

Europa-Universität Viadrina Frankfurt an der Oder

Fachbereich Kulturwissenschaft

Sommersemester 2002

PS: Geschichte des Design

Dozent: Prof. Dr. Christoph Asendorf

# Interface Design

**Die Gestaltung der Mensch- Maschine-Schnittstelle  
als neue Design-Theorie des Computerzeitalters**

vorgelegt von  
Christopher Paun  
Mat. Nr. 9409  
Berlin, Mai 2003

Christopher Paun  
Gabriel-Max-Str. 1  
10245 Berlin  
0700 – call-a-paun  
mail@paun.de  
www.paun.de

# Inhaltsverzeichnis

1. Einleitung.....	4
2. Begriffsklärung und Abgrenzung .....	6
3. Historische Entwicklung des Mensch-Computer-Interface.....	9
3.1 Xerox Star (1981).....	10
3.2 Apple Macintosh (1984).....	12
3.3 MS-Windows und X-Windows (1985).....	13
3.4 Personifizierte Agenten (1989).....	14
3.5 Das World Wide Web (1992) .....	15
3.6 Sprach- und Handschrifterkennung (1996) .....	16
3.7 Wireless Application Protocol (2000) .....	17
4. Prinzipien des Interface Design .....	18
4.1 Transparenz.....	18
4.2 Beschränkung auf das Wesentliche .....	19
4.3 Redundanz.....	20
4.4 Undo- / Storno-Funktion.....	20
4.5 Systemzustände sichtbar machen .....	21
4.6 Einheitlichkeit .....	21
4.7 Intuitive Bedienung durch Metaphern.....	22
4.8 Entlastung des Gedächtnisses.....	23
4.9 Unmittelbares Feedback .....	24
4.10 Fehler erkennen und erklären.....	24
4.11 Kontrolle muss stets beim Benutzer liegen .....	25
5. Bewertung .....	26
6. Literaturverzeichnis .....	28

## Abstract

The present paper concerns the interface design of human-computer-interface as a new theory of design in the information age. After defining interface design and showing the borders of the new theory in chapter two, a historic overview of the development of the human-computer-interface is given in chapter three. Thereby milestones which influenced the interface design are shown exemplary. In chapter four the principles of the interface design are shown. But as there is no consistent set of principles, also the borders of these principles and the conflicts of aims in-between these principles are shown.

## Abbildungen

Abb. 1: Screenshot von Microsoft MS-DOS	9
Abb. 2: Screenshot von Xerox Star. Quelle: Preim 1999: 40	10
Abb. 3: Titelseite des Computermagazins PC-Welt von 1983. Titelthema: „Fenster und Mäuse - Rechner werden ergonomischer“. Quelle: <a href="http://www.pcwelt.de//graphics/bild_db/16600/16670/original.idg">www.pcwelt.de//graphics/bild_db/16600/16670/original.idg</a>	11
Abb. 4: Screenshot vom Apple Macintosh Desktop. Quelle: Apple 1987: 21	12
Abb. 5: Screenshot von MS-Windows in der Version XP von 2002	14
Abb. 6: Screenshot des Hilfsprogramms „Karl Klammer“ von Microsoft MS-Office	15
Abb. 7: Screenshot des Webkatalogs Yahoo	15
Abb. 8: Suche nach einem nahegelegenen Restaurant mit einem WAP-Handy. Quelle: <a href="http://www-106.ibm.com/developerworks/ibm/library/i-lbs/">www-106.ibm.com/developerworks/ibm/library/i-lbs/</a>	17
Abb. 9: Screenshot von Microsoft WordPad	18
Abb. 10: Screenshot von Microsoft MS-Word	19
Abb. 11: Screenshot von Psion Agenda	22
Abb. 12: Screenshot des Dialogfensters „Zeichen“ von MS-Word	23
Abb. 13: Verschiedene Mauscursor. Quelle: <a href="http://www.it-resources.icsa.ch/SoftWare/InterFaceD.html">www.it-resources.icsa.ch/SoftWare/InterFaceD.html</a>	24
Abb. 14: Beispiel für ein Fenster ohne Handlungsoptionen	25

# 1. Einleitung

„Die intelligenten Maschinen arbeiten sich in der Unternehmenshierarchie immer weiter nach oben und übernehmen nicht nur Routinetätigkeiten, sondern auch typische Managementaufgaben. Dies geht soweit, dass in Hunderten US-amerikanische Unternehmen neuerdings Computersysteme für die Einstellung neuer Mitarbeiter zuständig sind. Ein System namens Resumix speichert über einen Scanner täglich 400 Bewerbungen in einer Datenbank, die nicht größer als ein kleiner Aktenschrank ist. Innerhalb von drei Sekunden ist die Bewerbung aufgenommen, dann druckt Resumix den passenden Bestätigungsbrief an den Absender aus.“ (Rifkin 2001: 14)

Dieses Zitat aus „Das Ende der Arbeit“ verdeutlicht recht gut, welche Ängste mit der Einführung von Computern verbunden sind. Hier wird suggeriert, dass eine so wichtige Entscheidung wie eine Einstellung, die für das Schicksal eines einzelnen Menschen so entscheidend sein kann, nicht mehr von einem Personalchef aus Fleisch und Blut getroffen wird, sondern von einem „kaltherzigen Computersystem“. Tatsächlich hat das Computer-System aber keinen eigenen Willen, sondern leistet eine Vorsortierarbeit, anhand von Kriterien die vorher vom Personalchef festgelegt wurden. Es ist also ein Werkzeug und keine Person.

Dennoch sprechen viele Menschen von Computersystem als wären sie Personen, weil die Systeme so schnell, komplex und intransparent sind, dass sich ihnen der Werkzeugcharakter nicht erschließt. Wenn die Aktion eines Computersystems für den Menschen nicht vorhersehbar ist, kann er es nicht als Werkzeug begreifen, das von ihm benutzt wird, sondern betrachtet es eher als ein Orakel.

Ob man einen Telefonanschluß beauftragen möchte oder eine Fahrkarte kaufen – zu oft schieben die Angestellten die Verantwortung von sich auf den Computer. „Der Computer lässt eine Kombination dieser Optionen nicht zu – ich kann ihnen da leider nicht weiter helfen“, könnte eine solche Aussage zum Beispiel lauten. In der Alltagssprache wird den Geräten sogar ein eigener Wille zuerkannt, wenn Leute davon sprechen, dass ihr Computer „mal wieder nicht so recht wollte“. Und zu oft beschimpfen Menschen ihre Computer und reden auf sie ein, als würde hinter den Schaltkreisen ein Geist wohnen, dem man nur gut zureden muss.

Jenseits dieser populären Vorstellungen, gibt es natürlich noch wissenschaftliche Forschung zum Thema künstliche Intelligenz. Die Hoffnungen liegen hier auf selbstlernenden Computern. Falls diese Computer jemals zu natürlicher Kommunikation fähig sein sollten, hätten sie höchst wahrscheinlich etwas ausgebildet, das wir als Persönlichkeit bezeichnen müssten (vgl. Turing 1994), was uns vor ganz andere moralphilosophische Fragen stellen würde, wie z.B. ob wir diese Wesen versklaven dürften. Derartige Vorstellungen liefern auch Stoff für Science Fiction Autoren. Eine weitergedachte Horrorvision wäre zum Beispiel eine Welt, wie sie Waldmann (1971) in dem Film „Dreht euch nicht um, der Golem geht um“ dargestellt hat. Eine Welt in der es keine Möglichkeit mehr für den Menschen gibt sich sinnvoll zu beschäftigen, weil alle körperliche und geistige Arbeit von Maschinen viel effizienter verrichtet wird und selbst die Herrschaft von einem Computer ausgeübt wird. Alle bisherigen Experimente mit Künstlicher Intelligenz sind jedoch gescheitert und ein Durchbruch auf diesem Gebiet ist auf lange Zeit nicht in Sicht. Daher sollten Computer so gestaltet werden, dass sie als Werkzeug erkennbar sind. Hierbei liegt das Augenmerk auf der Gestaltung der Schnittstelle zwischen Mensch und Maschine, dem Interface Design.

In dieser Arbeit soll der gegenwärtige Stand der Theorie zum Thema Interface Design zusammengefasst werden. In Kapitel zwei soll zunächst eine Begriffsklärung und eine Abgrenzung zu anderen Bereichen erfolgen. Anschließend wird in Kapitel drei die historische Entwicklung des Interface Design dargestellt. Dabei werden Interfaces exemplarisch vorgestellt, die als Meilenstein in der Entwicklung angesehen werden können. In Kapitel vier erfolgt schließlich eine Darstellung der aktuellen Regeln, die es in der Theorie zum Interface Design gibt. Da es sich nicht um ein Regelwerk aus einem Guss handelt, müssen hier auch Grenzen und Zielkonflikte der jeweiligen Regeln dargestellt werden. Die Ausführlichkeit und Tiefgründigkeit des vierten Kapitels wird designtheoretischen Erwartungen leider nicht ganz gerecht. Dies liegt daran, dass sich die Literatur hauptsächlich an ein eher praktisch orientiertes Publikum aus Informatik, Screen Design oder Computervisualistik richtet. Die Entstehung der Theorie zum Interface Design ist noch in einem sehr jungem Stadium.

## 2. Begriffsklärung und Abgrenzung

In der Entstehungszeit des Interface Design in den 80er Jahren des zwanzigsten Jahrhunderts wurde Design oftmals mit oberflächlicher Kosmetik gleichgesetzt. Und oftmals wurde der Designer erst sehr spät in den Entwicklungsprozess eines neuen Produktes eingebunden. Nachdem die Ingenieure schon das eigentliche Produkt konstruiert hatten, war es die Aufgabe des Designers, sich um das Make-up zu kümmern. Er gestaltet dann die Verkleidung des Produktes nach modischen Aspekten. Auch wenn es schon damals gute Beispiele für Ausnahmen von dieser Herangehensweise gab, war doch dies bis in die neunziger Jahre das vorherrschende Berufsbild des Designers. Design wurde daher oftmals als teuer, unpraktisch und kurzlebig angesehen. Entsprechend wenig Theoretisierung gab es daher im Design. Es handelte sich um eine „Domäne ohne Grundlagen, in der small talk gepflegt [wurde].“ (Bonsiepe 1996: 24)

Eine wesentliche Ausnahme von dieser mangelnden Theoretisierung gab es allerdings bereits auf dem Gebiet der Ergonomie. Die Maße des menschlichen Körpers wurden statistisch erfasst und die Funktion seines Bewegungsapparates wurden erforscht. So war es möglich beispielsweise Schreibtische und Stühle zu konstruieren, die nicht nur gut aussahen, sondern an denen man auch möglichst lange ohne körperliche Beschwerden arbeiten konnte. Besonders wichtig sind Fragen der Ergonomie bei Gebrauchsgegenständen, die einen Werkzeugcharakter haben. Bei solchen Werkzeugen ist nicht nur die physikalische Effizienz entscheidend, an der die Ingenieure arbeiten, sondern auch die Gebrauchseffizienz.

Am Beispiel einer Schere kann man diesen Unterschied sehr gut erläutern: Die beiden Schneiden der Schere sind der Ingenieurspart. Ihr Material, ihr Schliff und ihre mechanische Bewegungsanordnung bestimmen die physikalische Effizienz, mit der die Schere Papier oder ein anderes gewünschtes Material schneiden kann. Die Gestaltung der Griffe hingegen bestimmt, wie effizient das Werkzeug von einem Menschen benutzt werden kann. Die Griffe sind daher die Schnittstelle bzw. das Interface und ihre Gestaltung fällt in die Domäne des Interface Design. Es ist beispielsweise sinnvoll, die Griffe so zu gestalten, dass man sie mit einer Hand bedienen kann, um mit der anderen Hand noch das zu schneidende Objekt zu

halten. Man sollte auch wissen, dass es dem Menschen besonders leicht fällt, den Daumen getrennt von den anderen Fingern zu bewegen, während beispielsweise die zielgerichtete und getrennte Bewegung von kleinem Finger und Ringfinger sehr schwer fällt. Und zu guter Letzt ist es ratsam, einen Griff der Schere so groß zu gestalten, dass man zwei Finger hinein stecken kann, um die Schere sicherer führen zu können. All diese Eigenschaften einer Schere sind für uns selbstverständlich, doch bei neuen oder komplexeren Werkzeugen kann eine hohe Gebrauchseffizienz nur mit einer gewissen Vorarbeit erreicht werden. Die Anforderungen müssen zunächst erforscht werden.

Jenseits der Ergonomie, die sich mit den körperlichen Gegebenheiten des Menschen befasst, gibt es aber auch eine Ergonomie, die man als Informations-Ergonomie bezeichnen könnte. Im Englischen verwendet man hier den Begriff der „Affordances“, für den es leider keine deutsche Übersetzung gibt. Es handelt sich hierbei um Handlungsmöglichkeiten, die aufgrund des Aussehens eines Objektes impliziert oder vermutet werden. Preim (1999: 14) bezieht sich auf die These aus der Wahrnehmungspsychologie, dass der Mensch seine Umwelt nach solchen Affordances wahrnimmt. Bei dem Beispiel mit der Schere, wäre diese Affordance-Wahrnehmung kein Problem, da die Kenntnis dieses Werkzeug zu unserem kulturellen Wissen gehört. Es wäre auch sicher schwierig, irgendwo auf der Welt eine Versuchsperson zu finden, die eine Schere nicht kennt und sich ihre Funktion erst erschließen muss. Doch auch für diesen Fall, wäre es sicher kein Problem, die Funktionsweise der Schere zu verstehen, da sie ein sehr transparentes und wenig komplexes Werkzeug ist.

Einen wesentlichen Einfluss auf die Affordances der Objekte unserer Umwelt hatte die Entwicklung der Computertechnik. Vor der Erfindung des Computers haben Werkzeuge nur mechanische Aufgaben erleichtert. Ihre Funktion hat sich daher auch relativ leicht bei der Betrachtung des mechanischen Bewegungsablaufes erschlossen. Das Interface musste nur an die Physis des Menschen angepasst werden, wozu die Regeln der Ergonomie nötig sind. Der Computer ist allerdings ein Werkzeug für geistige Aufgaben und daher muss das Interface an die Psyche angepasst werden. Dies ist eine grundlegend neue Aufgabe für den Designer, zu deren Bewältigung Erkenntnisse aus der Wahrnehmungspsychologie nützlich sind.

Gleichzeitig bietet das Interface eine solide Legitimationsbasis für das Design, die unanfechtbarer ist, als reines Make-up oder der hohe Anspruch Kunstwerke zu schaffen. Da besonders in den 80er Jahren das Design in die Nähe der Kunst gerückt wurde, hat eine Gegenbewegung unter den Designern die entstehende Computertechnik dankbar aufgenommen und sich dem Interface Design zugewandt. Interface Design ist also eine Bewegung, die sich einerseits vom künstlerisch verstandenen Design und andererseits von den Ingenieurwissenschaften und der Informatik abgrenzt.

Bevor der Begriff des Interface Design aufkam, wurde dieser Gestaltungs-Bereich oftmals als Software-Ergonomie bezeichnet oder einfach unter dem Überbegriff Ergonomie subsumiert. (vgl. Abb. 3) Das Interface Design ist allerdings mehr als nur Ergonomie für die Psyche, da Computer ohne Interface nicht eine schlechte Effizienz hätten, sondern gänzlich unbrauchbar wären. Eine CD-ROM beispielsweise kann nur als eine solche erkannt werden, wenn man weiß, dass es sich um eine handelt. Genutzt werden kann sie hingegen auch von einem Fachmann nur mit einem entsprechendem Interface. Ohne Bildschirm, Tastatur, Software usw. wären die Informationen von über eintausend Büchern, die man auf einer CD-ROM speichern könnte, völlig unzugänglich und nutzlos.

„Das Interface erschließt den Werkzeugcharakter von Objekten und den Informationsgehalt von Daten. Interface macht Gegenstände zu Produkten. Interface macht aus Daten verständliche Informationen. Interface macht aus bloßer Vorhandenheit – in heideggerscher Terminologie – Zuhandenheit.“  
(Bonsiepe 1996: 20)

Das Interface dient dem Nutzer, um ein mentales Modell des intransparenten Objektes in seinem Kopf zu erstellen, das den Werkzeugcharakter dieses Objektes erschließt und dadurch die verborgenen Handlungsmöglichkeiten und Informationen zugänglich macht. Das Interface Design schlägt also die Brücke zwischen der zunehmend komplexen und intransparenten Welt der Technik und der Gebrauchswelt des Menschen.

### 3. Historische Entwicklung des Mensch-Computer-Interface

Bis 1960 wurden Computer nur zur Berechnung komplizierter Algorithmen benutzt und konnten ausschließlich von Experten mit Programmierkenntnissen benutzt werden. Die eigentliche Arbeit erfolgte nach wie vor auf dem Papier. Das Problem wurde eingegrenzt und ein möglicher Lösungsweg wurde in Form eines Algorithmus aufgeschrieben. Lediglich der Schritt der Berechnung wurde durch den Computer beschleunigt, die eigentliche geistige Arbeit erfolgte aber ohne Computer. Licklieder hat 1960 darauf hingewiesen, dass diese Herangehensweise nicht die Potentiale des Computers ausschöpft. Er hat stattdessen vorgeschlagen den Lösungsweg mit Hilfe des Computers zu finden, anstatt ihn vorher festzulegen und vom Computer nur berechnen zu lassen. Seitdem wandelt sich der Umgang mit Computern mehr und mehr zu einer interaktiven Benutzung. (vgl. Preim 1999: 34)

Diese interaktive Arbeitsweise erfolgte aber bis in die 80er Jahre meist mit einer Kommandozeilenorientierten Benutzeroberfläche, bei der nur der versierte Benutzer die ihm bekannten Kommandos über die Tastatur eingeben konnte. (vgl. Abb. 1)

```
D:\>cd scanned
D:\scanned>md design
D:\scanned>dir
Datenträger in Laufwerk D: ist P2_FAT32
Volumeseriennummer: C327-3237

Verzeichnis von D:\scanned
03.05.2003 16:54 <DIR>          .
03.05.2003 16:54 <DIR>          ..
03.05.2003 16:53          794.808 macintosh.jpg
03.05.2003 16:44          425.212 schreiber.tif
03.05.2003 16:49          955.253 xerox-star.jpg
03.05.2003 17:46 <DIR>          design
                3 Datei(en)      2.175.273 Bytes
                3 Verzeichnis(se), 120.344.576 Bytes frei

D:\scanned>cd desin
Das System kann den angegebenen Pfad nicht finden.

D:\scanned>cd design
D:\scanned\design>
```

Abb. 1: Screenshot von Microsoft MS-DOS

Außerdem mussten alle Programme nacheinander ausgeführt werden. Ein Nebeneinander oder gar eine Kombination der Fähigkeiten verschiedener Programme war so nicht möglich. 1968 wurde mit NLS erstmals ein Mausbasiertes und multitaskingfähiges System von Engelbart entwickelt. Dadurch konnten verschiedene Programme parallel ausgeführt werden (Multitasking) und durch die Bedienung mit der Maus nahm die Gestaltung des Bildschirms eine ganz neue Bedeutung ein. Denn jetzt konnten Eingaben nicht mehr nur in der aktuellen Kommandozeile erfolgen, sondern an jeder Stelle des Bildschirms. Bis solche Systeme einen Massenmarkt erobert haben, sind aber noch einige Jahre vergangen.

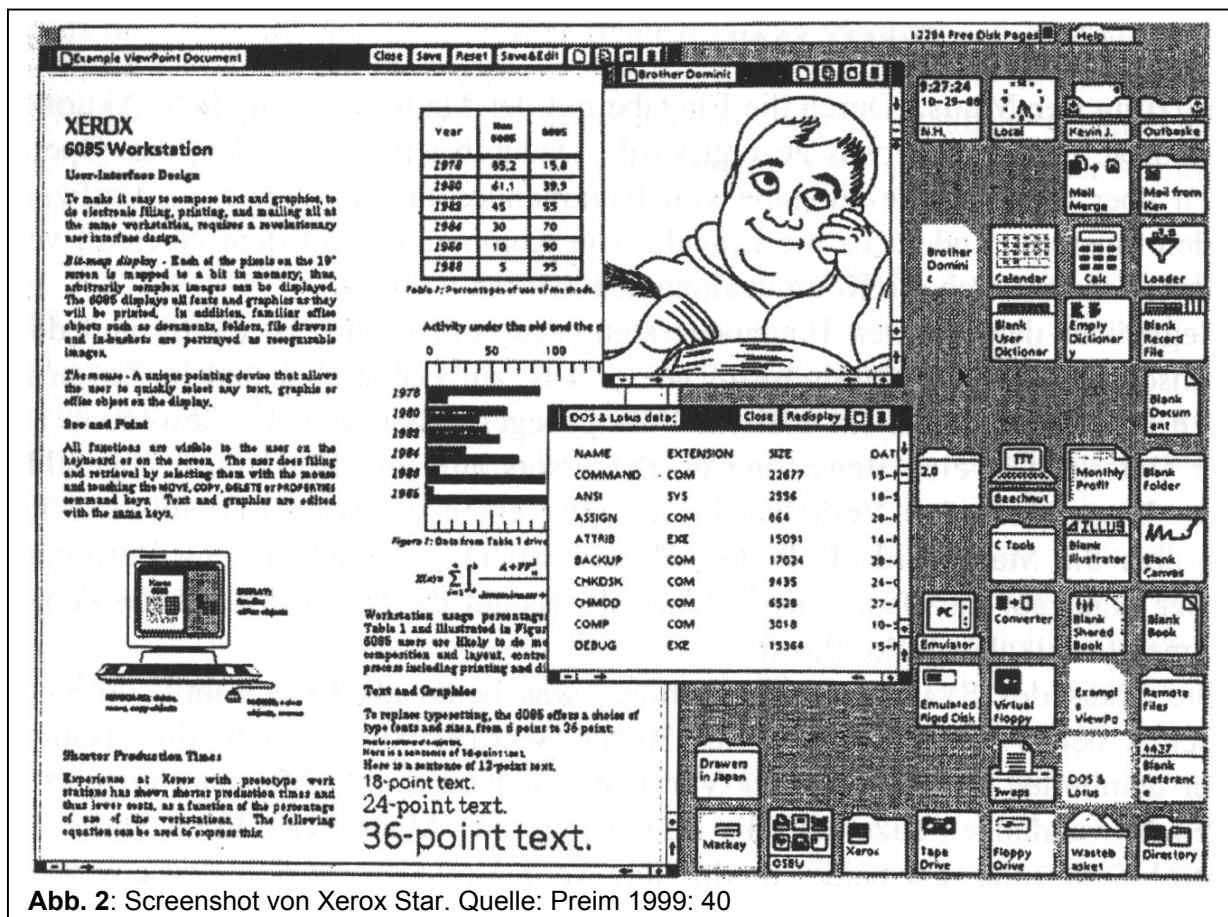


Abb. 2: Screenshot von Xerox Star. Quelle: Preim 1999: 40

### 3.1 Xerox Star (1981)

Der Xerox Star war ein erster Versuch ein Desktop-System für den breiten Markt zu entwickeln. Textverarbeitung, E-Mail, eine kleine Datenbanksoftware, rudimentäre Tabellenkalkulation und eine einfache Grafiksoftware waren hier in einer All-in-One-Workstation zusammengefasst. Der Bildschirm wurde in Anlehnung an einen Schreibtisch gestaltet. Die eine Hälfte des Bildschirms ist die eigentliche Arbeitsfläche um z.B. Texte zu erstellen. Die andere Hälfte ist die Organisationsfläche, auf der alle Dateien und Ordner sichtbar sind. Somit war der Desktop relativ starr. Überlappende Fenster wurden entgegen heutigen Programmen nur für temporäre Dialogfenster verwendet. Ursprünglich war dies generell geplant, doch in der Testphase des Xerox Star zeigte sich, dass der Benutzer unnötig von der eigentlichen Arbeit abgelenkt wird und zu viel Zeit mit der Anordnung der Fenster verbringt.

Die Organisationsseite wurde mit Symbolen für verschiedene Ordner, Dateitypen und Programme dargestellt. Dies sind die sogenannten Ikonen. Wenn der Benutzer hier etwas gemacht hat, erhielt er meist ein Feedback, das das Resultat seiner Aktion veranschaulicht hat. Beispielsweise wurden Ordner dicker, wenn sie gerade Dokumente aufgenommen hatten. Diese Schreibtisch- bzw. Desktop-Metapher wurde konsequent umgesetzt. Auf dem Desktop des Xerox Star hat man viele vertraute Dinge von einem echten Schreibtisch wiedergefunden: Ordner für den Post-Ein- und -Ausgang, einen Taschenrechner, einen Kalender oder eine Uhr. Alle Objekte auf dem Schreibtisch konnten durch Anklicken direkt aufgerufen oder bearbeitet werden. Ein Aufrufen mit Befehlssprache war nicht notwendig. Außerdem waren die verschiedenen Einzelprogramme des Xerox Star so einheitlich gestaltet, dass man nicht die Bedienung jedes einzelnen Programms von neuem erlernen musste. Die Menüs und Dialogfelder waren nach einem einheitlichen Schema gestaltet.

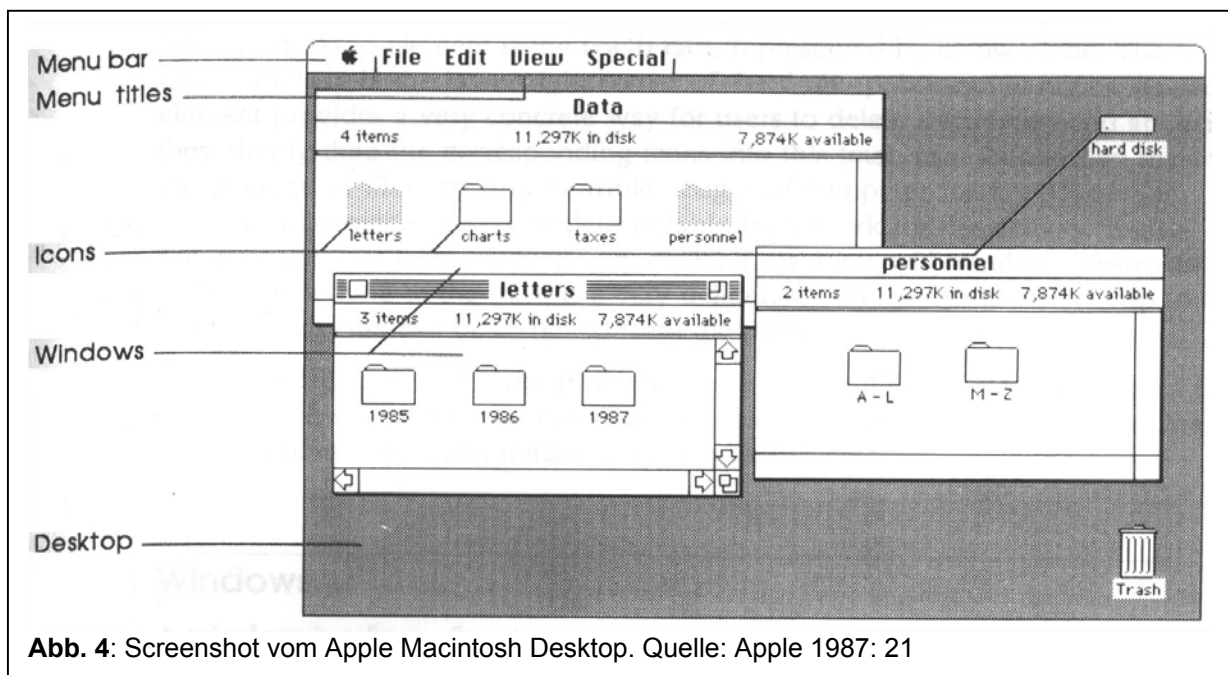
Dennoch konnte sich der Xerox Star nicht auf dem Markt durchsetzen. Die Gründe hierfür lagen aber nicht in dem richtungsweisenden Interface Design, an dem sich noch spätere Programmentwickler orientiert haben, sondern eher an anderen Problemen. So war beispielsweise das All-in-One-Konzept des Xerox Star ein Problem, wenn die Standard-Software nicht zur Lösung der individuellen Anwendungsprobleme geeignet war. Erweiterungsmöglichkeiten oder gar Software von Fremdherstellern gab es für den Xerox Star kaum. Bereits der häufige Wunsch nach einer leistungsfähigeren Tabellenkalkulation wurde von Xerox nicht erfüllt. Außerdem wurde der Xerox Star zwar als ein gutes System angesehen, dass für seine Leistung allerdings zu teuer war.



**Abb. 3:** Titelseite des Computermagazins PC-Welt von 1983. Titelthema: „Fenster und Mäuse - Rechner werden ergonomischer“. Quelle: [www.pcwelt.de//graphics/bild\\_db/16600/16670/original.idg](http://www.pcwelt.de//graphics/bild_db/16600/16670/original.idg)

### 3.2 Apple Macintosh (1984)

Der Xerox Star hatte eine nachhaltige Wirkung auf die Entwicklung neuer Computer-Systeme. Fenster und Mäuse eroberten die Computer-Welt. (vgl. Abb. 3) In Anknüpfung an den Star entstand auch der Apple Macintosh, der aber einige Neuerungen mit sich brachte. So wurde beispielsweise die direkte Manipulation der Objekte auf dem Desktop durch das sogenannte Drag-and-Drop erweitert. Bei dieser Bedienungsmöglichkeit, die von Jef Raskin entwickelt wurde, wird ein Objekt durch anklicken ausgewählt und dann mit gedrückter Maustaste an eine andere Stelle gezogen um es dort mit dem Loslassen der Maustaste zu platzieren. So konnten z.B. Dokumente gelöscht werden, indem man sie per Drag-and-Drop in den Papierkorb verschob, der ebenfalls eine der Neuerungen des Macintosh war. (vgl. Abb. 4) Wieder einmal wurde die Desktop-Metapher bemüht, um bekannte Aktionen, wie das Wegwerfen von Akten, mit einem unbekanntem Gerät, wie dem Computer, auszuführen.



Neu war beim Macintosh auch die Zwischenablage, in der Daten zwischengespeichert werden konnten, um sie unter verschiedenen Programmen auszutauschen. So gelangte beispielsweise eine Grafik in ein Textdokument, ohne die Grafik vorher separat speichern und wieder aufrufen zu müssen.

Ein wesentlicher Faktor für den Erfolg des Apple Macintosh war die Möglichkeit zur Integration von weiterer Software. Zum Apple Macintosh-Projekt unter der Leitung von Jef Raskin gehörte auch die Veröffentlichung fester Richtlinien für Entwickler in den „Human Interface Guidelines“ (Apple 1987) um so eine Integration der fremden Programme in den Apple Desktop und eine einheitliche Bedienung zu gewährleisten. Diese Richtlinien waren gleichzeitig das erste größere theoretische Werk zum Interface Design.

### **3.3 MS-Windows und X-Windows (1985)**

Die Firma Microsoft hat mit ihrem einfachen Betriebssystem MS-DOS (Abb. 1), das auf allen IBM-PCs und Kompatiblen lauffähig war, einen erheblichen Marktanteil erreicht. Eine Vielzahl von Programmen wurde für die universell verwendbaren IBM-PCs und das entsprechende Betriebssystem geschrieben. Doch MS-DOS hatte stets den Nachteil, dass nur ein Programm auf einmal ausgeführt werden konnte, während die neuen Konkurrenten Multitasking und einen benutzerfreundlicheren Desktop boten. Insbesondere Apple gewann mit dem Macintosh schnell neue Kunden und veranlasste Microsoft wenig später sein System nachzubessern. MS-Windows war seit 1985 als Zusatzprogramm erhältlich, um MS-DOS benutzerfreundlicher zu machen. Erst zehn Jahre später hat man mit Windows 95 diese beiden Produkte als eines angeboten. Mit der Version 95 hat Microsoft auch auf ein Problem reagiert, dass die Entwickler des Xerox Star bereits in der Testphase erkannt haben: den Arbeitsaufwand durch die Anordnung der Fenster. Während bei Vorgänger-Versionen von Windows der Desktop oftmals erst aufgeräumt werden musste um an ein Programm unter den Fenstern heranzukommen, gibt es seit Windows 95 eine sogenannte Taskleiste (Abb. 5 unten), die mit ihren ausfahrbaren Menüs permanent verfügbar ist. Ansonsten war MS-Windows weitgehend an den Apple Desktop angelehnt. Mit der Zeit wurde es zwar in verschiedenen Versionen farbiger, multimedialer, verspielter und individueller konfigurierbar – in Bezug auf das Interface Design gab es allerdings wenig Neuerungen. Dennoch wurde Microsoft mit Windows zum Marktführer bei den Betriebssystemen.

Während MS-Windows den breiten Markt bediente, war X-Windows für spezialisiertere Anwender konzipiert, die mit dem Betriebssystem Unix arbeiten. Die

Besonderheit des Interfaces von X-Windows liegt darin, dass Informationen gleich auf mehreren Monitoren getrennt voneinander angezeigt werden können. Dies ist zum Beispiel für Leitstände und Kontrollräume nützlich. Zusammen haben X-Windows und MS Windows dafür gesorgt, dass die Desktop-Metapher sich auf dem Markt so durchgesetzt hat, dass sie heute das Standard Interface Design ist.

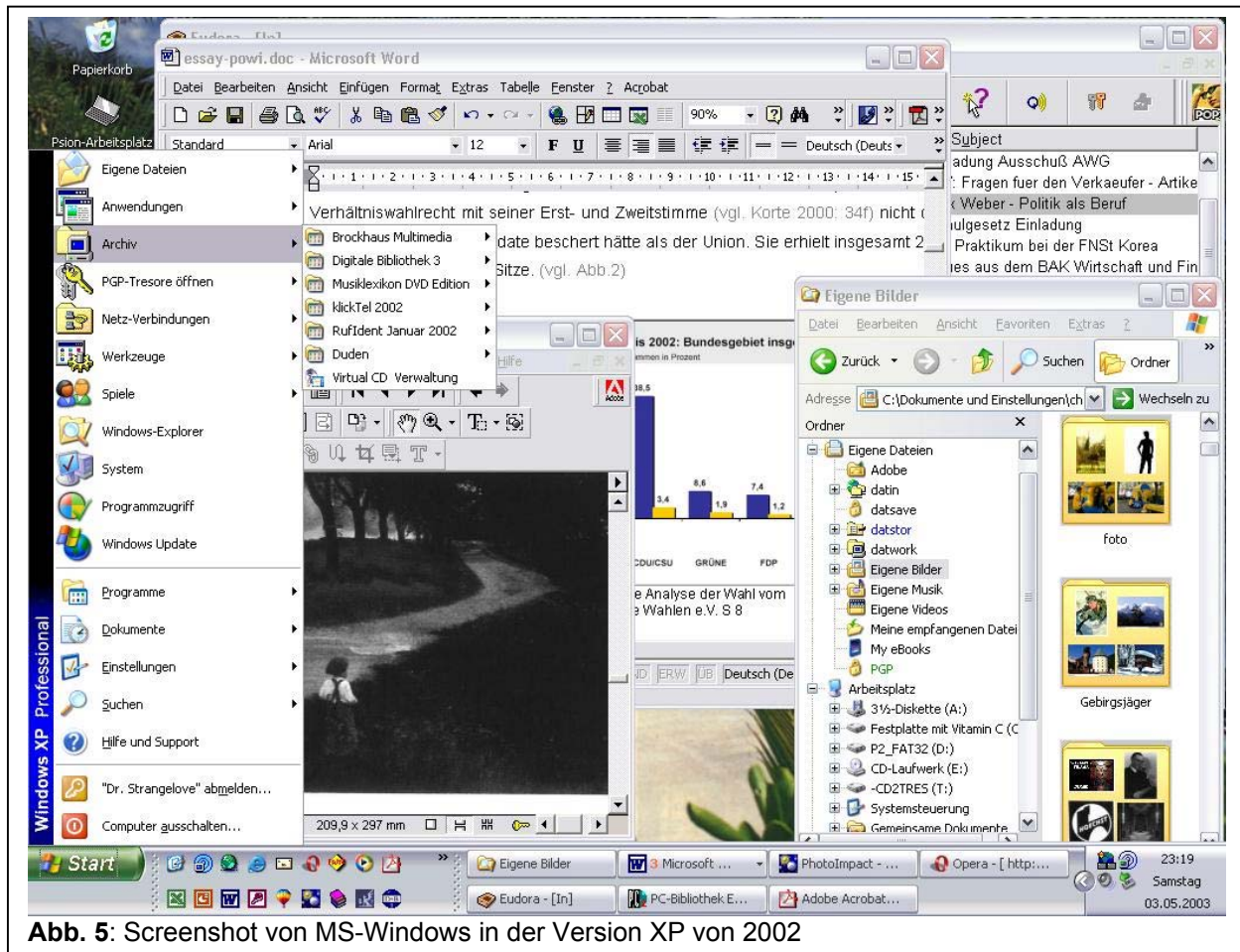


Abb. 5: Screenshot von MS-Windows in der Version XP von 2002

### 3.4 Personalisierte Agenten (1989)

Es gibt oftmals den Wunsch, den Computer nicht als ein Werkzeug, sondern als einen Assistenten zu begreifen, dem man ganz natürlich eine Frage stellen kann und darauf eine Antwort bekommt. Dieser Rückgriff auf die bekannte Mensch-zu-Mensch-Kommunikation erfolgt z.B. beim Hilfsprogramm „Karl Klammer“ von MS-Office. (vgl. Abb. 6) Der Benutzer kann zwar eine Frage in natürlicher Sprache eingeben, doch das Missverständnis ist vorprogrammiert. Denn das Hilfsprogramm versteht weder Syntax noch Inhalt der Frage, sondern sucht nur nach Einträgen aus einer Hilfedatenbank, die möglichst viele der eingegeben Schlagwörter enthält. Manchmal

kann ein Agent zwar auf diese Weise einen Treffer landen, doch die Hoffnung auf einen echten Assistenten wird hier nur geweckt und nicht erfüllt.

Erstmals wurde die Idee eines Computer-Agenten 1989 populär, als die Firma Apple eine Werbevideo mit dem Namen „The

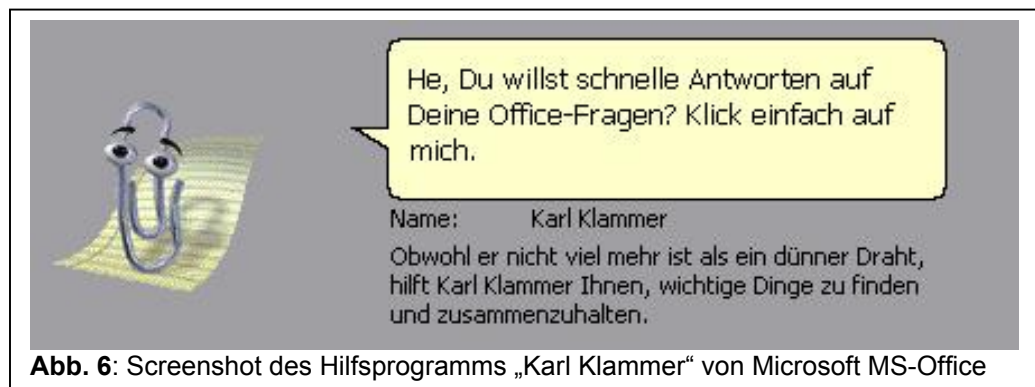


Abb. 6: Screenshot des Hilfsprogramms „Karl Klammer“ von Microsoft MS-Office

Knowledge Navigator“ vorstellt. Darin war ein Schauspieler als Buttler verkleidet in der rechten oberen Ecke eines Laptop-Bildschirms zu sehen, der dem Benutzer assistierte. Der Benutzer war ein Professor, der nach Informationen über den Regenwald sucht, bzw. seinen virtuellen Kammerdiener danach suchen ließ. Doch in der Realität scheiterten alle Bemühungen einen Computer-Agenten zu schaffen, mit dem echte Mensch-zu-Mensch-Kommunikation möglich ist.

### 3.5 Das World Wide Web (1992)

Seitdem es 1992 im Internet den benutzerfreundlichen Dienst World Wide Web (WWW) gibt, werden immer mehr Informationen digitalisiert zur Verfügung gestellt, tausendfach vervielfältigt und in aller Welt abgerufen. Doch man kann nicht nur Informationen aus aller Welt abrufen, man kann sie auch zur Verfügung stellen. Diese Allgegenwärtigkeit der Information hat auch die Distanz zwischen den Informationen austauschenden Menschen reduziert. Man spricht davon, dass uns das Internet zu einer Art „global village“ verbindet, da der Austausch von Informationen in einem Internet-Forum jetzt noch unkomplizierter ist als früher das Gespräch am Dorfbrunnen. Mit der Verfügbarkeit all dieser

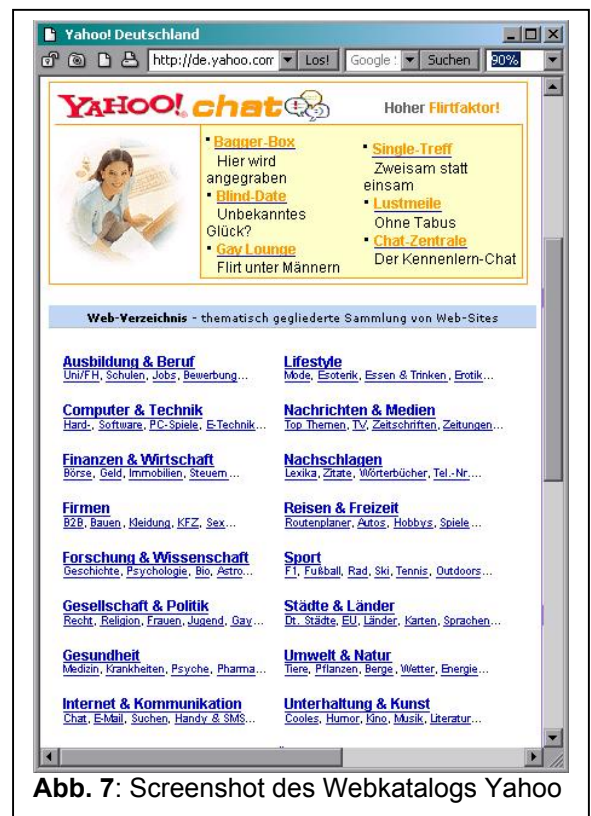


Abb. 7: Screenshot des Webkatalogs Yahoo

Informationen stellt sich aber auch die Frage danach, wie man die relevanten Informationen herausfiltert. Man benötigt eigene Programme - sogenannte Browser - um durch den virtuellen Informationsraum – auch Cyberspace genannt – zu navigieren. Das Interface Design gewinnt durch die Fülle dieser Informationen an Bedeutung.

### **3.6 Sprach- und Handschrifterkennung (1996)**

Der Wunsch mit dem Computer ganz „natürlich“ zu kommunizieren ist zwar deutlich älter, doch erst 1996 konnte auf zwei wichtigen Gebieten in diesem Bereich ein Durchbruch zur Marktreife erzielt werden. Bei der Spracherkennung spricht der Benutzer in ein Mikrofon und der Computer zeichnet nicht die Sprache als Geräusch auf, sondern erkennt sie und wandelt sie in Text um. Seit 1996 sind diese Systeme zwar so ausgereift, dass man sie überhaupt dem Endanwender zumuten kann, dennoch arbeiten diese Systeme noch fehlerhaft und verlangen vom Benutzer eine sehr deutliche, langsame und akzentfreie Sprechweise. Auch wenn Spracherkennung inzwischen zum Standardlieferumfang von Windows XP gehört, wird sie vergleichsweise selten genutzt und hat das Interface Design daher nur gering beeinflusst. Selbst bei originär sprachbasierten Systemen wie dem Telefon, ist die Bedienung eines Sprachmenüs (z.B. Anrufbeantworter eines Handys) deutlich einfacher, wenn man die Tasten benutzt, statt die Menüoptionen auszusprechen.

Etwas erfolgreicher war die Handschrifterkennung, bei der der Benutzer seine Texteingaben nicht mehr über die Tastatur macht, sondern mit einem Stift auf einem berührungsempfindlichen Bildschirm (Touchscreen) wie auf einem Blatt Papier schreibt. Bei stationären Computern ist diese Eingabemethode zwar nach wie vor selten, da Touchscreens teuer sind und die Eingabe mit einer großen ergonomischen Tastatur meist schneller ist als in Schreibschrift. Doch bei mobilen Kleinstcomputern – sogenannten Palmtops oder PDAs (Personal Digital Assistant) – hat die Handschrifterkennung die Tastatureingabe fast vollständig verdrängt. Kleinstcomputer mit Tastatur sind heute kaum noch auf dem Markt erhältlich.

### 3.7 Wireless Application Protocol (2000)

Nachdem das World Wide Web sich als neues Medium etabliert hatte und ein Internetzugang auf der Mehrzahl der stationären Computer verfügbar war, wollte man die Informationen dieses weltumspannenden Netzes auch auf mobilen Kleinstcomputern und Handys für Unterwegs verfügbar machen. Allerdings ist das Interface Design der meisten Webseiten auf eine Mindestbildschirmgröße ausgerichtet, die Kleinstcomputer und Handys nicht bieten können. Daher wurde das Wireless Application Protocol (WAP) entwickelt, das sich besonders an diesen Bedürfnissen orientiert.

Zur Zeit des Marktstarts im Jahr 2000 hatte WAP noch hauptsächlich technische Probleme in den Mobilfunknetzen, da diese geringe Übertragungskapazitäten und lange Wartezeiten mit sich brachten. Mit der Weiterentwicklung der Mobilfunknetze wurde aber auch WAP beliebter und in Japan wird das vergleichbare System i-mode bereits von einem Viertel der Handybesitzer genutzt. Der mobile Zugriff auf das Internet bringt viele Vorteile mit sich. So können beispielsweise Einrichtungen in der unmittelbaren Nähe des Nutzer gesucht werden, Angebote von einem Geschäft in dem man gerade steht können mit der Konkurrenz verglichen werden und man kann Wartezeiten z.B. in Verkehrsmitteln sinnvoll nutzen. Das WAP bringt aber auch neue Interface Design-Anforderungen mit sich. Da nur eine geringe Bildschirmgröße und wenig Eingabemöglichkeiten bestehen, muss sich der Designer auf das Wesentliche beschränken. (vgl. Abb. 8)



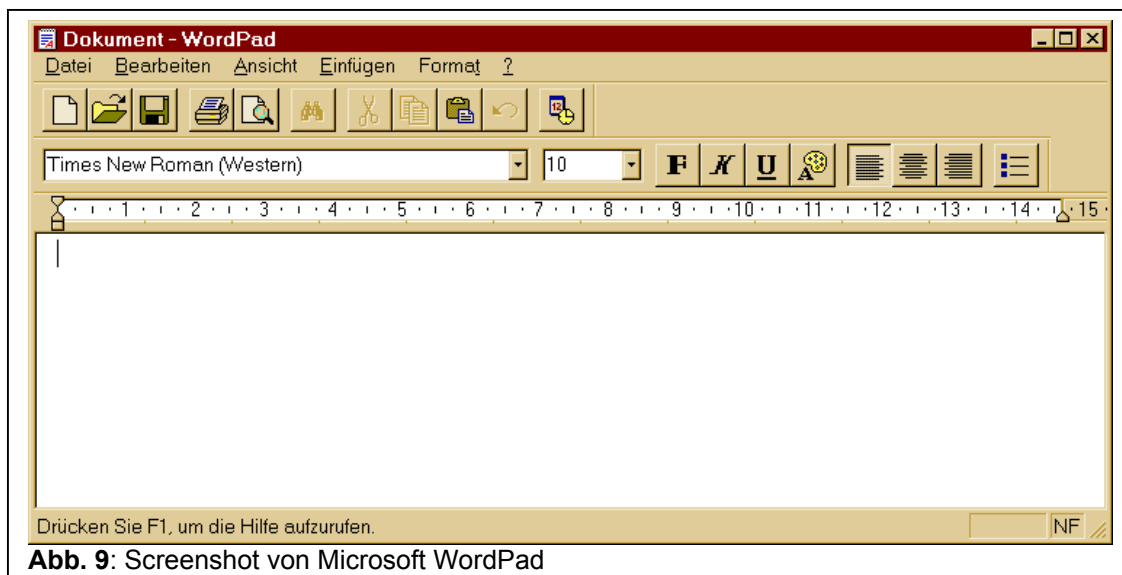
**Abb. 8:** Suche nach einem nahegelegenen Restaurant mit einem WAP-Handy.  
Quelle: [www-106.ibm.com/developerworks/ibm/library/i-lbs/](http://www-106.ibm.com/developerworks/ibm/library/i-lbs/)

## 4. Prinzipien des Interface Design

Es gibt kein Werk, das man als „die Bibel“ des Interface Design ansehen könnte. Daher gibt es auch kein Regelwerk, das in sich konsistent ist. Es sollen aber dennoch an dieser Stelle einige Regeln dargestellt werden, die sich in der Theorie des Interface Design herausgebildet haben. Gleichzeitig ist es auch nötig auf Grenzen dieser Regeln hinzuweisen oder Zielkonflikte zwischen den Regeln aufzuzeigen.

### 4.1 Transparenz

Ursprünglich war das Interface als eine Hilfe für den Benutzer gedacht, ein mentales Modell des Computerprogramms in seinem Kopf zu konstruieren, das die Kenntnis des Programmierers repliziert. Das Interface darf dabei keine Aufmerksamkeit erfordern sondern muss für den Benutzer quasi durchsichtig sein und den direkten Blick auf die Funktionen des Programms freigeben. Dazu müssen Aufbau, Möglichkeiten und Grenzen des technischen Handlungsspielraums stets erkennbar sein. Bei sehr einfachen Programmen, wie beispielsweise Microsoft WordPad (Abb. 9), ist das noch leicht möglich. Der Zweck des Textverarbeitungsprogramms kann sofort mit dem Eintippen von Text erreicht werden. Gleichzeitig sind alle möglichen Optionen wie beispielsweise Drucken, Speichern, Schriftart, Fettdruck oder zentrierte Ausrichtung über grafisch illustrierte Schaltflächen an der oberen Leiste verfügbar. Selbst Optionen die augenblicklich nicht verfügbar sind, werden als solche angezeigt, indem sie nicht farbig dargestellt werden.



**Abb. 9:** Screenshot von Microsoft WordPad

## 4.2 Beschränkung auf das Wesentliche

Mit zunehmender Komplexität der Programme wird es auch schwieriger, alle Funktionen eines Programms auf einem Bildschirm anzuzeigen. In Abbildung 10 ist zu sehen, wohin es führt wenn man MS-Word so einstellt, dass es nahezu alle Funktionen mit einer eignen Schaltfläche darstellt. Das Interface nimmt den ganzen Bildschirm ein und das eigentliche Arbeitsobjekt Text ist darunter kaum noch zu erkennen.

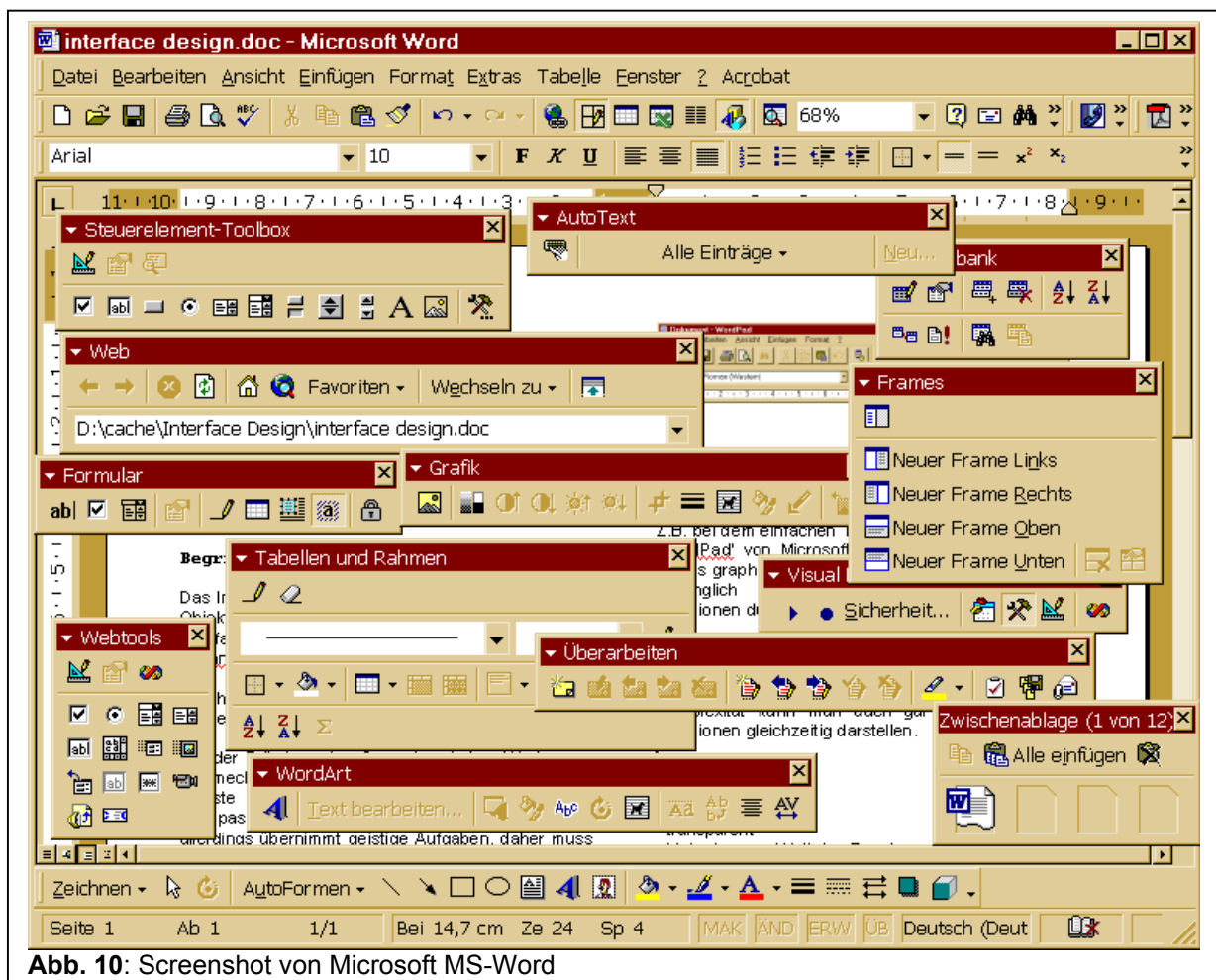


Abb. 10: Screenshot von Microsoft MS-Word

Daher ist es nötig, sich auf das Wesentliche zu beschränken. Doch das ist leichter gesagt als getan, denn man kann als Designer schlecht wissen, welche Option für den Nutzer augenblicklich wesentlich ist. Eine wesentliche Vereinfachung des Interfaces ist es, wenn man nur die Optionen anzeigt, die auf das ausgewählte Objekt angewendet werden können. Die Sprache des Interface ist also einfacher, wenn sie nach dem Objekt-Aktion-Schema erfolgt, anstatt im Aktion-Objekt-Schema zuerst eine der möglichen Aktionen auszuwählen um dann anzuzeigen, an welchem Objekt diese Aktion ausgeführt werden soll. In MS-Word beispielsweise werden die


Aktionen zum bearbeiten einer Tabelle, wie Zelle löschen oder Zeile hinzufügen, nur dann angezeigt, wenn sich der Cursor auch in einer Tabelle befindet. Eine weitere Möglichkeit das Interface übersichtlicher zu gestalten besteht darin, nur häufig benutzte Aktionen mit einem grafischen Icon direkt verfügbar zu machen. Seltener verwendete Aktionen kann man hingegen zu Gruppen zusammenfassen und in Menüs verfügbar machen.

### **4.3 Redundanz**

Da Benutzer nicht alle gleich sind, ist es ratsam verschiedene Wege anzubieten um ein Problem zu lösen. Experten ziehen es in der Regel vor Aktionen mit einer kurzen schnellen Befehlssprache durchzuführen. Beispielsweise kann ein Objekt in der Windows-Umgebung mit der Tastenkombination [Strg] + [C] in die Zwischenablage kopiert werden. Gelegentliche Nutzer ziehen hingegen eine Schaltfläche vor, die mit einer Grafik als Gedächtnisstütze versehen ist. Wohingegen seltene Nutzer lieber ein Menü benutzen, dass durch seine Beschriftung die Aktionen eindeutig beschreibt. (vgl. Shneidermann 2002: 90f, Preim 1999: 100)

Raskin (2001: 89) vertritt die absolut gegenteilige Auffassung. Seiner Meinung nach wird das Interface unnötig verkompliziert, wenn jede Aktion mit mehreren Wegen erreicht werden kann. Sobald der Benutzer erkennt, dass es mehrere Wege gibt, macht er sich zu viele Gedanken darüber, welcher Weg für ihn der beste ist. Dadurch wird er von seiner eigentlichen Arbeit abgelenkt und das Interface erhält mehr Aufmerksamkeit, als ihm aufgrund des Gebotes der Transparenz eigentlich gebührt.

### **4.4 Undo- / Storno-Funktion**

Damit der Benutzer unbeschwert die verschiedenen Funktionen eines Programms erkunden kann, müssen Fehler stets revidierbar sein. Dazu muss jede erfolgte Aktion auch wieder rückgängig gemacht werden können. Bei den meisten Programmen geht dies mit dem Menü-Befehl „Bearbeiten“-„Rückgängig“, dem Tastenkürzel [Strg]-[Z] oder dem Icon  .

#### **4.5 Systemzustände sichtbar machen**

Ein so universelles System wie ein Computer hat sehr viele verschiedene Zustände, von denen die Wirkung einer Eingabe abhängt. Selbst ein einzelnes Programm kann noch einmal in verschiedene Modi unterteilt werden, z.B. weil sonst das Interface wie in Abb. 10 zu viel Platz wegnehmen würde. Diese unterschiedlichen Modi sind aber auch mit Gefahren verbunden. Denn wenn ein Benutzer einen anderen Modus erwartet als tatsächlich vorliegt, erwartet er von seinen Eingaben ein spezielles Ergebnis und ist überrascht, dass dieses Ergebnis nicht eintritt. Raskin (2001: 57f) geht sogar soweit Modi gänzlich abzulehnen. Einzig sogenannte Quasi-Modi würde er akzeptieren, da sich in Untersuchungen gezeigt hat, dass sich mit solchen Quasi-Modi Bedienfehler aufgrund von Modi nahezu ausschließen lassen. Ein Quasi-Modi ist ein Systemzustand, in dem sich das System nur solange befindet, wie der Benutzer eine Taste gedrückt hält. Ein Beispiel wäre die Umschalt-Taste, mit der man den Quasi-Modus „Großbuchstaben“ an der Tastatur einstellt. Die Feststelltaste wäre hingegen ein nach Raskin abzulehnender Modus.

Weitere Modi, die Preim (1999: 62) und Shneiderman (2002: 30) befürworten, während sie Raskin (2001: 91) strikt ablehnt, sind Modi, die das Interface an die Bedürfnisse von Anfängern und Experten anpassen oder sonstige individuellen Konfigurationen zulassen. Während die Befürworter den Vorteil in einem System sehen, dass optimal auf die individuellen Bedürfnisse des Nutzers abgestimmt ist, hat Raskin gleich mehrere Gegenargumente. Zum einen ist der Benutzer von seiner eigentlichen Arbeit abgelenkt und widmet zuviel Zeit der Konfiguration des Interface. Zum zweiten kann sich der Benutzer nur schwer an ein Interface gewöhnen, wenn sich dieses Interface ständig ändert. Und zum dritten verstößt diese individuelle Konfiguration gegen das Gebot der Einheitlichkeit.

#### **4.6 Einheitlichkeit**

Damit der Benutzer nicht jedes Mal Zeit auf die Interpretation des Interface verwenden muss, sollten gleiche Funktionen von gleich aussehenden Schaltflächen oder gleich lautenden Menübefehlen realisiert werden. Innerhalb der Windows-Oberfläche gibt es z.B. Mindeststandards, wie das Schließen eines Fensters mit der Schaltfläche „x“ rechts oben. (vgl. Abb. 5, Abb. 7, Abb. 9, Abb. 10)

## 4.7 Intuitive Bedienung durch Metaphern

In Zusammenhang mit Interface Design wird oftmals die Forderung gestellt, auf Handbücher gänzlich zu verzichten und stattdessen Programme intuitiv bedienbar zu gestalten. Als intuitiv werden Interfaces in der Regel erachtet, wenn sie bereits aus anderen Programmen oder nicht computerisiertem Arbeiten bekannt sind. Ein Beispiel für eine solche Metapher aus dem nicht computerisierten Bereich ist die Desktop-Metapher, die seit dem Xerox Star-System verwendet wird. (vgl. 3.1) Ein weiteres Beispiel wäre die Abbildung eines klassischen Ringbuch-Terminplaners aus Papier in einem elektronischen Terminplaner. (vgl. Abb. 11)

Dabei muss allerdings beachtet werden, dass der Benutzer evtl. mehr mit der

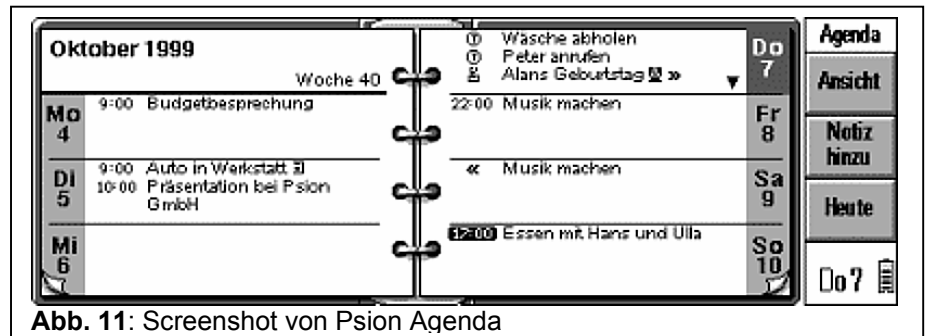


Abb. 11: Screenshot von Psion Agenda

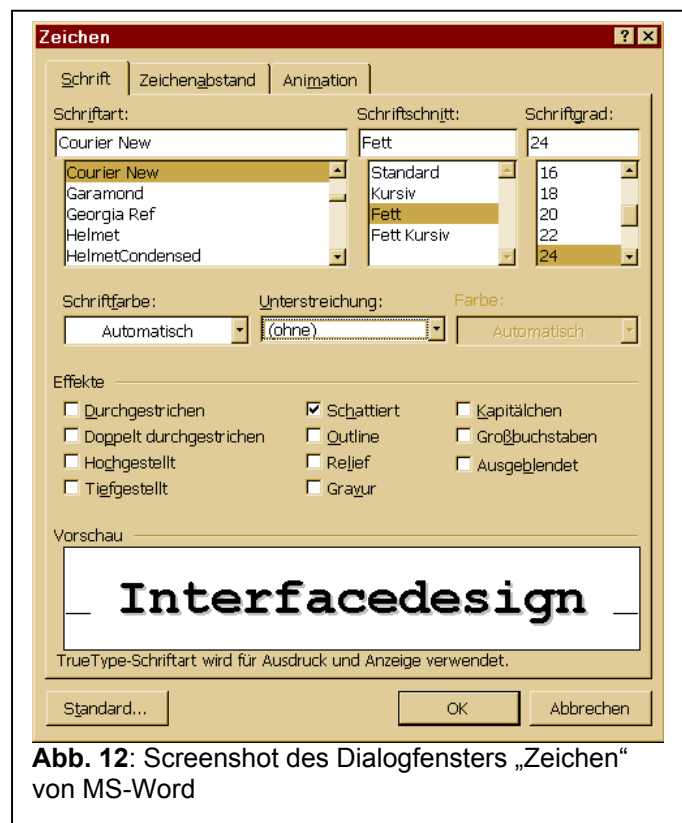
Metapher verbindet, als das eine beabsichtige und nützlich beschriebene Handlungsziel. Metaphern können stets Fehlschlüsse implizieren. Außerdem besteht die Gefahr, dass das wahre Potential an Funktionen und Handlungsmöglichkeiten im Verborgenen bleibt. Der Benutzer hat dann nur die Funktionen eines Papier-Terminplaners, kann aber die Vorteile eines elektronischen Planers nicht nutzen, da sie mit der Metapher nicht abgebildet werden können. So ist beispielsweise in Abbildung 11 nicht innerhalb der Metapher ersichtlich, wie von der Wochenansicht zur Tages- oder Monatsansicht der Termine gewechselt werden kann. Dazu ist die „Ansicht“-Schaltfläche außerhalb der Metapher vorgesehen.

Zudem sind Metaphern wie auch weitere intuitive Bedienerwartungen stets kulturell bedingt. So entspricht beispielsweise die Suchrichtung auf dem Bildschirm der Leserichtung in der jeweiligen Landessprache. (vgl. Shneiderman 2002: 38) Daher ist eine gewisse multikulturelle Interfacevielfalt nötig, wenn es einem nicht gelingt - dem Gebot der Einheitlichkeit folgend - ein interkulturelles Interface zu gestalten, was zudem in der global vernetzten Computerwelt ratsam ist.

Eine große Gefahr sieht Raskin (2001: 177) zudem in dem Bestreben ein intuitiv bedienbares Interface zu gestalten, wenn dadurch nur die Interfaces von weitgehend bekannten Programmen übernommen werden. Dadurch wird nur der Fortschritt im Interface Design behindert. Im Sinne des Fortschritts ist es ratsam, bestehende Interface Design-Paradigmen in Frage zu stellen anstatt sie als intuitiv zu betrachten. So ist selbst die heute so selbstverständliche Bedienung eines Computers mit einer Maus nicht intuitiv, wenn man es nicht erlernt hat. Eine Versuchsperson die zwar einen Computer kennt, aber noch nie mit einer Maus gearbeitet hat, würde bei der Anweisung „klicken Sie mit der Maus auf das x in der rechten oberen Fensterecke“ vermutlich die Maus hochheben und damit auf den Bildschirm klopfen.

#### 4.8 Entlastung des Gedächtnisses

Die Orientierung in einer Vielzahl von Handlungsmöglichkeiten und Informationen erfordert nicht nur eine klare Beschreibung und Strukturierung der selbigen, sondern auch eine Darstellung, die das Gedächtnis des Benutzers entlastet. Durch graphische Übersichten oder Ablaufmodelle muss dem Benutzer immer angezeigt werden, wo er sich im Programm befindet und welche Wege ihm offen stehen. In Abbildung 12 ist das Fenster zu Wahl des Zeichenformats von MS-Word zu



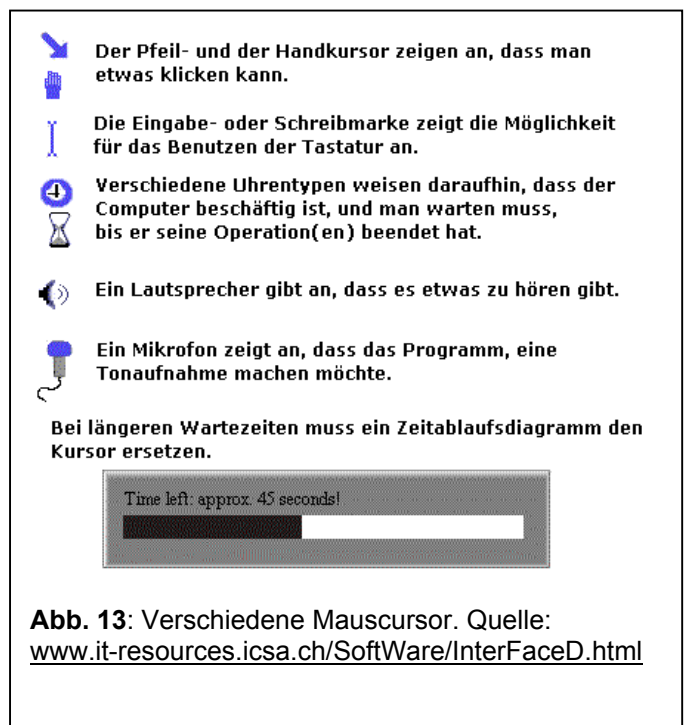
**Abb. 12:** Screenshot des Dialogfensters „Zeichen“ von MS-Word

sehen. Hier wird durch das Karteikartensystem die Orientierung erleichtert und das Vorschaufenster erinnert an das Objekt, das mit dem Dialogfenster augenblicklich bearbeitet wird. Im Bereich „Schriftschnitt“ wurde hier das Gebot der Beschränkung auf das Wesentliche nicht beachtet. Obwohl es nur zwei Optionen „Fett“ und „Kursiv“ gibt, werden hier vier Optionen angezeigt. „Standard“ oder „Fett Kursiv“ hätte man auch mit einer Nichtwahl oder der Kombination der beiden Optionen realisieren können, wie es auch im Bereich „Effekte“ erfolgt.

## 4.9 Unmittelbares Feedback

Shneiderman (2002: 428f) hat ausführlich untersucht, wie Benutzer auf Latenzzeiten bei Computerprogrammen reagieren und festgestellt, dass die Mehrzahl der Benutzer sehr ungeduldig ist. Allerdings nicht weil sie einen besonders schnellen Computer erwarten, sondern eher weil sie fürchten, dass ihre Eingabe nicht richtig verstanden wurde oder dass der Computer nicht die gewünschte Aktion ausführt. Daher muss stets durch ein unmittelbares Feedback angezeigt werden, dass die Eingabe

richtig verstanden wurde. Bei technisch bedingten längeren Wartezeiten muss die geschätzte Wartezeit und nach Möglichkeit auch der Grund benannt werden, wie beispielsweise die Menge der durchsuchten Datensätze. Das Feedback sollte dabei stets im Zentrum der Aufmerksamkeit erfolgen, das sich in der Regel an der Stelle des Cursors befindet. Daher gibt es auch eine Reihe von Cursor-Formen, die unterschiedliches Feedback signalisieren. (vgl. Abb. 13)



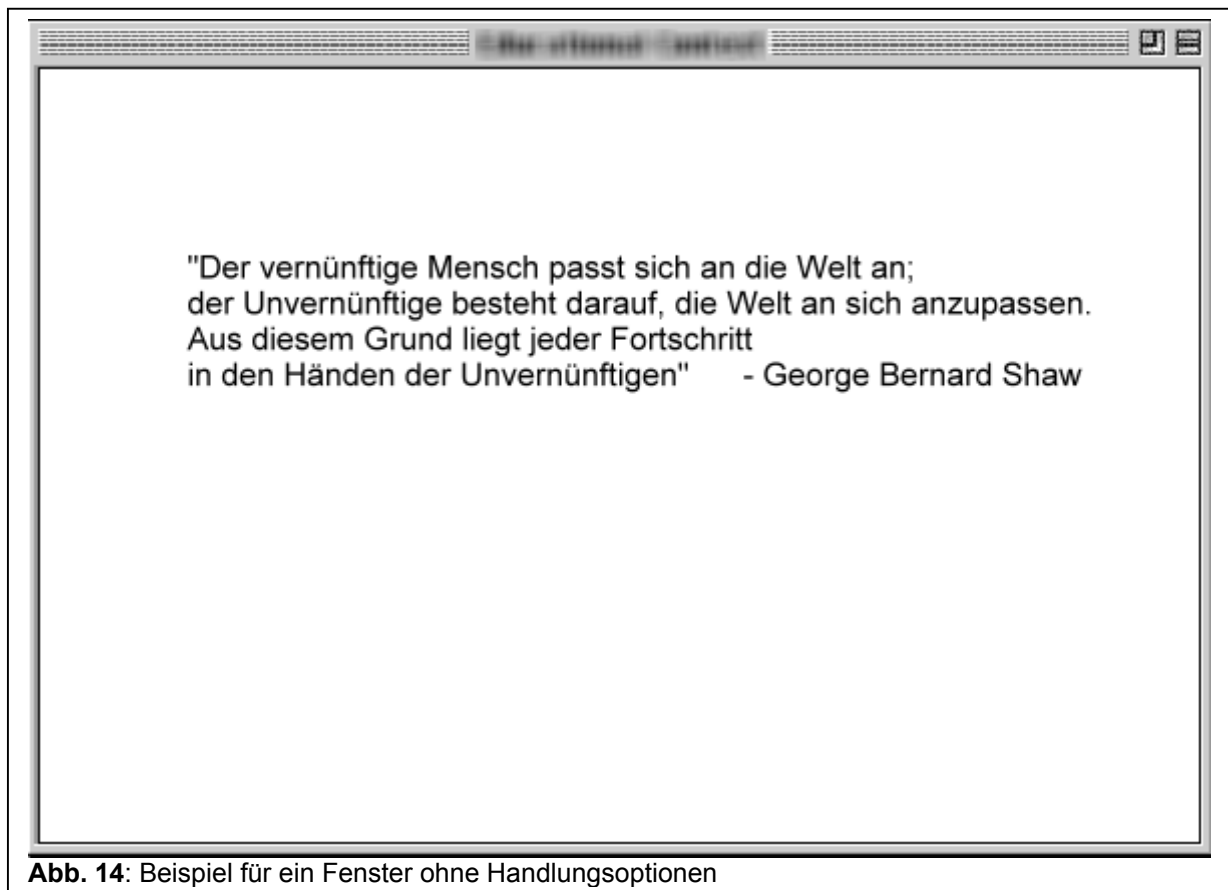
**Abb. 13:** Verschiedene Mauscursor. Quelle: [www.it-resources.icsa.ch/SoftWare/InterFaceD.html](http://www.it-resources.icsa.ch/SoftWare/InterFaceD.html)

## 4.10 Fehler erkennen und erklären

Neben dem selbstverständlichen Bestreben Fehler des Computers zu vermeiden, sollte man sich bei der Programmierung aber auch mit möglichen Fehlern des Benutzers befassen und eine erklärende Reaktion des Computers vorsehen. Beispielsweise sind bei Datumseingaben im Feld „Monat“ in der Regel nur Eingaben von 1 bis 12 zulässig. Anstatt den Benutzer alle Eingaben machen zu lassen und dann mit einem kryptischen „illegal data“ zu konfrontieren oder gar eine fehlerhafte Eingabe ohne Kommentar zu unterbinden, sollte man ihn besser unmittelbar bei der Eingabe eine freundliche Erläuterung geben, dass im Feld „Monat“ nur Eingaben von 1-12 zulässig sind.

#### 4.11 Kontrolle muss stets beim Benutzer liegen

Der Computer muss stets dem Menschen dienen und nicht umgekehrt. Daher darf das Programm vom Benutzer keine unnötigen Eingaben verlangen, angefangene Prozeduren müssen stets abgebrochen werden können und während eines laufenden Programms müssen andere Programme verfügbar sein. Kein Dialogfenster darf weniger als zwei Schaltflächen enthalten, die beispielsweise „Ja“ und „Nein“ oder „Weitermachen“ und „Abbrechen“ lauten können. Fenster die nur eine Information anzeigen und den Benutzer dazu zwingen diese Information mit einem Klick auf „OK“ hinzunehmen sind unbedingt zu vermeiden. Noch schlimmer sind Fenster wie in Abbildung 14 ohne jegliche Handlungsoption, die einem eine didaktische Präsentation aufdrängen.



**Abb. 14:** Beispiel für ein Fenster ohne Handlungsoptionen

## 5. Bewertung

Das Interface Design ist eine Design Theorie, die sich zwar erst mit der Entstehung und Verbreitung von Computern entwickelt hat, die aber in der Tradition funktionalistisch begründeter Design Theorien steht. Die Funktion die das Interface Design begründet, ist das bestmögliche Verfügbarmachen von Informationen und/ oder Handlungsmöglichkeiten. Im Falle der Handlungsmöglichkeiten zielt das Interface Design weniger auf die ergonomische Anpassung an die Physis des Menschen, sondern eher auf die Anpassung der computerisierten, nicht mechanischen Schnittstellen an die Psyche des Menschen. Dazu bedient sich das Interface Design Erkenntnissen aus der Wahrnehmungspsychologie.

Da neue Mensch-Maschine-Schnittstellen immer seltener mechanisch und immer öfter computerisiert sind, nimmt auch die Bedeutung des Interface Design stetig zu. Prinzipiell ist die Idee des Interface Design aber auch auf nicht computerisierte Informationen und Handlungsmöglichkeiten anwendbar. Es geht dabei nicht nur um Fragen der Gebrauchseffizienz, sondern oftmals um die Aufgabe eine Information oder eine Handlungsmöglichkeit durch ihr Interface überhaupt nutzbar zu machen und dadurch für den Menschen erst zu konstituieren.

Daher muß der Aspekt des Interface Design möglichst früh im Entwicklungsprozess neuer Produkte oder Informationen berücksichtigt werden. Im Falle von Informationen ist es bereits Aufgabe des Designers als Co-Autor die Information zu strukturieren, wenn er nicht bloß das Erscheinungsbild modisch aufpolieren soll. Im Falle von Produkten mit Handlungsmöglichkeiten muss der Interface Designer die Arbeitsprozesse beobachten, analysieren und interpretieren. Er darf dabei nicht zu oberflächlich bleiben, da sonst die Gefahr besteht, dass der Arbeitsprozess lediglich digitalisiert wird, ohne weitere Vorteile des Computers nutzbar zu machen.

Bei der Vielfalt von verfügbaren Informationen und Handlungsmöglichkeiten an Computern kann das Interface Design nicht alles auf einmal darstellen, sondern muss strukturieren, hierarchisieren und sich auf das Wesentliche beschränken. Für zusätzliche Verzierungen ist da kein Platz mehr. Diese funktionalistische Begründung

des Interface Design bietet wenig Raum künstlerischer Kreativität oder narzisstischen Neigungen nachzugehen, da sich das Design nicht in den Vordergrund drängen sollte, sondern lediglich Information oder Handlungsmöglichkeiten vermitteln muss.

Betrachtet man allerdings Werke aus der ureigenen Domäne des Interface Design, dem Screen Design, dann wird man oftmals feststellen, dass sich die Prinzipien des Interface Design in der Praxis keineswegs durchgesetzt haben. Nicht nur wegen mangelnder Reflexion, sondern auch wegen der Verfolgung anderer ästhetischer Vorstellungen, mit denen man hofft er den Geschmack des Kunden zu treffen.

Trotz seiner inzwischen bald zwanzigjährigen Geschichte, ist das Interface Design noch sehr jung, und seine Theorie nicht sehr gefestigt. Die Frage ob das Interface Design die eine oder auch nur die neue Design Theorie des Computerzeitalters sein wird, kann derzeit noch nicht abschließend beantwortet werden.

## 6. Literaturverzeichnis

- Apple (1987):** Human Interface Guidelines: The Apple Desktop Interface. Reading etc.: Addison-Wesley
- Bonsiepe, Gui (1996):** Interface Design neu begreifen. Mannheim: Bollmann
- Johnson, Steven (1999):** Interface Culture – Wie neue Technologien Kreativität und Kommunikation verändern. Stuttgart: Klett-Cotta
- Laurel, Brenda (Hrsg.) (1990):** The Art of Human-Computer Interface Design. Reading etc.: Addison-Wesley
- Preim, Bernhard (1999):** Entwicklung Interaktiver Systeme. Berlin etc.: Springer
- Raskin, Jef (2001):** Das intelligente Interface – Neue Ansätze für die Entwicklung interaktiver Benutzerschnittstellen. München: Addison-Wesley
- Rifkin, Jeremy (2001):** Das Ende der Arbeit und ihre Zukunft. Frankfurt/Main: Fischer
- Röllinghoff, Andreas (1999):** 15 Regeln für das Interface Design, [www.it-resources.icsa.ch/SoftWare/InterFaceD.html](http://www.it-resources.icsa.ch/SoftWare/InterFaceD.html)
- Shneiderman, Ben (2002):** User Interface Design. Bonn: mitp
- Turing, Alan (1994):** Kann eine Maschine denken? In: Zimmerli, Walther / Wolf, Stefan (1994): Künstliche Intelligenz – Philosophische Probleme. Stuttgart: Reclam S 39-78
- Waldmann, Dieter (1971):** Dreht euch nicht um, der Golem geht um – Das Zeitalter der Muße. Stuttgart: Südwestfunk. Regie: Peter Beauvais
- Wolski, David (2003):** Kult 20: Die Entwicklung der Betriebssysteme <http://www.pcwelt.de/ratgeber/extras/30324/>