, genau ein Viertel oder etwas mehr als ein Viertel ist, während bei einem Balken ohne Markierungen lediglich zu erkennen ist, dass es wohl ungefähr ein Viertel ist. Nur wenn ein Balken permanente sichtbare Markierungen hat, sind Schätzungen wie beim Kreis ebenfalls möglich. Ein Balken mit zehn Segmenten oder ein Kreis mit zwölf überschreitet die „magische Nummer 7“, also die Anzahl der Objekte, die Menschen noch auf einen Blick erfassen können. Eine so genaue Einteilung sollte daher nicht vorgenommen werden. Viertel scheinen für das Schätzen der Restzeit gut geeignet zu sein.

Balken sind vorteilhaft, wenn wie in Statusleisten wenig Platz zur Verfügung steht. Sie sollten dann aber unbedingt eine permanente sichtbare Markierung der Viertel haben, um eine dem Kreis vergleichbar gute Schätzbarkeit zu gewährleisten.

Der Einsatz von Farbe bringt nur eine geringe Verbesserung der Schätzbarkeit. Die verwendete Farbkombination muss auf alle Fälle intuitiv vom Benutzer verstanden werden können, da sie ansonsten ablenken kann. Akustische Signale bringen höchstens als Ankündigung vom Ende eines sehr langen Prozesses einen Vorteil.

---

<sup>1</sup> Auch der Mund des Smiley, der bei 50% eine waagerechte Linie ist, nutzt diesen Effekt.

## 7.2 Datei kopieren

Das Kopieren von Dateien ist eine häufige Aufgabe, die insbesondere beim Kopieren mehrerer und größerer Dateien einige Minuten dauern. Daher ist es wichtig, dem Benutzer so viele Informationen über den Ablauf des Prozesses zu geben, wie dieser benötigt. Abbildung 119 zeigt zwei Dialoge, die während des Kopierens angezeigt werden. Der obere stammt von Windows NT 4.0, der untere vom Windows Commander 3.02. Windows öffnet einen Dialog für jede zu kopierende Datei und schließt ihn wieder, während der Windows Commander das Kopieren aller Dateien in dem selben Dialog anzeigt. Windows hat daher nur einen Fortschrittsbalken für die gerade zu kopierende Datei; der Windows Commander hat je einen für die aktuell zu kopierende Datei und einen für den kompletten Kopiervorgang. Der Windows Commander informiert mit der kompletten Pfadangabe für Quelle und Ziel, während Windows lediglich den Namen der Quelldatei und des Zielverzeichnisses angibt. Windows zeigt unter dem Fortschrittsbalken eine Berechnung der Restzeit. Perfekt ist keine der beiden Lösungen. Der größte Kritikpunkt an Windows ist der fehlende Überblick über den Gesamtprozess. Die gekürzten Informationen über Quelle und Ziel sind ebenfalls nicht optimal<sup>1</sup>. Positiv hervorzuheben ist die Berechnung der Restzeit bei Windows. Ihr Fehlen ist gleichzeitig der größte Kritikpunkt am Windows Commander.

Wie sieht nun eine optimale Lösung aus? [Cooper, 1995] fordert, dass alle Funktionen, die integraler Bestandteil des im Hauptfenster gezeigten Inhalts sind, nicht in einem separaten Dialog, sondern ebenfalls im Hauptfenster angezeigt werden sollen ([Cooper, 1995], Seite 320). Demnach dürfte hier kein separater Dialog verwendet werden. Das halte ich aber nicht für praktikabel, da entweder der Bildschirm zu überladen wird oder zu viel Platz im Hauptfenster reserviert bleiben muss. Ein separater Dialog, der über dem Hauptfenster liegt, nutzt den beschränkten zur Verfügung stehenden Platz besser aus.

Wichtig für den Benutzer ist auf jeden Fall die noch verbleibende Zeit für den Gesamtprozess, da er sich aufgrund dieser Angabe eventuell zwischendurch mit anderen

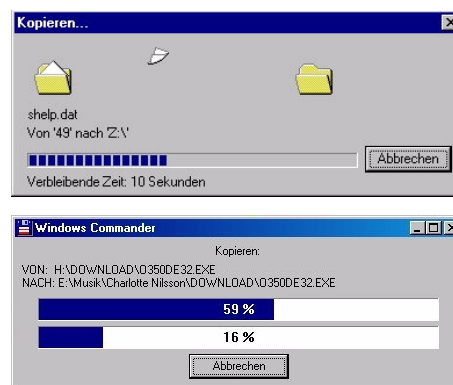


Abbildung 119: Kopieren mit Windows (oben) und dem Win Commander

<sup>1</sup> Auch wenn auf den ersten Blick vielleicht der Anfänger von der weniger technischen Darstellung profitieren mag, so reichen gekürzte Informationen nicht wenn mehrere Verzeichnisse kopiert werden, die Dateien mit gleichem Namen enthalten.

Dingen beschäftigen kann. Es sollte daher eine numerische Anzeige der verbleibenden Zeit des gesamten Vorgangs enthalten sein. Um schnell eine grobe Schätzung zu ermöglichen, sollte der Gesamtprozess ebenfalls grafisch dargestellt werden. Der Benutzer kann dann mit einem flüchtigen Blick auf die Anzeige aufgrund seines Gefühls für die bereits vergangene Zeit die restliche Dauer bestimmen.

Die Ergebnisse von Kapitel 7.1 legen nahe, einen Balken mit deutlich sichtbaren Vierteln oder einen Kreis mit kontinuierlichem Zuwachs zur Visualisierung des Prozessfortschritts zu verwenden. Eine zusätzliche numerische Anzeige des prozentualen Wertes ist überflüssig, da sie lediglich zur Berechnung der verbleibenden Zeit verwendet wird, die aber bereits angezeigt wird: Wer seine Aufmerksamkeit einem numerischen Wert zuwendet, sollte nicht einen Zwischenwert ablesen und diesen verarbeiten müssen, sondern den ihn eigentlich interessierenden Wert sofort erhalten. Und wer eine schnelle Schätzung möchte, ist mit der leichter zu erfassenden Grafik besser bedient als mit einem numerischen Wert.

Wichtig ist außerdem, dass der Benutzer stets weiß, dass das System noch arbeitet. Insbesondere bei sehr langen Kopiervorgängen oder großen Dateien wird sich die grafische Anzeige nur sehr langsam ändern, so dass der Benutzer während längerer Intervalle keine Änderung feststellen wird. Eine weitere grafische Anzeige für die aktuell zu kopierende Datei löst das Problem nur ungenügend, da bei großen Dateien auch hier ein Stillstand wahrgenommen werden kann. Daher sollten die Größe der momentan zu übertragenden Datei sowie die bereits übertragenen Bytes numerisch angezeigt werden. Ein flüchtiger Blick zeigt dem Benutzer das Durchrauschen der Ziffern und damit, dass das System arbeitet, ein genauer Blick liefert außerdem exakte Informationen.

Außerdem sollten Quelle und Ziel der gerade zu kopierenden Datei möglichst vollständig angegeben werden.

### **Die Beispiele**

Um die im vorigen Abschnitt vorgestellten Ideen zu illustrieren, habe ich drei Beispiele entworfen. Allen Beispielen gemeinsam sind

- eine numerische Anzeige der Restzeit des gesamten Prozesses,
- eine vollständige Angabe von Quelle und Ziel,
- Name und Größe der zu kopierenden Datei,
- Name des Ziels und dynamisch aktualisierter bereits kopierter Teil sowie

- eine Schaltfläche zum Abbrechen.

Der Unterschied liegt einzig in der Grafik zur Anzeige des Gesamtprozesses und in dadurch notwendigen Änderungen des Layouts:

1. Kontinuierlich zunehmender einfarbiger Balken ohne Markierungen. Die Informationen über die aktuell kopierte Datei stehen darunter.
2. Kontinuierlich zunehmender Balken mit vier verschieden hellen Segmenten und permanent sichtbaren Markierungen bei 25%, 50% und 75%. Die Helligkeit der Viertel nimmt zum Ende hin ab. Die Informationen über die aktuell kopierte Datei stehen darunter.
3. Kontinuierlich zunehmender einfarbiger Kreis. Die Informationen über die aktuell kopierte Datei stehen rechts daneben.

Die Dialoge verwenden als Hintergrund die Fläche des Windows-Dialogs. Damit ist sichergestellt, dass sie eine praktikable Größe nicht überschreiten.

### **Benutzerbefragung**

Ziel der Befragung war es, die Ergebnisse aus Kapitel 7.1 mit Hilfe eines komplexeren Layouts zu bestätigen. Das erste Beispiel dient dabei als Referenz. Es enthält alle von mir für notwendig erachteten Elemente, ist jedoch durch den einfarbigen Balken dem Windows-Original auf den ersten Blick ähnlich. Das zweite Beispiel soll versuchen herauszufinden, ob meine Schlussfolgerung aus Kapitel 7.1, dass der Benutzer einen Balken mit deutlich erkennbaren Vierteln wünscht, bestätigt werden kann. Das dritte Beispiel soll zeigen, ob der Kreis, der bei der Befragung in Kapitel 7.1 als sehr gut zum Schätzen beurteilt wurde, sich in einem komplexeren Layout bewähren kann.

Auch hier sind die Ergebnisse nicht statistisch signifikant, zum Erkennen eines Trends jedoch geeignet.

Die angebotene Information wird in allen Fällen von den Benutzern verstanden. Alle Benutzer sehen anhand der durchlaufenden Zahlen, dass der Prozess fortschreitet – für einige Benutzer ist dieses die wichtigste Information überhaupt. Informationen über die aktuell kopierte Datei sind nur für erfahrene Benutzer hilfreich, entweder bei langen Prozessen oder zur Kontrolle der eigenen Handlungen. Computerlaien brauchen die detaillierten Informationen nie. Gewünscht, wenn auch nicht wirklich benötigt, wird eine numerische Anzeige der Summe aller zu kopierenden Bytes sowie der Summe der bereits kopierten. Eine Prozentangabe wurde von keinem Benutzer gefordert, allerdings wurde auch nicht konkret danach gefragt.



Der kontinuierliche einfarbige Balken ließ sich am schlechtesten schätzen. Die verschiedenen Farben des grob segmentierten Balkens brachten zu viel Unruhe ins Bild. Jedoch wollte die Mehrheit der Benutzer die permanent sichtbaren Markierungen bei den Vierteln behalten, sie wünschten sich also einen einfarbigen Balken mit permanent sichtbaren Markierungen der Viertel. Das gezeigte Layout des Dialogs mit der kreisförmigen Anzeige wurde von den meisten Benutzern als unübersichtlich empfunden, da anders als bei den Dialogen mit den Balken das Auge nicht einfach von oben nach unten wandern kann, um alle Informationen zu erfassen, sondern sich in zwei verschiedenen Spalten orientieren muss.

Insgesamt ergab sich bei der Befragung eine Präferenz für einen Dialog mit einem Balken. Hauptgrund hierfür war die bessere Übersichtlichkeit des gesamten Dialogs.

### Interpretation

Ich habe bereits beim Entwurf gemerkt, dass der Kreis schwerer in ein komplexeres Layout zu integrieren ist als ein Balken. Die Benutzer haben dann auch das entstandene Layout kritisiert und meinten, mit einem Balken sei es übersichtlicher. Damit kehrt sich paradoxerweise die Situation um: Während der Kreis alleine besser bewertet wurde als der Balken, wurde der komplexe Dialog mit einem Balken besser beurteilt als mit einem Kreis. Daraus schließe ich, dass bei einer komplexen Statusanzeige die Übersichtlichkeit des Layouts, also die Anordnung der einzelnen Elemente, am wichtigsten ist und erst dann die verwendete Visualisierung.

Es gibt zwei Lösungen, die aufgrund der Benutzerbefragung empfehlenswert sind:

1. *Für Anfänger:* Dieses Layout enthält von oben nach unten eine kreisförmige Fortschrittsanzeige, darunter die Restzeit. Etwas nach unten abgesetzt stehen in einer Zeile die bereits kopierten sowie die insgesamt zu kopierenden Bytes, wobei die Anzahl der bereits kopierten Bytes hochgezählt wird. Eventuell können in einer weiteren Zeile noch die Zahl der aktuell kopierten Dateien sowie die gesamte Zahl stehen, wobei hier ebenfalls dynamisch aktualisiert wird. Es enthält nicht die Namen der aktuell kopierten Datei, da die Benutzer mit wenig Erfahrung bei der Befragung keinen Nutzen in dieser Information erkennen konnten.
- *Für Experten:* Diese Lösung entspricht den Beispielen mit den Balken, wobei der einfarbige Balken drei permanente Markierungen bei 25%, 50% und 75% enthält. Eventuell kann in der rechten oberen Ecke noch die Zahl der insgesamt zu kopierenden Bytes sowie der bereits kopierte Teil angezeigt werden.

Layouts mit einer kreisförmigen Fortschrittsanzeige und vollständigen Dateinamen haben das Problem, dass der Kreis im Verhältnis zu den Dateinamen eher schmal und hoch und damit schlecht zu integrieren ist. Abbildung 120 zeigt einen Vorschlag, wie ein Kreis so integriert werden kann, dass die Übersichtlichkeit des Layouts vergleichbar mit der Übersichtlichkeit eines der Beispiele mit Balken ist. Trotz eines im Vergleich mit den Beispielen kleineren Kreises ist jedoch eine leere Fläche oben rechts deutlich zu erkennen.

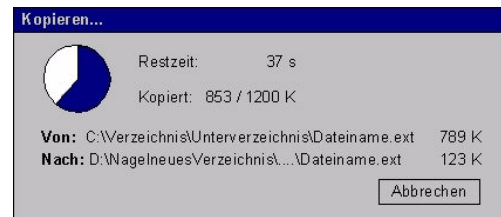


Abbildung 120: Möglicher Kopieren-Dialog mit kreisförmiger Fortschrittsanzeige

## 8 Zusammenfassung

Der Status eines Systems ist seine momentane Lage oder Situation. Hierzu gehören dynamische Zustände, insbesondere der Fortschritt von Prozessen, sowie statische Zustände, hier insbesondere ein Modus des Programms, in dem der Benutzer sich befindet. Statusanzeigen lassen sich eindeutig zu Fehlermeldungen und Online-Hilfe abgrenzen. Es gibt aber Überschneidungen mit Bedienelementen, da diese einen Zustand nicht nur setzen, sondern auch anzeigen können.

Bei der Gestaltung von Statusanzeigen müssen die menschliche Physiologie und psychologische Aspekte berücksichtigt werden. Auch Grundsätze der Informationsdarstellung sind zu berücksichtigen. Statusanzeigen helfen dem Benutzer sowohl beim Aufbau eines mentalen Modells vom System als auch Fehler zu vermeiden.

Statusanzeigen heutiger Programme sind fast ausschließlich optischer Art. In Programmen zum Arbeiten sind sie eher abstrakt, in Programmen zur Unterhaltung eher bildlich-verspielt mit Bezug zum Gegenstand des Programms gestaltet. Akustische Statusanzeigen gibt es nur bei Programmen zur Unterhaltung. Jedoch werden auch Betriebsgeräusche vom Benutzer zur Information über den Systemstatus herangezogen.

Statusanzeigen enthalten Fehler bezüglich Informationsumfang, Wahrnehmbarkeit, Verständlichkeit und Nützlichkeit. Die schwerwiegendsten Fehler sind

- ein nicht angezeigter Modus,
- ein fehlender Überblick über einen Gesamtprozess,
- die Darstellung verschiedener Zustände ausschließlich durch verschiedene Farben,
- ein zu knapper Platz für die Ausgabe längerer Texte,
- eine zu verspielte oder zu abstrakte Darstellung sowie
- die Wahl einer falschen Metapher.

Ein Benutzertest zeigte, dass für die Visualisierung eines Prozessfortschritts eine kreisförmige, kontinuierlich zunehmende Anzeige die am besten zu schätzende Darstellungsart ist. Jedoch muss bei komplexen Statusanzeigen wie dem Fortschritt beim Kopieren mehrerer Dateien zuallererst ein übersichtliches Layout geschaffen werden, da ein unübersichtliches Layout die Vorteile einer besseren Visualisierungsmethode zunichte macht.

## Danke

„So eine Arbeit wird eigentlich nie fertig, man muss sie für fertig erklären, wenn man nach Zeit und Umständen das Möglichste getan hat.“ [Johann Wolfgang von Goethe, Italienische Reise, 16.3.1787] Dieser Zeitpunkt ist für meine Diplomarbeit über Statusanzeigen nun gekommen. Ohne Unterstützung und Hilfe einiger Personen wäre er allerdings nicht erreicht worden. Danken möchte ich an erster Stelle ganz besonders Prof. Horst Oberquelle, der mich nicht nur auf das Thema aufmerksam gemacht hat, sondern auch die Arbeit umfassend betreut und für gute Arbeitsbedingungen gesorgt hat. Betreut wurde die Arbeit außerdem durch Prof. Christian Freksa, auch ihm ein herzliches Dankeschön.

Ohne meinen Kommilitonen Tim Wohlberg hätte ich dieses Thema vermutlich nicht bearbeitet. Seine Begeisterung für das Thema hat jedoch auf mich abgefärbt, so dass ich mich doch entschieden habe, Statusanzeigen näher zu untersuchen. Unterhalten haben wir uns auf einer Feier der ehemaligen Doktorandin Wiebke Oeltjen. Danken möchte ich ihr hier aber nicht für die Einladung (das habe ich bereits getan), sondern für die vorbildliche persönliche Betreuung, die sie mir und den anderen Studierenden in ihren Veranstaltungen zukommen ließ. Dadurch hat sie mir nicht nur beim Wiedereinstieg ins Studium nach einer längeren, durch private Umstände bedingten Pause sehr geholfen, sondern mich auch für die Software-Ergonomie gewonnen.

Für das Korrekturlesen sowie die konstruktive Kritik mit etlichen Denkanstößen danke ich meinem Kollegen Berend Cornelius.

Nicht zuletzt gebührt auch meinen Eltern Dank, denn ohne ihre finanzielle Hilfe hätte ich neben dem Studium noch mehr arbeiten müssen und die notwendige Zeit für diese Arbeit nicht gefunden.

## Literaturverzeichnis

- [Apple, 1992] Apple Computer, Inc.: "Macintosh Human Interface Guidelines". Addison Wesley, Reading, MA, 1992
- [Baecker, Small und Mander, 1991] Baecker, Ronald; Small, Ian; Mander, Richard: "Binging Icons To Life". (in: ACM: "Proc. CHI '91 Human Factors in Computing Systems"). ACM, New York, 1991
- [Baecker und Small, 1991] Baecker, Ronald; Small, Ian: "Animation at the Interface". (in: Laurel, Brenda: "The Art of Human-Computer Interface Design"). Addison Wesley, Reading, MA, 3. Auflage, 1991
- [Blake, 1991] Blake, Tyler: "Adventures with Hybrid Systems: Integrating the Macintosh Interface with ..". (in: Laurel, Brenda: "The Art of Human-Computer Interface Design"). Addison Wesley, Reading, MA, 3. Auflage, 1991
- [Blattner, Sumikawa und Greenberg, 1989] Blattner, Meera M.; Sumikawa, Denise A.; Greenberg, Robert M.: "Earcons and Icons: Their Structure and Common Design Principles". In: Human-Computer Interaction, Band 4, Nr. 1, 1989
- [Brockhaus, 1993] Brockhaus Enzyklopädie in vierundzwanzig Bänden. F. A. Brockhaus, Mannheim, 1993
- [Brodbeck und Rupietta, 1994] Brodbeck, Felix C.; Rupietta, W.: "Fehlermanagement und Hilfesysteme". (in: Eberleh, E.; Oberquelle, H.; Oppermann, R.: "Einführung in die Software-Ergonomie"). Walter de Gruyter, Berlin, New York, 2. Auflage, 1994
- [Buxton, 1989] Buxton, William: "Introduction to this Special Issue on Nonspeech Audio". In: Human-Computer Interaction, Band 4, Nr. 1, 1989
- [Card, Moran und Newell, 1983] Card, Stuart K.; Moran, Thomas P. und Newell, Allen: "The Psychology of Human-Computer Interaction". Lawrence Erlbaum Associates, Hillsdale, NJ, 1983
- [Cooper, 1995] Cooper, Alan: "About Face". IDG Books Worldwide, Foster City, CA, 1995
- [Crawford, 1991] Crawford, Chris: "Lessons from Computer Game Designs". (in: Laurel, Brenda: "The Art of Human-Computer Interface Design"). Addison Wesley, Reading, MA, 3. Auflage, 1991
- [Dutke, 1994] Dutke, Stephan: "Mentale Modelle: Konstrukte des Wissens und Verstehens". Verlag für angewandte Psychologie, Göttingen, 1994
- [Eberleh, 1994] Eberleh, Edmund: "Industrielle Gestaltungsrichtlinien für graphische Benutzeroberflächen". (in: Eberleh, E.; Oberquelle, H.; Oppermann, R.: "Einführung in die Software-Ergonomie"). Walter de Gruyter, Berlin, New York, 2. Auflage, 1994
- [Eberleh, Oberquelle und Oppermann 1994] Eberleh, Edmund; Oberquelle, Horst; Oppermann, Reinhard: "Einführung". (in: Eberleh, E.; Oberquelle, H.; Oppermann, R.: "Einführung in die Software-Ergonomie"). Walter de Gruyter, Berlin, New York, 2. Auflage, 1994

- [Frese, 1978] Frese, Michael: "Partialisierte Handlung und Kontrolle: Zwei Themen der ind. Psychopathol.". (in: Frese, Michael; Greif, Siegfried; Semmer, Norbert: "Industrielle Psychopathologie"). Verlag Hans Huber, Bern, 1978
- [Frumkes, 1990] Frumkes, Thomas E.: "Classical and Modern Psychophysical Studies of Dark and Light Adaption". (in: Leibovic, K. N.: "Science of Vision"). Springer Verlag, New York Berlin Heidelberg, 1990
- [Galitz, 1988] Galitz, Wilbert O.: "Handbook of Screen Format Design". QED Information Sciences, Inc., Wellesley, MA, 3. Auflage, 1988
- [Gaver, 1989] Gaver, William W.: "The Sonic Finder: An Interface That Uses Auditory Icons". In: Human-Computer Interaction, Band 4, Nr. 1, 1989
- [Gerdes, 2001] Dr. Gerdes, Claudia: "Loading...". In: PAGE. Digitale Gestaltung und Medienproduktion, Ausgabe 02.2001
- [Glaser, 1994] Glaser, Wilhelm R.: "Menschliche Informationsverarbeitung". (in: Eberleh, E.; Oberquelle, H.; Oppermann, R.: "Einführung in die Software-Ergonomie"). Walter de Gruyter, Berlin, New York, 2. Auflage, 1994
- [Gonzales, 1996] Gonzales, Cleotilde: "Does Animation in User Interfaces Improve Decision Making?". (in: ACM: "Proc. CHI '96 Human Factors in Computing Systems"). ACM, New York, 1996
- [Hacker, 1986] Hacker, Winfried: "Arbeitspsychologie. Psychische Regulation von Arbeitstätigkeiten". Verlag Hans Huber, Bern Stuttgart Toronto, 1986
- [Herzer und Noll, 1990] Herzer, Christian; Noll, Per: Feedback in Systemoberflächen. Diplomarbeit. Universität Hamburg, Fachbereich Informatik, 1990
- [Icebox, 2000] (ohne Autor): "diverse Zeichentrickfilme". [www.icebox.com](http://www.icebox.com) (Aufruf im Dez. 2000)
- [Ishii, Wisneski, Brave, Dahley und Gorbet, 1998] Ishii, Hiroshi; Wisneski, Craig; Brave, Scott; Dahley, Andrew; Gorbet, Matt: ambientROOM: Integrating Ambient Media with Architectural Space (Video). (auf: CHI 98 Video Program). ACM, New York, 1998
- [Mallot, 1998] Mallot, Hanspeter A.: "Sehen und die Verarbeitung visueller Information". Vieweg, Braunschweig, Wiesbaden, 1998
- [Mountford, 1991] Mountford, S. Joy: "Technique and Technology: Introduction". (in: Laurel, Brenda: "The Art of Human-Computer Interface Design"). Addison Wesley, Reading, MA, 3. Auflage, 1991
- [Mountford und Gaver, 1991] Mountford, S. Joy; Gaver, William W.: "Talking and Listening to Computers". (in: Laurel, Brenda: "The Art of Human-Computer Interface Design"). Addison Wesley, Reading, MA, 3. Auflage, 1991
- [MSDN, 2000/1] (ohne Autor): "Key Areas for Improvement". [www.msdn.microsoft.com/library/books/winguide/ch01e.htm](http://www.msdn.microsoft.com/library/books/winguide/ch01e.htm) (Aufruf am 18.12.2000)
- [MSDN, 2000/2] (ohne Autor): "User-Centered Design Principles". [www.msdn.microsoft.com/library/books/winguide/ch02b.htm](http://www.msdn.microsoft.com/library/books/winguide/ch02b.htm) (Aufruf am 18.12.2000)

- [MSDN, 2000/3] (ohne Autor): "Mouse Input".  
[www.msdn.microsoft.com/library/books/winguide/ch05b.htm](http://www.msdn.microsoft.com/library/books/winguide/ch05b.htm) (Aufruf am 18.12.2000)
- [MSDN, 2000/4] (ohne Autor): "Transfer Operations".  
[www.msdn.microsoft.com/library/books/winguide/ch06f.htm](http://www.msdn.microsoft.com/library/books/winguide/ch06f.htm) (Aufruf am 18.12.2000)
- [MSDN, 2000/5] (ohne Autor): "Basic Window Operations".  
[www.msdn.microsoft.com/library/books/winguide/ch07d.htm](http://www.msdn.microsoft.com/library/books/winguide/ch07d.htm) (Aufruf am 18.12.2000)
- [MSDN, 2000/6] (ohne Autor): "Menus".  
[www.msdn.microsoft.com/library/books/winguide/ch08b.htm](http://www.msdn.microsoft.com/library/books/winguide/ch08b.htm) (Aufruf am 18.12.2000)
- [MSDN, 2000/7] (ohne Autor): "Controls".  
[www.msdn.microsoft.com/library/books/winguide/ch08c.htm](http://www.msdn.microsoft.com/library/books/winguide/ch08c.htm) (Aufruf am 18.12.2000)
- [MSDN, 2000/8] (ohne Autor): "Toolbars and Status Bars".  
[www.msdn.microsoft.com/library/books/winguide/ch08d.htm](http://www.msdn.microsoft.com/library/books/winguide/ch08d.htm) (Aufruf am 18.12.2000)
- [MSDN, 2000/9] (ohne Autor): "Design of Visual Elements".  
[www.msdn.microsoft.com/library/books/winguide/ch14c.htm](http://www.msdn.microsoft.com/library/books/winguide/ch14c.htm) (Aufruf am 18.12.2000)
- [Muller, 1988] Muller, Michael J.: "Multifunctional Cursors for Direct Manipulation User Interfaces". (in: ACM: "Proc. CHI '88 Human Factors in Computing Systems"). ACM, New York, 1988
- [Mullet und Sano, 1995] Mullet, Kevin; Sano, Darrell: "Designing Visual Interfaces: Communication Oriented Techniques". SunSoft Press / Prentice Hall, Englewood Cliffs, NJ, USA, 1995
- [Musil, 1993] Musil, Sabine: "Beyond Bars and Hour Glasses: Designing Performance and Progress Indicators". (in: Grechenig, Thomas; Tscheligi, Manfred: "Human Computer Interaction. VCHCI '93"). Springer Verlag, Berlin Heidelberg, 1993
- [Myers, 1985] Myers, Brad A.: "The Importance of Percent-Done Progress Indicators for Computer-Human Int.". (in: ACM: "Proc. CHI '85 Human Factors in Computing Systems"). ACM, New York, 1985
- [Mynatt, Back, Want, Baer und Ellis, 1998] Mynatt, Elizabeth; Back, Maribeth; Want, Roy; Baer, Michael; Ellis, Jason: "Designing Audio Aura". (in: ACM: "Proc. CHI '98 Human Factors in Computing Systems"). ACM, New York, 1998
- [NeXTSTEP, 1992] (ohne Autor): "NeXTSTEP User Interface Guidelines". Addison Wesley, Reading, MA, 1. Auflage, 1992
- [Nielsen, 1989] Nielsen, Jakob: "Executive Summary: Coordinating User Interfaces for Consistency". (in: Nielsen, Jakob: "Coordinating User Interfaces for Consistency"). Academic Press, Inc., San Diego, CA, 1989

- [Nievergelt, 1983] Nievergelt, Jürg; Ventura, Andrea: "Die Gestaltung interaktiver Programme". B. G. Teubner, Stuttgart, 1983
- [Norman, 1989] Norman, Donald: "Dinge des Alltags". Campus Verlag, Frankfurt/New York, 1989
- [Oberquelle, 1994] Oberquelle, Horst: "Formen der Mensch-Computer-Interaktion". (in: Eberleh, E.; Oberquelle, H.; Oppermann, R.: "Einführung in die Software-Ergonomie"). Walter de Gruyter, Berlin, New York, 2. Auflage, 1994
- [Oberquelle, 1997] Oberquelle, Horst: "Vorlesungsskript zu Software-Ergonomie 1, WS 97/98". Universität Hamburg, 1997
- [Oppermann und Reiterer, 1994] Oppermann, Reinhard; Reiterer, Harald: "Software-ergonomische Evaluation". (in: Eberleh, E.; Oberquelle, H.; Oppermann, R.: "Einführung in die Software-Ergonomie"). Walter de Gruyter, Berlin, New York, 2. Auflage, 1994
- [OSF/Motif, 1991] (ohne Autor): "OSF/Motif Style Guide". Prentice-Hall, Englewood Cliffs, NJ, 1991
- [Palastorchester, 2001] (ohne Autor): "Hörproben".  
[www.palastorchester.de/platten.html](http://www.palastorchester.de/platten.html) (Aufruf am 3.2.2001)
- [Russo und Boor, 1993] Russo, Patricia; Boor, Stephen: "How Fluent is Your Interface? Designing for International Users". (in: ACM: "Proc. CHI '93 Human Factors in Computing Systems"). ACM, New York, 1993
- [Salomon, 1991] Salomon, Gitta: "New Uses for Color". (in: Laurel, Brenda: "The Art of Human-Computer Interface Design"). Addison Wesley, Reading, MA, 3. Auflage, 1991
- [Sellen und Nicol, 1991] Sellen, Abigail; Nicol, Anne: "Buiding User-centered On-line Help". (in: Laurel, Brenda: "The Art of Human-Computer Interface Design"). Addison Wesley, Reading, MA, 3. Auflage, 1991
- [Shneiderman, 1998] Shneiderman, Ben: "Designing the User Interface: Strategies for Effective Human-Computer Inter". Addison Wesley, Reading, MA, USA, 3rd ed., 1998
- [Tognazzini, 1992] Tognazzini, Bruce: "Tog on Interface". Addison-Wesley, Reading, MA, 1992
- [Tognazzini, 1996] Tognazzini, Bruce: "Tog on Software Design". Addison Wesley, Reading, MA, 1996
- [Tovée, 1996] Tovée, Martin J.: "An Introduction to the Visual System". Cambridge University Press, Cambridge, 1996
- [Tufte, 1984] Tufte, Edward R.: "The Visual Display of Quantitative Information". Graphics Press, Cheshire, CO, 1984
- [Tufte, 1990] Tufte, Edward R.: "Envisioning Information". Graphics Press, Cheshire, CT, 1990
- [Volpert, 1979] Volpert, Walter: "Der Zusammenhang von Arbeit und Persönlichkeit aus handlungspsychol. Sicht". (in: Groskurth, Peter: "Arbeit und Persönlichkeit"). Rowohlt, Reinbek, 1979



- [Wildstrom, 1998] Wildstrom, Stephen H.: "A COMPUTER USER'S MANIFESTO". <http://www.businessweek.com/1998/39/b3597037.htm> (Aufruf am 14.11.2000)
- [Zwicker und Feldtkeller, 1967] Zwicker, Eberhardt; Feldtkeller, Richard: "Das Ohr als Nachrichtenempfänger". S. Hirzel Verlag, Stuttgart, 1967

## Anhang

### Benutzerbefragung

Dieser Abschnitt enthält stichwortartige Protokolle der Befragungen. Befragt wurden folgende Benutzer:

1. Weiblich, 28 Jahre. Informatik-Studentin. Viel Computererfahrung.
2. Männlich, 33 Jahre. Programmierer. Viel Computererfahrung.
3. Weiblich, 59 Jahre. Sachbearbeiterin. Langjährige PC-Nutzerin, jedoch nur wenig Hintergrundwissen.
4. Männlich, 67 Jahre. Handwerkermeister. Computer-Neuling.

Benutzer 3 und 4 wurden gemeinsam befragt, die beiden anderen Benutzer einzeln. Die Protokolle der Befragung sind nach den beiden Themenbereichen sortiert.

### Entwürfe für Fortschrittsanzeigen

Ich habe die verschiedenen Entwürfe vorgestellt und dabei die Benutzer gefragt,

- wie ihnen die Entwürfe gefallen,
- wie gut sie sich zum Schätzen der verbleibenden Zeit eignen,
- mit welchem von zwei konkreten Entwürfen die verbleibende Zeit leichter schätzbar ist,
- ob einzelne Merkmale eines Entwurfs (Farbe, Klang) gegenüber dem Standard beim Schätzen helfen.

*Benutzer 1:* Benutzerin meinte, man könne bei Kreisen besser schätzen: Viertel, Hälfte, drei Viertel. Aber bei Balken würde ja die %-Zahl immer dabei stehen. Benutzerin mag segmentierte Kreise ohne Trenner lieber, meint aber, mit Trennern wäre es noch eine Winzigkeit einfacher. Die Zeiger hat sie nicht verstanden, erst dann, als sie die Prozentangaben dazu gelesen hatte. Der Smiley ist zwar schlecht zum Schätzen, aber witzig. Bei der Formwandlung ist keine Schätzung möglich, weil das Ende nicht abzusehen ist. Nach mehrmaligem Gebrauch wird es etwas besser, dann kann wenigstens die 50% geschätzt werden. Die Schätzung für den zunehmenden Kreis ist schwierig, mit Markierungen ist wenigstens das Schätzen von Vierteln möglich. Töne zwischendurch sind eher überflüssig, der Ton am Ende reicht. Wenn zwischendurch Töne verwendet werden, sollten sie unterscheidbar sein. Insgesamt sollten Töne aber eher fortgelassen werden.

*Benutzer 2:* Benutzer hat die Jahreszeiten nicht erkannt. Fragt, was die Ampel zu bedeuten hat: Bei Grün kann gefahrlos abgebrochen werden? Kreis hat was für sich, spontane leichte Begeisterung. Kreise sind ohne Trennung zwischen den Segmenten netter anzusehen, die Striche stören. Ohne Striche hat eine kreisförmige Anzeige was von einer Uhr. Ob mit oder ohne Striche, es lässt sich gleich gut schätzen. Begeisterung beim Farbwechsel in vier Phasen. Vierfarbig sind zu viele Farben auf einmal. Countdown ist gegen den Uhrzeigersinn besser, da diese Richtung umgekehrt zum üblichen Füllen ist und so in einem logischen Zusammenhang steht. Es wäre verwirrend, wenn in einem Programm für Zunahme und Abnahme die gleiche Drehrichtung verwendet wird. Der Kreis ist zum Schätzen besser geeignet als ein Balken. Hat kein Vertrauen zum Balken, da zu viele schlechte Erfahrungen – Kreis ist neu und hat damit einen Vertrauensvorsprung. Ein Kreis ist unterhaltsamer, ist etwas Neues, ist eine Abwechslung. Ein Tacho zeigt eine Geschwindigkeit, nicht die zurückgelegte Strecke: Passt nicht zum Anzeigen von Fortschritt. Die Zeiger sind nicht auf den ersten Blick verständlich, es müssen erst die Zahlen gelesen werden. Der Benutzer meint, er mag Segmente nicht, weil sie zum Zählen und Rechnen verleiten, ein kontinuierlicher Zuwachs wäre besser, wenn er möglich ist. Den Formwandler fand er zwar interessant, aber meinte, das Schätzen würde damit sehr schwer fallen. Beim ersten Mal wisse man nicht, wohin die Reise geht, ob Farbe oder Form wichtig sind. Er meint, man könne sich entweder nur auf die Farbe oder die Form konzentrieren, das andere würde verloren gehen. Der Ring gefällt nicht, weil es dazu keine Metapher gibt. Auch nach mehrmaligem Gebrauch ist der Formwandler schlecht zum Schätzen geeignet. Den Smiley findet der Benutzer lustig. Er kann gut für das Schätzen der Viertel verwendet werden. Die Größenänderung ist keine gute Idee, da die Flächenänderung nicht linear zum Prozessfortschritt verhält. Genaue Schätzungen sind nicht möglich. Markierungen erleichtern das Schätzen, aber sie müssen gezählt werden und sind nicht intuitiv erfassbar. Akustische Signale empfand der Benutzer eher als lästig, insbesondere während des Prozesses – schließlich könne man den Fortschritt sehen. Auf Nachfrage konnte der Benutzer sich vorstellen, dass zumindest bei einigen Minuten dauernden Prozessen ein akustisches Signal hilfreich sein kann, insbesondere dann, wenn die optische Anzeige verschwindet.

*Benutzer 3:* Die Kreise sind zu groß, nehmen zu viel Fläche ein. Der Tacho wird abgelehnt. Ein rechtsdrehender Zeiger ist besser als ein linksdrehender. Die Drehrichtung sollte immer der Uhrzeigersinn sein, das ist der Mensch gewohnt. Spontane Antwort auf die Frage, welche Form am besten zum Schätzen geeignet ist: Der Balken. Nach etwas Überlegung wurde der Kreis bevorzugt, weil die einzelnen Viertel einfach zu erkennen sind. Dafür braucht man auch keine numerische Prozentangabe. Ein segmentierter Balken ist besser zu schätzen als ein kontinuierlicher.

Ein Kreis ohne Trennstriche ist ruhiger. Die Trennstriche helfen nicht beim Schätzen, sie sind eher unangenehm. Der Jahreszeiten-Balken wurde spontan abgelehnt. Die Benutzerin stimmte der Aussage von Benutzer 4 zu, dass der Entwurf *Näher kommend* nichts signalisiert. Sie stimmte auch seiner Aussage zu, es sollte „*ein Bezug zum täglichen Leben*“ vorhanden sein. Der Balken mit drei verschiedenen Farbphasen gefiel ihr nicht so gut wie der mit drei verschiedenfarbigen Segmenten. Dort wäre leicht zu erkennen, dass bereits ein Drittel vorbei ist, weil die Einteilung in die Drittel immer sichtbar ist. Den Balken mit kontinuierlichem Farbwechsel von Rot über Gelb nach Grün fand sie „*blöd*“. Wenn die Farbe des kompletten Balkens sich ändert, hilft Farbe nicht beim Schätzen. Verschiedenfarbige Abschnitte sind besser. Auf den Entwurf *Näher kommend* angesprochen meinte sie, wenn eine Idee erst erklärt werden muss, kann man sie sich sparen. Bei Kreisen ist eine Farbänderung überflüssig, da die Viertel gut zu erkennen sind. Farbänderungen würden eher irritieren. „*Wozu soll eine dritte Farbe rein?*“ [gemeint sind beim zweifarbigen Kreis die beiden Grüntöne sowie das Weiß]. Vierfarbig macht in ihren Augen mehr Sinn, da alle Viertel hervorgehoben werden. Ein Farbwechsel für den ganzen Kreis nach jedem Viertel (Entwurf *Kreis, Farbwechsel*) ist überflüssig. Ein Countdown sollte im Uhrzeigersinn erfolgen. Diese Drehrichtung ist bekannt, sie wirkt ruhiger. Der zusätzliche Einsatz von Farbe beim Countdown unterstützt nicht, sondern irritiert eher. Auf einen Vergleich zwischen Kreis und Balken befragt, antwortete die Benutzerin, dass der Kopf beim Kreis nicht überlegen muss, die Viertel sind bekannt und überschaubar. Bei einer Strecke kann man sich leicht verschätzen. Auch spielt beim Kreis die Größe keine Rolle, ein Viertel ist immer gleich gut zu erkennen. Verschiedene Längen beim Balken machen es jedoch unterschiedlich schwer. Ihre Wunschanzeige ist ein kontinuierlich („*ruhig*“) fließender Kreis ohne Trennstriche. Eventuell könnten Linien für die Einteilung der Viertel ständig sichtbar sein („*dann weiß ich, wie weit ich über's Viertel bin*“). Wenn ein Balken verwendet wird, sollte er vier unterscheidbare Abschnitte haben, damit die Viertel leicht erkannt werden. Drittel sind schwieriger einzuschätzen. (Sieht den Smiley und fängt an zu lachen) „*Der ist niedlich!*“ Der Smiley ist ihrer Meinung nach auflockernd. In Bezug auf die Möglichkeit zum Schätzen meinte sie, man wisse zwar, dass er die Mundwinkel nach oben ziehe, aber nicht sehen, wann es zu Ende ist – nur mehr oder weniger als die Hälfte könne erkannt werden. Bei der Formwandlung weiß man nicht, was das Ergebnis sein wird. Auch bei mehrmaligem Sehen ist immer noch schlecht einzuschätzen, wie viel bereits vorbei ist und wie lange es noch dauert. Außerdem muss man sich merken, was mit Farbe und Form passiert. Den größer werdenden Kreis empfand die Benutzerin als unangenehme Bewegung. Das Abschätzen des Vorgangs fiel ihr wegen der ungewohnten Bewegung schwer. Die Markierungen sind zwar eine Hilfe für das

Schätzen, aber insgesamt ist das kein angenehmer Pausenfüller. Bei den ersten akustischen Signalen kam sofort der Kommentar, dass man um Himmels Willen nicht noch mehr Geräusche ins Büro holen sollte. Außerdem sei es zu viel, dass man sich dann sowohl auf die Optik als auch auf die Akustik konzentrieren soll. Die akustische Erinnerung durch Töne macht ihrer Ansicht nach eher nervös.

*Benutzer 4:* Fragt nach dem Vorführen der Formen spontan nach einem kontinuierlichen Kreis mit zwölf Trennstrichen. Der Ring wird eher abgelehnt („*Ich weiß nicht recht...*“). Den Tacho fand er nicht so schlecht, aber die Sprünge störten ihn. Bei dem rechtsdrehenden Zeiger kam spontan die Idee, ob nicht alle 10% der Wert daneben geschrieben werden sollte. Am besten zu schätzen sind der Ring oder der Kreis, da die Waagerechte und die Senkrechte leicht zu erkennen sind. Eine zusätzliche numerische Angabe in % ist immer gut, sie verwirrt nicht. Ein segmentierter Balken ist besser zu schätzen als ein kontinuierlicher. Aufgrund eines Darstellungsfehlers (die waagerechte Linie des segmentierten Kreises mit Trennstrichen erschien dicker als die anderen Trennstriche) kam die Idee, die Viertel durch dickere waagerechte und senkrechte Linien zu betonen. Der kontinuierliche Kreis wirkt schneller als der segmentierte. Der Entwurf *Näher kommend* signalisierte dem Benutzer nichts. Die verwendeten Farben sollten einen Bezug zum täglichen Leben haben. Der Benutzer zieht einen Farbwechsel für den kompletten Balken verschiedenfarbigen Segmenten vor. Der Balken Kontinuierliche Ampel gefiel ihm von allen Balken am besten. Farbe für den Balken hilft beim Schätzen. Mit Bezug auf *Näher kommend* und *Jahreszeiten* meinte der Benutzer, dass man alles sofort ohne Erklärung verstehen sollte. Bei den farbigen Kreisen regte er an, mit einer hellen Farbe anzufangen und dann zu dunkleren Farben zu wechseln: Der leere Kreis sei schließlich weiß, und er würde dann immer mehr eingefärbt. Grün sollte am Ende sein, da es „Fertig!“ signalisiert. Ein Countdown sollte im Uhrzeigersinn erfolgen. Auf Nachfrage meinte der Benutzer, dass Farbe beim Balken das Schätzen erleichtert, bei Kreisen nicht, weil Balken nicht von vornherein eingeschätzt werden können. Ein Kreis ruft Assoziationen zu Zeit und Zeitablauf hervor. Seine Wunschanzeige ist ein kontinuierlicher Kreis, ohne Farbänderungen oder Trennstriche. Beim Kreis ist eine Schätzung einfach bequemer zu machen als bei einem Balken. Der Smiley ist zwar niedlich, aber nicht seriös und zweckmäßig. Aber vielleicht passt er zu bestimmten Gebieten. Die Formwandlung ist eher als negatives Beispiel geeignet. Der zunehmende Kreis ist schlecht zu schätzen, es ist nur gut zu sehen, wann er fertig ist. Markierungen als „*Hilfskrücke*“ helfen beim Schätzen, aber die vielen Kreise verwirren. Ein Klang am Ende ist gut, wenn man gerade nicht auf den Bildschirm sieht. Töne können gut genutzt werden, um die Aufmerksamkeit (wieder) zu erregen. Wenn mittendrin Töne verwendet werden, sollten sie unterschiedlich sein. Daher gefiel ihm der Viertelstunden-Schlag gut.

## Entwürfe für Datei-Kopieren-Dialog

Den Benutzern wurden jeweils zwei Beispiele gezeigt. Zuerst wurden die beiden Balken gezeigt, danach der einfache Balken und der Kreis, anschließend der segmentierte Balken und der Kreis. Ich habe die Benutzer gebeten, die Beispiele anhand folgender Fragen zu kommentieren:

- Wird die angebotene Information verstanden?
- Ist zu sehen, dass das System noch arbeitet?
- Wie gut kann geschätzt werden, wie lange der Vorgang noch dauert?
- Fehlen Informationen?
- Kann auf einige der angebotenen Informationen verzichtet werden?

*Benutzer 1:* Hat offensichtlich alle angebotenen Informationen bei allen drei Varianten verstanden. Die durchlaufenden Zahlen zeigen ihr, dass das System noch arbeitet. Eventuell möchte sie noch mehr Informationen zum Gesamtprozess haben, beispielsweise die gesamte Größe aller Dateien und der hiervon bereits geladene Teil. Das wären vermutlich nicht zu viele Informationen, aber essenziell wichtig wären sie nicht. Wichtiger als die Zahlen ist ihr die Grafik, weil hier ein schnelles Erfassen möglich ist, die Zahlen sollten aber zusätzlich für genauere Information vorhanden sein. Die Informationen zu der aktuell kopierten Datei sind nicht unbedingt notwendig, schaden aber auch nicht. Die Details sind ein netter Pausenfüller, wenn es länger dauert. Bei kurzen Vorgängen reicht ihr die Grafik, weil die Zeit nicht zum Lesen der Zahlen reicht. Bei längeren Vorgängen (gemeint sind etliche Minuten) sind die Zahlen wichtig, da die Grafik sich zu selten ändert. Die verschiedenen Farben beim segmentierten Balken fand sie störend, auch wenn die einzelnen Viertel dadurch besser zu erkennen sind. Ihr wurde die Anzeige zu unruhig. Sie zieht eine Lösung mit einem Balken der Lösung mit dem Kreis vor, kann es aber nicht begründen. Die Zahlen sind bei den Balken-Lösungen besser zuzuordnen als bei der Kreis-Lösung. Zum Schätzen sind markierter Balken und der Kreis gleich gut geeignet.

*Benutzer 2:* Der Benutzer hat alle die Bedeutung aller angebotenen Informationen verstanden. Die durchlaufenden Zahlen und die sich ändernde Grafik zeigen ihm, dass das System arbeitet. Den segmentierten Balken mag der Benutzer nicht so gerne, da die Segmente ihn dazu verleiten nachzurechnen, wie viel Zeit für ein Segment benötigt wird. Kreise sind am besten zum Schätzen geeignet. Aber die Uhr-Metapher und eine Restzeit passen seiner Meinung nach nicht zusammen. Die Details sind ihm wichtig, da er dann sehen kann, was das System macht und eine Kontrollmöglichkeit hat, ob er die korrekten Dateien gewählt hat. An zusätzlichen Informationen wollte

der Benutzer die Anzahl der Dateien und ihre aufsummierte Größe sowie von beiden Werten den bereits erledigten Teil. Eventuell möchte er auch einen Balken für den Teilprozess haben. Auch auf mehrfache Nachfrage konnte er jedoch keinen Vorteil nennen, den ihm ein zweiter Balken bringen würde. Nach etwas Überlegung meinte er, ein zweiter Balken würde die Anzeige eventuell zu unruhig gestalten, das würde man schließlich bei Windows sehen.

*Benutzer 3:* Dass etwas passiert, sagen die durchlaufenden Zahlen. Die Markierungen beim segmentierten Balken sollten bleiben, die Farben nicht. Der Balken sollte hell sein, damit die Markierungen ständig sichtbar sind. Die Benutzerin kann insgesamt auf den Balken verzichten, Zahlen würden ihr reichen – sie könne dann ja rechnen. Die Namen der Dateien braucht sie nicht. Beim Vergleich von Balken und Kreis stellte sie fest, dass der Balken eine Einteilung in Viertel braucht, die ein Kreis implizit hat. Bei den Lösungen mit Balken hat man alle Informationen immer im Blick. Bei der Lösung mit dem Kreis muss das Auge wandern, entweder hat man die Angaben über den gesamten Prozess oder die Details im Blick. Kreis und Texte sind quasi zwei Spalten.

*Benutzer 4:* Die Frage, ob das System noch arbeitet, wurde von ihm spontan als die wichtigste Frage herausgestellt. Die sich ändernden Zahlen zeigen, dass das System noch arbeitet. Wichtig ist der Gesamtprozess, die Details fand der Benutzer auf Nachfrage eher unwichtig. Vielleicht könnte man noch angeben, wie viele Dateien insgesamt kopiert werden, aber richtig wichtig ist das nicht. Hauptsache, die Zahlen bewegen sich, so dass man sieht, dass noch etwas passiert. Der mehrfarbige Balken liefert ihm zu viel Informationen. Texte sind generell schwer zu lesen, eine Grafik ist einfacher zu erfassen.

## **Inhalt der CD**

Im Verzeichnis *Beispiel* liegt die Datei *index.html*, die einen Überblick über die Beispiele gibt und einen einfachen Zugriff auf sie ermöglicht. Die Beispiele selbst liegen in den Unterverzeichnissen *Balken*, *Kreise*, *Form*, *Akustik* und *Kopieren*. Zu jedem Beispiel gehören eine HTML-Datei für den Aufruf der SWF-Datei, die SWF-Datei mit dem kompilierten Film sowie eine FLA-Datei mit dem Flash-Source.

Im Verzeichnis *Bilder* sind sämtliche Abbildungen aus dem Text enthalten. Der Name der Datei ist *Bild*, gefolgt von der Nummer der Abbildung. Bei mehrteiligen Abbildungen folgt noch ein weiterer Buchstabe zur Identifikation des Teils. Mehrfach im Text verwendete Bilder sind auch mehrfach auf der CD enthalten.

# Abbildungen in Farbe



Abbildung 28



Abbildung 29



Abbildung 30

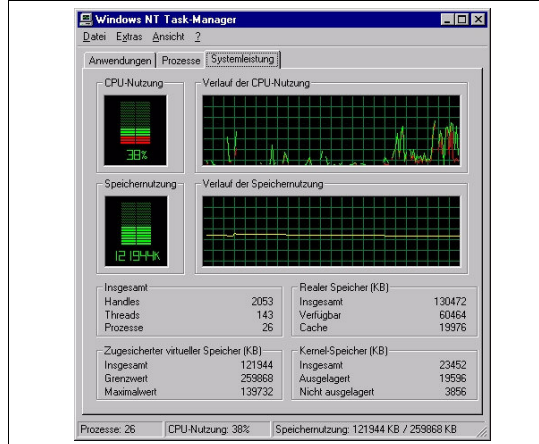


Abbildung 32

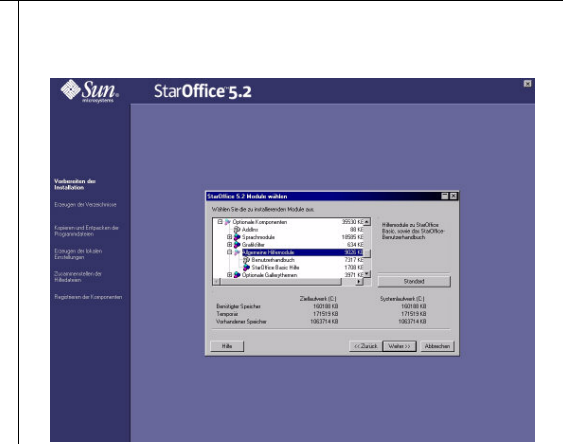


Abbildung 34

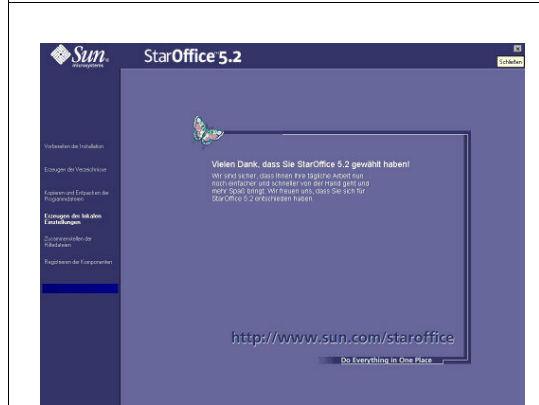


Abbildung 37

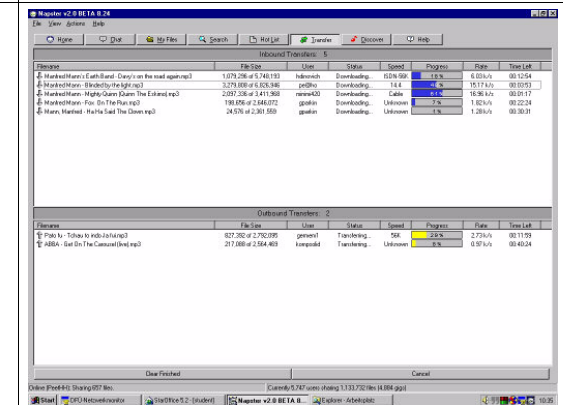


Abbildung 48

Autor	Titel			Erstellungsdatum	Priorität
Peter Thielmann	Markierte Mail			15.01.01, 15:18:06	3 (Normal)
Peter Thielmann	Ungelesene Mail			15.01.01, 15:1...	3 (Normal)
Peter Thielmann	Gelesene Mail			15.01.01, 15:17:39	3 (Normal)

Abbildung 61



Abbildung 64



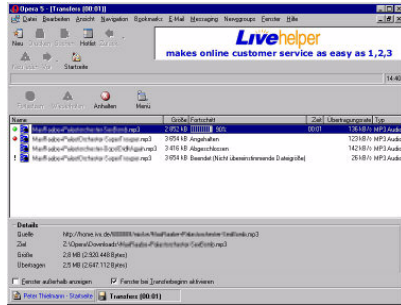


Abbildung 71



Abbildung 72



Abbildung 73



Abbildung 74



Abbildung 76

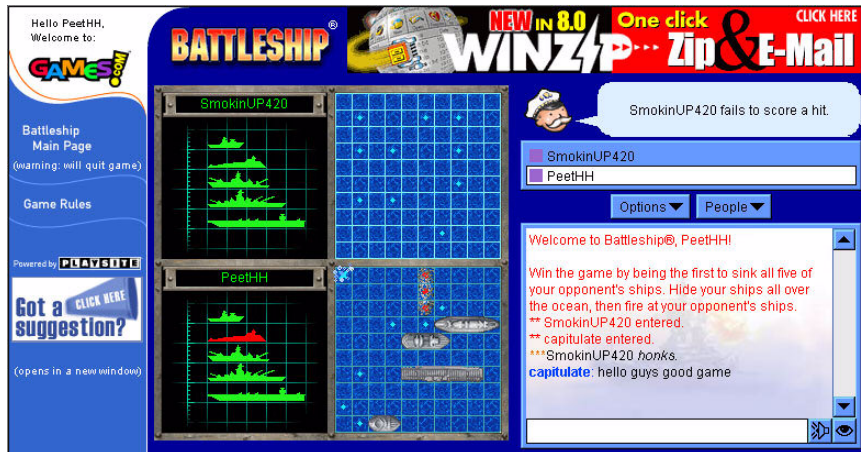


Abbildung 77



Abbildung 81



Abbildung 82



Abbildung 83



Abbildung 84



Abbildung 89



Abbildung 90



Abbildung 91



Abbildung 110