

10 Interaktionsformen

Lernziele

Sie lernen die verschiedenen Interaktionsformen kennen, deren Einsatzgebiete sowie die Vor- und Nachteile. Somit können Sie bei einer Neugestaltung eines Systems die optimalen Interaktionsformen auswählen.

Zeitbedarf

Für die Bearbeitung des Kapitels „Interaktionsformen“ sind **5 Stunden** vorgesehen.

Die Unterkapitel dieses Kapitels sind:

- 10.1 Worum geht es im Kapitel Interaktionsformen?
- 10.2 Effizienz von Eingaben
- 10.3 Interaktionen mittels Tastatur
- 10.4 Interaktionen mittels Maus
- 10.5 Weitere Dialogformen
- 10.6 Zusammenfassung

10.1 Worum geht es im Kapitel Interaktionsformen?

Wir benutzen am Computer die Tastatur und Maus so selbstverständlich nebeneinander, dass viele Systemdesigner und Software-Architekten sich kaum mehr Gedanken machen, welche Vor- und Nachteile die verschiedenen Interaktionsformen haben.

Bereits im Kapitel „Hardware für die Kommunikation“ haben wir verschiedene Ein- und Ausgabegeräte betrachtet. In diesem Kapitel knüpfen wir an die dort behandelten Ein- und Ausgabegeräte an und zeigen Vor- und Nachteile der Interaktionsformen auf.

Einführendes Beispiel

Derzeit sind Ribbons modern. Bei **Ribbons** handelt es sich um eine Art Kombination von klassischer Menüsteuerung und Symbolleiste. Ribbons wurden durch Microsoft Office 2007 bekannt.



Abbildung 204 : Ribbon aus Microsoft Word 2007

Ribbons werden in immer mehr Anwendungen außerhalb der Office-Programme eingesetzt. Selbst dann, wenn das eigentliche Problem, welches zur Entwicklung des Ribbon-Konzepts geführt hat, bei den Anwendungen nicht gelöst werden muss.

Ob eine Interaktion mit der Tastatur oder der Maus stattfindet, ob Menüleisten oder Ribbons eingesetzt werden, ob sich Dialogfenster übereinander öffnen oder eben nicht, dass wird bei noch zu vielen Anwendungsentwicklungen eher als „Geschmackssache“ und nicht als die ermittelte optimale Lösung angesehen.

In diesem Kapitel „Interaktionsformen“ beschäftigen wir uns mit den unterschiedlichen Möglichkeiten und den zugehörigen Vor- und Nachteilen.

10.2 Effizienz von Eingaben

Bevor wir die Interaktionsformen mittels Tastatur und Maus betrachten, soll hier sozusagen als Vorbereitung auf empirische Effizienzmessungen von Fitts und Hick eingegangen werden. Früher standen Effizienzbetrachtungen oftmals im Vordergrund von Usability-Analysen, da durch verringerte Interaktionszeiten ein schnelleres Arbeiten ermöglicht wird.

Die Unterkapitel dieses Kapitels sind:

- 10.2.1 Fitts' Law
- 10.2.2 Hick's Law
- 10.2.3 GOMS

10.2.1 Fitts' Law

Fitts hat die mittlere **Zeit für die Positionierung einer Maus** mittels mathematischer Formeln beschrieben (siehe z.B. http://en.wikipedia.org/wiki/Fitts's_law (http://en.wikipedia.org/wiki/Fitts's_law)). Wir wollen hier die mathematische Betrachtung außer Acht lassen, da die genaue Berechnung von zu vielen Annahmen abhängt und uns bei der gebrauchstauglichen Gestaltung nicht weiter hilft.

Erstaunlich ist, dass Fitts [Fitts54] seit 1954 und somit lange vor der Erfindung der ersten Maus ein Modell entwickelt hat, bei dem es um die Bestimmung der Bewegungszeiten für grafische Interaktionen geht.

Die wichtige Aussage aus Fitts' Law ist, dass die **Zeit zur Positionierung** einer Maus abhängt von der **Entfernung der Maus** vom Ziel und der **Größe des Ziels**. Dabei kann das Ziel umso kleiner sein, umso geringer die Entfernung zwischen Maus und Ziel ist, denn das wichtige Maß ist der **Quotient aus Entfernung und Zielgröße**.

Beispiel

Wir kennen die Auswirkungen aus Fitts' Law im täglichen Umgang recht gut. Versucht man die Maus in einem Text zwischen zwei Buchstaben zu positionieren, so fällt es uns bei einer großen Schrift viel leichter als bei einer kleinen Schrift.

Folgerungen aus Fitts' Law für die Umsetzung:

- Zielobjekte der Interaktion müssen gut **erkannt, gefunden und getroffen** werden können. Somit sollten die Ziele nicht zu klein dargestellt werden.
- Bei Checkboxen auch den Text anklickbar machen.
- Im Text nicht nur wenig Zeichen (z.B. Weitere Hinweise gibt es „hier“ oder „>>>“), sondern einen ganzen Bereich anklickbar machen.
- Eine Besonderheit ist, dass Ziele in den Ecken des Bildschirms auch sehr klein sein können, da die Maus nicht über die Ecken hinaus gezogen werden kann. Deshalb werden die Ecken besonders gut und schnell getroffen.
- Wenn mehrere Interaktionen notwendig sind, so sollten diese möglichst an der aktuellen Eingabe vorhanden sein.

Wer sich mit der Interpretation von Fitts' Law zur Verbesserung der Gebrauchstauglichkeit bei MS Office 2007 interessiert, dem sei folgender Blog-Eintrag empfohlen:

<http://blogs.msdn.com/jensenh/archive/2006/08/22/711808.aspx> (<http://blogs.msdn.com/jensenh/archive/2006/08/22/711808.aspx>)

10.2.2 Hick's Law

Hick's Law [Hick52] beschreibt mathematisch die Zeit, die benötigt wird, um ein **Mausziel** aus einer Menge von möglichen Zielen zu **erkennen** und **auszuwählen**. Auch hier ist es wie bei Fitts' law, dass wir die mathematische Betrachtung vernachlässigen, da viele andere Einflussgrößen vorhanden sind.

Diese Schlussfolgerung aus Hick's Law ist noch heute gültig:
eine Auswahl aus vielen Alternativen benötigt mehr Zeit als eine Auswahl aus wenigen Alternativen.

Wenn man jedoch statt **a) 8 Menüpunkte** nur **b) 2 Menüpunkte mit je 4 Untermenüpunkten** anbietet, so ist aufgrund der doppelten Auswahl die Variante b) im Prinzip langsamer, wobei die Unterscheidbarkeit der Menüpunkte, die Beschriftungen etc. einen entscheidenden Einfluss haben. Die Unterschiede befinden sich jedoch im Bereich deutlich unter einer Sekunde.

Demgegenüber hat Krug [Krug06] unter Einbeziehung des gedanklichen Aufwands bei der richtigen Auswahl einen anderen Ansatz für die Navigation in Webseiten:

"Es ist egal, wie oft ich klicken muss, solange jeder Klick eine gedankenlose, unzweideutige Auswahl ist." [Krug06]

10.2.3 GOMS

Ursprünglich wurde das GOMS-Modell zur Usability-Evaluation verwendet. Heutzutage ist es noch immer ein guter Ansatz um eine quantifizierbare Bewertung der Interaktionszeiten mit Softwaresystemen zu erhalten. Im GOMS-Modell wurden durchschnittliche Zeiten für typische Interaktionen genannt und mit Abkürzungen versehen.

Abkürzung	Zeit / Sekunden
K (=Keyboard, Tastatureingabe, Mausklick)	0,2
P (=Pointing; Mauspositionierung)	1,1

H (Homeing; Wechsel Maus-Tastatur)	0,4
M (Mentally prepare; Mentale Reaktionszeit)	1,35

Tabelle 13 : GOMS-Modell, Zeiten für einzelne Aktionen

Die Zeiten berücksichtigen weder einen Lerneffekt noch eine Ermüdung. Auch wird zwischen den Benutzerklassen nicht unterschieden und möglicherweise im Ablauf auftretende Fehler werden nicht berücksichtigt. Und doch ist das GOMS-Modell sehr gut geeignet um anhand der Abkürzungen eine komplette Interaktion einmal in seine Einzelteile zu zerlegen.

Aufgabe 10.2.3-A1

Nehmen Sie mit einem Camcorder die folgenden Interaktionen auf und vergleichen Sie die Zeiten nach dem GOMS-Modell mit ihren wirklichen Zeiten.

Aufgabe: Einfügen eines Bildes in ein leeres Dokument (MS Word oder OpenOffice) auf zwei verschiedene Arten.

- Verwenden von Copy-and-Paste mit Hilfe von Tastatur und Maus
- Einfügen nur mit Hilfe der Maus über die Menüstruktur

Anhand des Videos können Sie die einzelnen vorgenommenen Interaktionsschritte notieren und dann über die GOMS-Zeiten die theoretische Gesamtzeit errechnen. Diese theoretischen Zeiten können Sie mit den real benötigten Zeiten aufgrund des Videos sehr gut vergleichen. Auch können Sie erkennen, für welche Interaktionsschritte Sie real Zeit benötigt haben, die in dem GOMS-Modell nicht vorgesehen ist.

10.3 Interaktionen mittels Tastatur

Eine Tastatur wird immer dann verwendet, wenn viele Daten eingegeben werden müssen. Sofern eine Tastatur vorhanden ist, bietet sich diese auch für die Steuerung der Dialoge an. Wir müssen also die Steuerung eines Dialogs unbedingt von der Eingabe von Massendaten, wie zum Beispiel zur Erstellung eines Textes getrennt betrachten.

Die Dialogsteuerung mittels Tastatur war lange Zeit die einzige Art mit dem Computer zu interagieren. Die Kommandos wurden direkt in eine „Befehlszeile“ eingegeben. Bestes Beispiel sind DOS und Unix/Linux.

Bei der Eingabe von Daten sind die Finger auf der Tastatur. Somit bietet sich die Tastatur auch heute in vielen Fällen zur zusätzlichen Steuerung an, da der „Griff zur Maus“ entfällt.

Die Unterkapitel dieses Kapitels sind:

[10.3.1 Kommandosysteme](#)

[10.3.2 Funktionstasten](#)

[10.3.3 Tastenkombinationen](#)

[10.3.4 Cursorblock und Pfeiltasten](#)

10.3.1 Kommandosysteme

Unter der Dialogart „Kommandosystem“ wird die direkte Eingabe von Kommandos in eine Systemzeile verstanden.

Beispiel

Beispiel für Benutzer von Windows:

Windows-Benutzer können das Kommandofenster unter „Start“ | „Ausführen“ und der Eingabe von „cmd“ erreichen. Sie sehen den Pfad des aktuellen Verzeichnisses und können beispielsweise das Kommando „dir“ für die Auflistung der Inhalte erhalten. (Eine Übersicht über die möglichen Kommandos finden Sie unter „Start“ | „Hilfe- und Supportcenter“ und der Suchen nach „Befehlszeilenreferenz“.

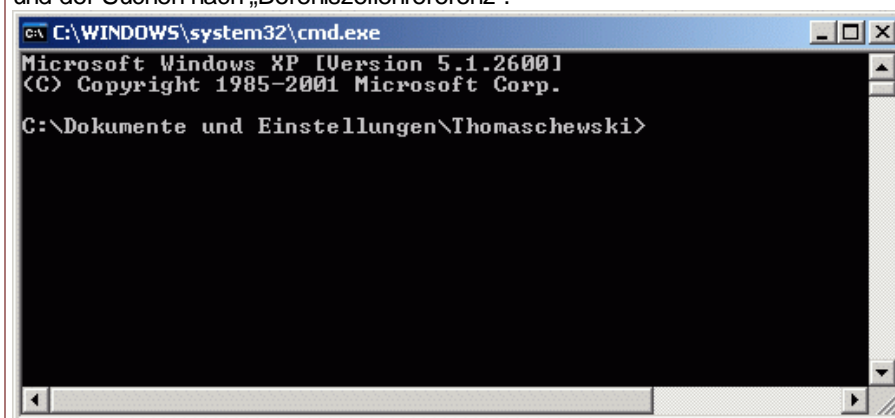


Abbildung 205 : Windows Kommandofenster

Noch heutzutage laufen viele Linux-Serversysteme ohne grafische Benutzeroberfläche, da die Administratoren die benötigten Befehle auswendig kennen und effizienter über die Kommandozeile arbeiten.

Vorteile von Kommandosystemen

- Komplette Befehlseingabe kann über die Tastatur erfolgen und wird automatisch in einem Befehlspeicher als „Historie“ abgespeichert. Damit sind Eingaben (z.B. für andere Administratoren) leicht nachvollziehbar.
- Experten können effizient mit der Kommandozeile arbeiten, da Kommandos mit Parametern komplexe Aktionen ausführen können, für die ansonsten viele Interaktionsschritte notwendig wären.
- Kommandos können in Batch-Dateien zu ganzen Kommandoblöcken zusammengefasst werden.

Nachteile von Kommandosystemen

- Hoher Lernaufwand, da Kommandos direkt aus der Erinnerung abgerufen werden müssen.
- System führt die Befehle ohne visuelle Rückmeldung aus, daher „Try-and-Error“. Nicht sinnvoll einsetzbar.
- Tippfehler sind leicht möglich.

10.3.2 Funktionstasten

Funktionstasten F1 – F12 (bzw. bei Apple F1 – F15) bieten eine **sehr gute Möglichkeit** die **Interaktionszeiten** einer grafischen Oberfläche zu **reduzieren**. Die Funktionstasten sollten bei der Auslieferung einer Software vorbelegt sein, doch sollte sich die Belegung individualisieren lassen.

Die Funktionstaste F1 sollte stets einen Schnellzugang zu den Hilfetexten der Software ermöglichen.

Wenn Sie betriebliche Bürosoftware erstellen, sollten Sie in Erwägung ziehen ihren Benutzern eine schmale, aufklebbare Leiste mit den vorbelegten Funktionstasten mitzuliefern. So muss der Benutzer ihre Funktionen nicht erst erlernen, sondern kann gleich bei Beginn die Funktionstasten zur effizienteren Bearbeitung verwenden.

Alternativ bieten sich viele Dinge des Bürobedarfs an, damit der Benutzer ihre Belegung der Funktionstasten „immer vor Augen“ hat: bedruckte Tassen, Kalender etc. Lassen Sie sich etwas einfallen um den Lerneffekt zu unterstützen.

10.3.3 Tastenkombinationen

Für Tastenkombinationen unter Verwendung der Tasten *Shift*, *Alt* und *Strg* gelten prinzipiell dieselben Überlegungen wie für die im letzten Unterkapitel behandelten Funktionstasten.

Vergleich Tastenkombination – Funktionstasten

- Bei der Lernförderlichkeit sind die Tastenkombinationen gegenüber den Funktionstasten im Vorteil, da sich ein Buchstabe als Funktionsanfang sinnvoll einsetzen lässt (*Strg* + *f* = finden; *Strg* + *z* = zurück).
- Auch die Erreichbarkeit der Tastenkombinationen ist besser als bei den Funktionstasten, da der Handballen nicht von der Unterlage genommen werden muss.
- Jedoch ist bei der Belegung der Tastenkombinationen noch stärker als bei den Funktionstasten auf das schon vorhandene mentale Modell des Benutzers zu achten. Es ist also zu berücksichtigen, welche anderen Tastenkombinationen dem Benutzer schon bekannt sind.
- Die Erreichbarkeit der Tastenkombinationen muss an verschiedenen Tastaturen getestet werden. Einige Tastenkombinationen sind – auch wenn der Buchstabe für den Funktionsanfang sinnvoll ist – schlechter erreichbar als andere.

Tastenkombinationen und Funktionstasten aus Sicht der Barrierefreiheit

- Absolute Computeranfänger und Benutzer mit motorischen Einschränkungen werden Tastenkombinationen vermeiden. Das ist nicht weiter beachtenswert, sofern alle über Tastenkombinationen erreichbaren Funktionen auf alternativen Interaktionswegen (z.B. Menüs) erreichbar sind.
- Für Blinde und sehbehinderten Benutzer sind Tastenkombinationen als Interaktionsform sehr gut geeignet.

10.3.4 Cursorblock und Pfeiltasten

Für die Bewegung innerhalb einer visuellen Oberfläche ist der Cursorblock mit den vorbelegten Kommandos *Einf*, *Entf*, *Pos1*, *Ende*, *Seite hoch*, *Seite runter* sowie der Pfeiltasten mit den Pfeilen zur Bewegung nach links, rechts, oben und unten sehr sinnvoll.

Sofern eine Software erstellt wird, bei der neben Interaktionen mit der Maus auch Eingaben mittels Tastatur notwendig sind, ist es oftmals viel effizienter für den Benutzer den Cursorblock und die Pfeiltasten zu benutzen, da gemäß Fitts' Law und dem

GOMS-Modell ein Wechsel zwischen Tastatur und Maus weitgehend vermieden werden sollte.

Auch im Bereich der Formulargestaltung auf Webseiten sollte unbedingt darauf geachtet werden, wohin der Cursor bei der Benutzung der „Tab-Taste“ springt.

10.4 Interaktionen mittels Maus

Heutzutage ist die Bedienung eines Büro-Computers mittels Maus und Tastatur die häufigste Interaktionsform. In diesem Kapitel werden wir unterschiedliche Interaktionsformen vorstellen, die für Zeigegeräte mit relativer Positionierung gelten. Die in diesem Kapitel getroffenen Aussagen gelten somit nicht nur für die Computer-Maus, sondern auch für Touchpads, Trackballs und Joysticks.

Die Unterkapitel dieses Kapitels sind:

- 10.4.1 Direkte Manipulation und WIMP
- 10.4.2 Der Nachteil von WIMP und andere Konzepte
- 10.4.3 Fenstersysteme
- 10.4.4 Kritische Betrachtung von Dialogfenstern
- 10.4.5 Dialogfenster mit Erweiterungen
- 10.4.6 Menüs
- 10.4.7 Icons
- 10.4.8 Ribbons
- 10.4.9 Kontextmenü
- 10.4.10 Mausgesten

10.4.1 Direkte Manipulation und WIMP

In dem Kapitel „[Interaktionen mittels Tastatur](#)“ haben wir ausschließlich Interaktionen betrachtet, die durch Eingabe von Text in eine Befehlszeile oder Drücken von Tasten aufgerufen wurden. Die direkte Manipulation wurde in den Jahren 1980 - 1984 von Xerox und Apple als eine neue Interaktionsform entwickelt. Zugrunde liegt dieser Interaktionsform das WIMP-Konzept.

- **W** = Windows (Fensterdarstellung)
- **I** = Icons
- **M** = Menüs
- **P** = Pointing Device (Zeigegerät)

Die direkte Manipulation verwendet demnach bildlich dargestellte Objekte, die auf Interaktionen mit den Zeigegeräten reagieren und entsprechende Funktionen aufrufen.

Die sehr großen Vorteile der direkten Manipulation sind:

- Aufgrund der anklickenden, beschrifteten Menüs und Icons wurde die **Lernförderlichkeit** sehr stark verbessert und die Anwendungen wurden so einem **großen Benutzerkreis** ohne umfangreiche Schulungen oder umfangreiches Selbststudium zugänglich. Die **Objekte und Aktionen** sind für den Benutzer **direkt erkennbar**.
- Auf Aktionen erfolgt eine **unmittelbare Reaktion**, da verbunden mit dem Anklicken eine Änderung des Icons oder des Menüs erfolgt.
- Funktionalitäten und Abläufe können mittels **Metapher** auf eine nachvollziehbare Aktion aus der **Lebenswelt der Benutzer** abgebildet werden.
- Es werden **motorische Handlungen** vorgenommen, die der realen Welt mehr entsprechen als die Eingabe von Kommandos.

Definition

„Die direkte Manipulation ist eine Interaktionsform, bei welcher der Benutzer den Eindruck hat, direkt mit Objekten auf dem Bildschirm zu interagieren.“ (aus Heinecke [\[Heine04\]](#))

Der große Unterschied zu den Kommandosystemen ist, dass der Benutzer mit Objekten interagiert.

Heinecke beschreibt die Vorteile der direkten Manipulation sehr gut mit: *„Die direkte Manipulation erlaubt eine angstfreie Benutzung, da nahezu alle Interaktionsformen reversibel sind. Sie vermittelt Zutrauen und Sicherheit auch deswegen, weil der Dialog stark benutzerbestimmt ist. Den Benutzern stehen meist alle Möglichkeiten der Manipulation zur Verfügung. Sie können das Objekt, das sie bearbeiten möchten, in der Regel frei auswählen und bei der Arbeit zwischen allen Objekten und Anwendungen wechseln. Syntaktische Fehler, wie sie bei Kommandodialogen auftreten können, sind ausgeschlossen. Es sind kaum Fehlermeldungen nötig, da viele Fehlhandlungen gar nicht möglich sind.“* [\[Heine04\]](#)

Mit diesen Objekten kann der Benutzer (vergl. [Heine04])

- Selektieren
- Bewegen
- Öffnen
- Löschen
- Erzeugen
- Kopieren
- Ablegen
- Drucken

Diese Basisinteraktionen ermöglichen zusammengesetzte Interaktionen wie „Drag and Drop“. Wir werden auf diese Basisinteraktionen nochmals im Bereich der Dialoge für Touchbildschirme eingehen.

H i n w e i s

Der Begriff **Direkte Manipulation** stammt aus einer Zeit, als die absolute Positionierung mittels Touchscreens und Bildschirmtablett bei Büro-Computern noch nicht etabliert war. Somit meint der Begriff eigentlich die „**Direkte Manipulation mit relativ positionierenden Zeigegeräten**“ (Maus, Touchpad etc.), als Grundlage von WIMP. Die „**Direkte Manipulation mit absolut positionierenden Zeigegeräten**“ (Touchscreen, Bildschirmtablett etc.) stellt eigene Anforderungen an die Planung der Interaktionsformen einer gebrauchstauglichen Software, wie wir in späteren Unterkapiteln erkennen werden.

10.4.2 Der Nachteil von WIMP und andere Konzepte

Das WIMP-Konzept hat viele Vorteile, aber auch Nachteile:

- Schnell schreibende Benutzer werden durch häufige Wechsel zwischen der Tastatur zur Eingabe von Massendaten und der direkten Manipulation mit der Maus langsamer als bei einem rein tastaturgesteuertem Programm.
- Die Menüs und Icons müssen auf der Oberfläche gesucht werden. Dies kann bei umfangreichen Interaktionen sehr aufwändig sein. Man denke beispielsweise an Microsoft Word 2003, Photoshop oder andere sehr umfangreiche Programme.

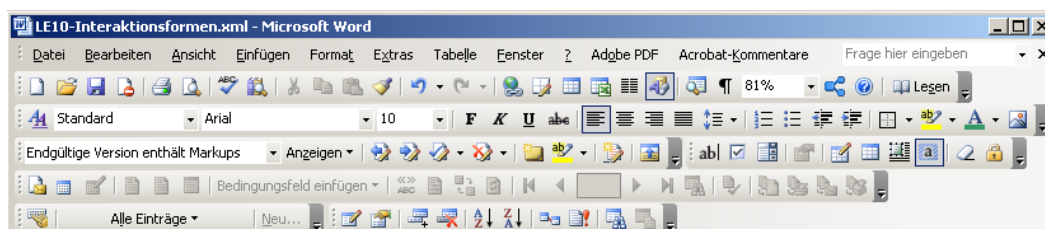


Abbildung 206 : Nachteil des WIMP-Konzepts – Suchen der richtigen Funktion

Wenn mit umfangreichen Datenmengen (Dateien, Suchergebnissen etc.) oder mit umfangreichen Funktionalitäten gearbeitet wird, dann stößt das WIMP-Konzept schnell an seine Grenzen.

Seit vielen Jahren werden Alternativen zum WIMP-Konzept erforscht, doch der Durchbruch gelingt wie in vielen Fällen erst, wenn die technischen Systeme kostengünstig sind und das Konzept auch außerhalb der Business-Anwendungen gefragt ist.

Hier ist zu unterscheiden zwischen den **Optimierungen von WIMP** wie beispielsweise dem **Cover-Flow** als Suchfunktionalität in homogenen großen Datenmengen und Anwendungen, die nicht auf WIMP setzen, beispielsweise das iPhone und die Spielekonsole Wii.



Abbildung 207 : Cover-Flow als Optimierung der Suchfunktionalität in homogenen großen Datenmengen

Seit der Einführung des iPhone (Jan. 2007), der Spielekonsole Wii (Dez. 2006) und dem iPad (April 2010) werden Konzepte jenseits von WIMP vermehrt diskutiert und erforscht.

Aufgabe 10.4.2-A1

Suchen Sie gute Artikel oder Vortragsfolien in denen entweder die Optimierung des WIMP-Konzepts oder Anwendungen jenseits von WIMP überlegt und genannt werden.
Schicken Sie diese Artikel an ihren Modulbetreuer oder stellen Sie diese in die entsprechenden Foren.

10.4.3 Fenstersysteme

Fenstersysteme in denen die Dateiverwaltung stattfindet und in denen Programme ablaufen kennen wir alle. Die Fenstersysteme sind der Hauptbestandteil des WIMP-Konzepts und auch wenn Fenster sich je nach Betriebssystem leicht unterscheiden, so sind die Hauptfunktionalitäten ähnlich.

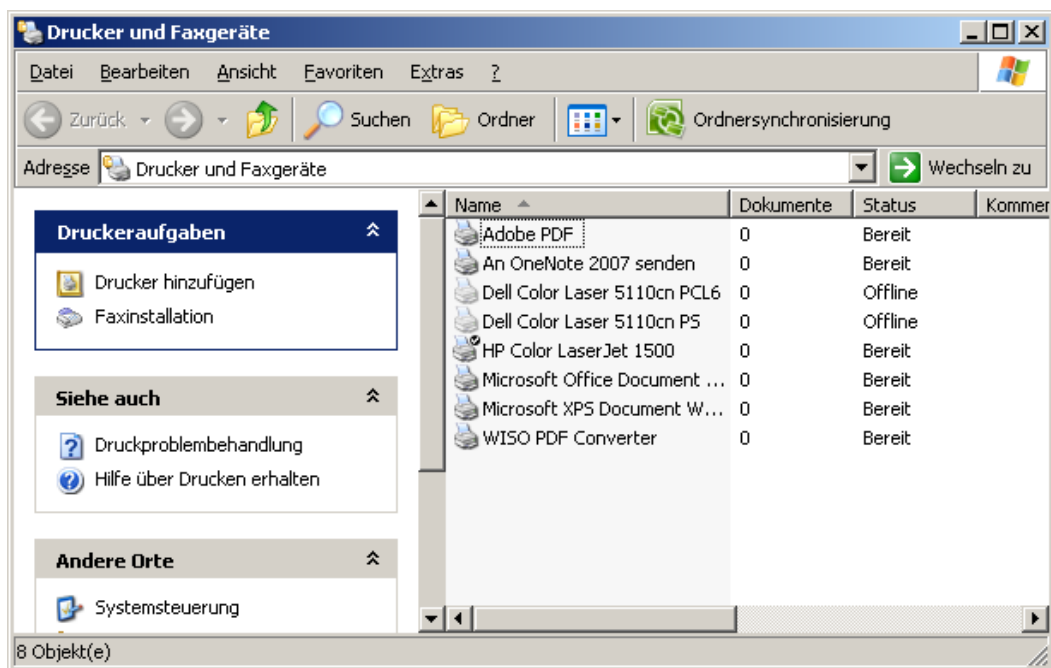


Abbildung 208 : Typisches Fenster eines Betriebssystems auf Grundlage des WIMP-Konzepts

- Hauptfunktionalitäten eines Fensters
Steuerung des Fensters: Bewegen, Größe verändern, Schließen
- Steuerung der Anwendung: Menüleiste, Icons, Scrollbalken

Bei den Fenstern wird unterschieden in:

- **Anwendungsfenster:** Dienen zur Steuerung einer Anwendung. Hierzu gehören auch die Fenster auf Betriebssystemebene (siehe Abbildung).
- **Unterfenster:** Werden durch Interaktionen innerhalb eines Anwendungsfensters erzeugt, beispielsweise wenn mehrere

Dokumente in einer Anwendung gleichzeitig angezeigt werden.

- **Dialogfenster:** Öffnen sich zur Dateneingabe. Oftmals öffnen sich Dialogfenster als Pop-Up-Fenster oder am linken oder rechten Rand einer Anwendung.
- **Mitteilungsfenster:** Enthalten eine Mitteilung, die mit einer Schaltfläche geschlossen werden.

Bei den Interaktionsmöglichkeiten wird unterschieden in:

- Modale Fenster: bei diesem Typ muss der Benutzer erst den Dialog im Fenster bearbeiten und das Fenster schließen, bevor er die Anwendung weiter nutzen kann.
- Nichtmodale Fenster: hier kann beliebig zwischen den Fenstern gesprungen werden („modeless“-Bedienung).

Bei den **Dialogfenstern** und **Mitteilungsfenstern** handelt es sich oftmals um **modale Fenster**

Bei den **Anwendungsfenstern** und **Unterfenstern** handelt es sich normalerweise um **nichtmodale Fenster**.

Hinweis

Da es für den Programmierer einfacher ist ein modales Fenster zu erzeugen, finden wir in den Anwendungen leider unnötig viele davon. So kann man in der Hauptanwendung „nicht mal eben was nachsehen“. Modale Fenster sollten nur eingesetzt werden, wenn es zwingende Gründe dafür gibt.

10.4.4 Kritische Betrachtung von Dialogfenstern

Der große Vorteil von (nichtmodalen) Fenstern ist, dass der Benutzer beliebig **zwischen verschiedenen Anwendungen wechseln kann**, ohne Anwendungen zu schließen. Somit wird kein Arbeitsmodus erzwungen, weshalb auch von einer „**modeless“-Bedienung** gesprochen wird.

Diese „modeless“-Bedienung ist beim Aufruf von Dialogfenstern dann nicht vorhanden, wenn die Weiterbearbeitung zwingend vom Abschluss des Unterdialogs abhängt (modale Fenster). Solch eine Kaskadierung von Dialogfenstern (siehe Abbildung) sollte jedoch unbedingt vermieden werden.

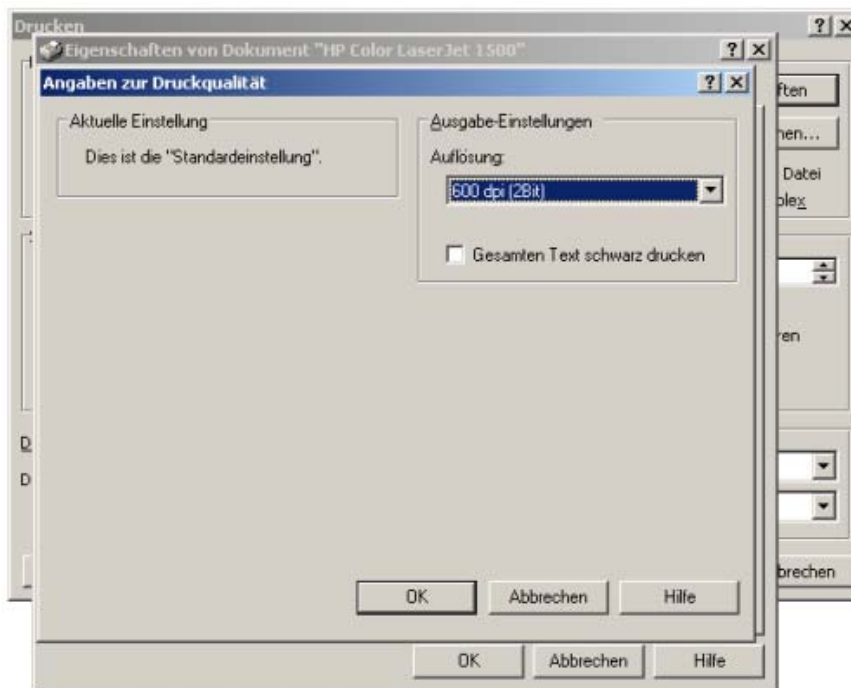


Abbildung 209 : Kaskadierung von Dialogfenstern

Probleme bei der Kaskadierung von modalen Fenstern:

- Wenn es sich bei Dialogfenstern um modale Fenster handelt, kann der Benutzer das übergeordnete Fenster nicht einsehen. Somit muss er gegebenenfalls notwendige Informationen dadurch erhalten, dass er das überlagernde Fenster schließt, sich die Informationen merkt und dann das Dialogfenster wieder öffnet.
- Der Benutzer ist es gewohnt, dass bei Dialogfenstern der OK-Button ganz unten ist. Somit kommt es häufig vor, dass gerade Routinebenutzer in der oben gezeigten Abbildung auf den untersten OK-Button klicken, anstelle des korrekten OK-Buttons des aktuellen Dialogfensters.

Um eine Erweiterung der Eingabedialoge zu erhalten können anstelle weiterer Dialogfenster Karteikarten verwendet werden.



Abbildung 210 : Karteikarten zur Erweiterung des Dialogfensters

Probleme bei der Verwendung von Karteikarten:

- Karteikarten eines Dialogfensters haben einen hohen Suchaufwand beim Benutzer zur Folge.
- Beim Benutzer besteht eine Unsicherheit bezüglich des OK-Buttons und des Abbrechen-Buttons, sofern auf einer Karteikarte Änderungen vorgenommen werden und weitere Änderungen auf anderen Karteikarten gewünscht werden.

10.4.5 Dialogfenster mit Erweiterungen

Eine alternative Erweiterung des Dialogfensters ist das Aufklappen. Dies hat den großen Vorteil, dass die bisher sichtbaren Einstellungen weiterhin für den Benutzer sichtbar bleiben. In der Abbildung wird das Dialogfenster für die Suche im Dokument dargestellt.

Positiv fällt hierbei auf:

- Die Karteikarten werden für einen Kontextwechsel (Suchen, Ersetzen, Gehe zu) verwendet und nicht für weitere Einstellungen.
- Die Auswahl des Dialogfensters kennt die Modi „reduziert“ und „erweitert“. Der Benutzer erhält beim Aufruf des Dialogs einen einfachen Suchdialog und wird nicht mit unnötigen Informationen überfrachtet.
- Bei der Erweiterung des Dialogs sind alle Informationen für den Benutzer sichtbar, im Gegensatz zu einem weiteren Pop-Up-Dialogfenster.
- Der Dialog ermöglicht eine „modeless“-Bedienung. Das heißt das Dialogfenster kann geöffnet bleiben und ein Wechsel in das Hauptdokument ist möglich, beispielsweise um einen Satz zu korrigieren.
- Weitere Einstellungen sind über ein ausklappbares Pull-Down-Menü gelöst. Hierfür wurde jedoch die innere Konsistenz zu anderen Dialogen mit den OK- und Abbrechen-Button verletzt.

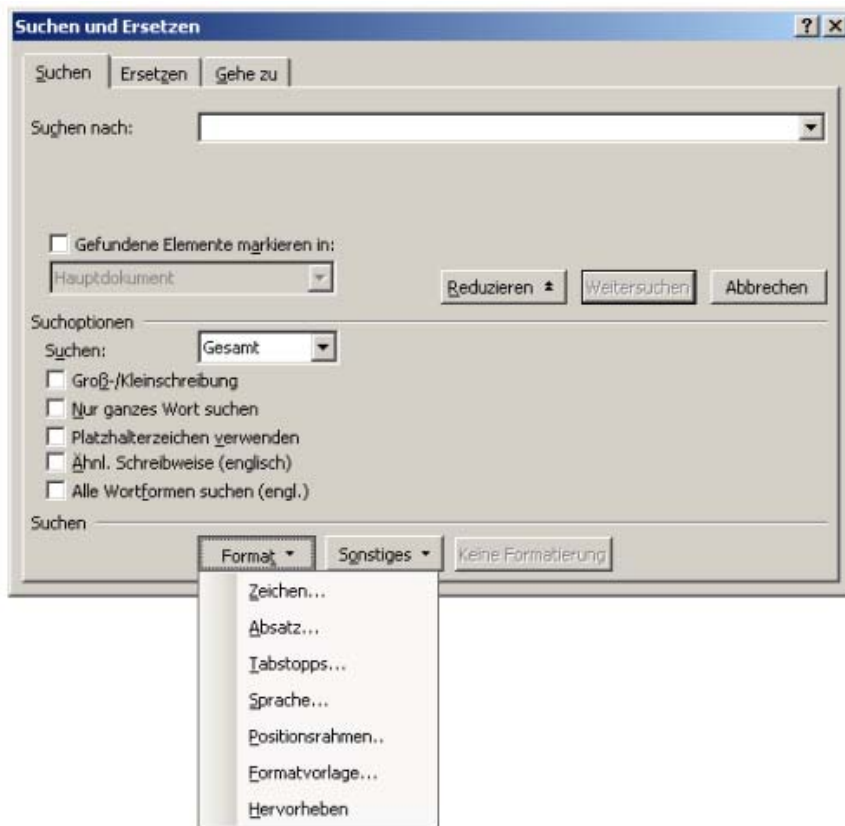


Abbildung 211 : Bei diesem Dialogfenster mit ausklappbarem „Erweitert“-Bereich

Selbstverständlich gibt es auch in anderen Betriebssystemen entsprechende Dialoge, beispielsweise der Druckdialog bei MAC OS 10.

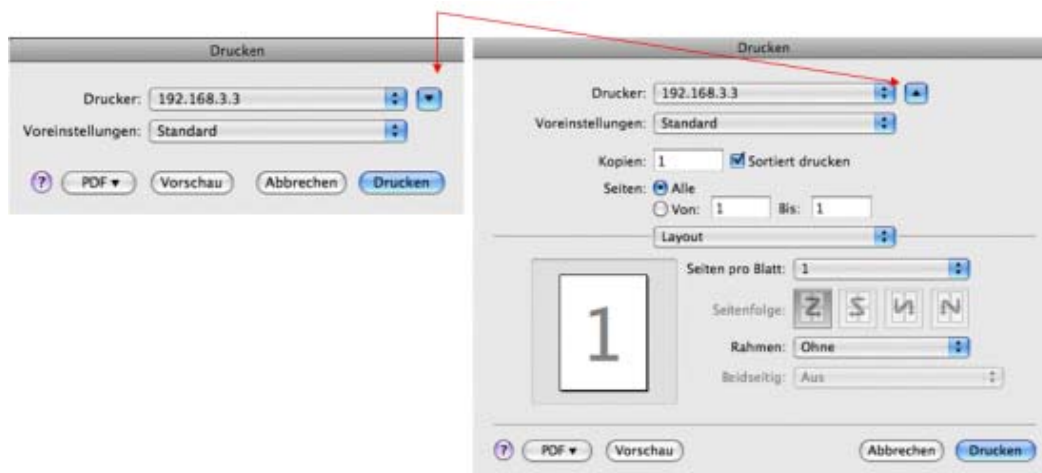


Abbildung 212 : Dialogfenster mit ausklappbarem „Erweitert“-Bereich. Links: eingeklappt – Rechts: ausgeklappt

Wie haben somit gesehen, dass sich Dialogfenster sehr unterschiedlich gestalten lassen. Bei einer gebrauchstauglichen Gestaltung sollte die Kaskadierung von Dialogfenstern vermieden werden. Besser ist die Verwendung eines ausklappbaren „Erweitert“-Bereich.

In den weiteren Unterkapiteln gehen wir auf die Elemente in den Fenstern ein: Menüs, Icons und Ribbons.

10.4.6 Menüs

Das klassische Menü dient zur Steuerung der Anwendung über Begriffe. Menüs geben eine Übersicht über die möglichen Kommandos mit denen sich das Programm steuern lässt. Somit sind Menüs eigentlich eine **Aufzählung der Kommandos in einer hierarchischen Anordnung**.

Ein Benutzer muss die Kommandos nicht erlernen, sondern nur die Kommandos im Menü wahrnehmen und auswählen. Dadurch wird das Gedächtnis zweifach stark entlastet:

- Die gelernten Wege mit der Maus laufen im **prozeduralen Gedächtnis** ab (links oben, „Datei“).

- Der Benutzer kann die zur Verfügung stehenden **Kommandos bei Bedarf** ansehen und auswählen.
- Das **Wiedererkennen** von Wörtern oder Zeichen aus dem Gedächtnis ist viel **leichter als die aktive Erinnerung**.

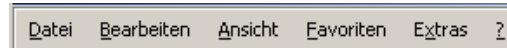


Abbildung 213 : Typische Menüleiste

Weitere Informationen zu unterschiedlichen Menüs sowie eine umfangreiche Auflistung der Vor- und Nachteile finden sich bei Heinecke [Heine04].

10.4.7 Icons

Sie haben im Kapitel Wahrnehmung gelernt, dass Bilder schneller aufgenommen werden können als Text. Auf dieser Idee basiert die Verwendung von Icons anstelle einer textlichen Beschreibung. Icons entlasten das Gedächtnis. Hierzu ist jedoch notwendig, dass die Icons auf Metaphern aus der realen Welt des Benutzers basieren.

Icons werden in vielen Programmen in einer Toolbar unterhalb der Menüleiste angeordnet.

Die Verwendung von **zu vielen Icons** in einer Icon-Liste ist andererseits **nicht sinnvoll**, da die **Aufnahmefähigkeit** des Gehirns begrenzt ist (Stichwort 7 ± 2 Chunks) und sich die Zeit für die Auswahl gemäß **Hick's Law** vergrößert.

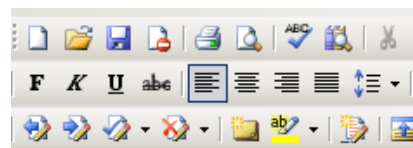


Abbildung 214 : Beispiel für Icons

Über die Kombination aus Menüleiste und Toolbar mit den Icons sind die Funktionalitäten auf zwei alternative Arten zu erreichen. Während die Toolbar in vielen Programmen individualisiert werden kann, ist der Zugriff über die Menüleiste statisch aufgebaut.

Literaturhinweise für die Erstellung von Icons

- 10 typische Icon-Fehler (Blogeintrag turbomilk) (http://www.turbomilk.com/blog/cookbook/criticism/10_mistakes_in_icor)
- Icons von Microsoft (<http://msdn.microsoft.com/en-us/library/ms997621.aspx>)
- Blogbeitrag bei boxesandarrows.com (http://www.boxesandarrows.com/view/icon_analysis)
- Blogbeitrag bei namics (http://blog.namics.com/2006/08/icon_usability.html)

10.4.8 Ribbons

Mit Einführung von Microsoft Office 2007 wurde das WIMP-Konzept verlassen. Microsoft hat erkannt, dass sich der Funktionsumfang der Anwendungen nicht mehr sinnvoll mit dem Konzept der Menüs und Icons abbilden lässt.

Der Begriff Ribbon (übersetzt: „Band“) ist eine Multifunktionsleiste. Das Ribbon ersetzt die Menüleiste sowie die Toolbar mit den Icons.

Beim Ribbon werden die Icons gemäß ihrer Funktionalitäten zusammengefasst und in Karteikartenreitern (=Tab bar) geordnet dargestellt.



Abbildung 215 : Ribbon aus Microsoft Word 2007

Die Einführung der Ribbon wurde von den Benutzern kontrovers diskutiert. Die zahlreichen Diskussionen haben oftmals den Ursprung in der mit den Ribbons verbundenen notwendigen Änderung der Mentalen Modelle der Benutzer. Da die Benutzerklassen oftmals den Routinebenutzern zuzurechnen sind, ist hier die Ablehnung des (mühsam) erlernten Konzepts besonders stark.

Beim Ribbon-Konzept von Microsoft entfällt die Individualisierbarkeit der Icon-Anordnung zugunsten einer immer gleich aussehenden Anordnung der Icons. Eine geringe Individualisierbarkeit ist zwar noch über Umwege erreichbar, aber nicht mehr

in dem von dem Benutzer erwarteten Umfang.

Ebenfalls entfallen ist die alternative Zugriffsmöglichkeit über Menüs oder Icons.

Das Ribbon-Konzept ist nur verwendbar für die Bearbeitung von Einzeldokumenten, wie diese in der Textverarbeitung oder Tabellenkalkulation vorkommt. Für Anwendungen, die auf einen Kontextwechsel angewiesen sind, beispielsweise um zwischen einer Darstellung der Suchergebnisse und einer Darstellung der Einzeldokumente hin und her zu springen, ist das Ribbon nicht konzipiert.

10.4.9 Kontextmenü

Fitts' Law wird beispielsweise beim Kontextmenü verwendet. Um die Entfernung der Maus vom Ziel zu verringern wird mit der rechten Maustaste ein Kontextmenü geöffnet.

In der Abbildung ist gut zu erkennen, dass Microsoft das Kontextmenü bei Word 2007 gegenüber Word 2003 noch weiter ausgebaut hat. Neu in Microsoft Office 2007 ist, dass das Kontextmenü sich dem Inhalt anpasst auf dem der Mauszeiger gerade steht. Somit spricht man von einem „Kontextsensitiven Menü“.

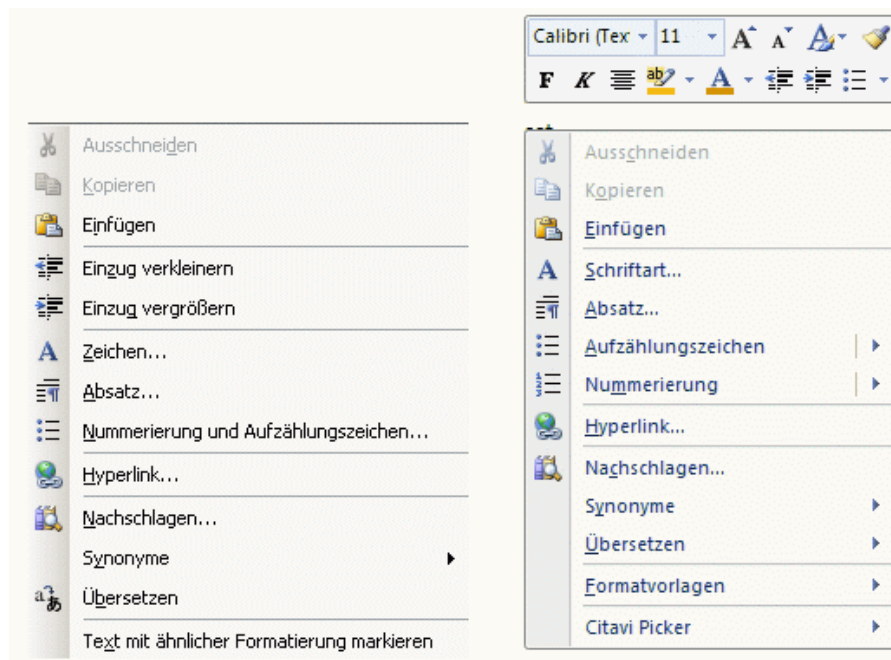


Abbildung 216 : Links: Kontextmenü aus MS Word 2003; Rechts: Kontextsensitives Menü aus MS Word 2007, bei dem insbesondere der obere Formatierungsblock in Abhängigkeit vom Mauszeiger kontextsensitiv ist

Der Einsatz von Kontextmenüs bietet sich überall dort an, wo insbesondere Experten, fortgeschrittene Benutzer und Routinebenutzer mit einer Anwendung arbeiten.

Die Hürde, die zum Aufruf des Kontextmenüs führt, ist die rechte Maustaste. Sehr viele Benutzer benutzen „gar nicht“ bis „selten“ die rechte Maustaste, obwohl die Benutzer den Zweck und den Nutzen des Kontextmenüs kennen.

Inzwischen bieten auch viele Tastaturen schon eine spezielle Taste für den Aufruf des Kontextmenüs an, deren Nutzung ebenfalls bislang als sehr gering eingeschätzt wird.



Abbildung 217 : Taste zum Aufruf des Kontextmenüs

Da die Nutzung der rechten Maustaste für viele Benutzerklassen ungewohnt ist, wird beispielsweise über ein Add-On für den Internet Explorer 8 ein Symbol eingeblendet. Wenn der Benutzer einen Text markiert wird ein Icon in der Nähe des markierten Bereichs eingeblendet.

In diesem Abschnitt beschäftigen wir uns mit den Technologien, die bislang selten an einem Büroarbeitsplatz zum Einsatz kommen.

Die Unterkapitel dieses Kapitels sind:

10.5.1 Dialoge mittels Stellregler

10.5.2 Dialoge für Touch und Grafiktablett

10.5.3 Dialoge für Spracheingabe

Ideen für weitere Dialoge nimmt der Autor unter jt@imut.de (mailto:jt@imut.de) sehr gerne entgegen.

10.5.1 Dialoge mittels Stellregler

Dreh- und Schieberegler gibt es schon sehr lange. Das man anhand eines Drehreglers die Lautstärke eines alten Autoradios einstellen kann, ist für die heutigen Generationen ein selbstverständlich klares Konzept.

Indem der Drehregler wie eine Art Joystick mit vier weiteren Funktionalitäten ausgestattet wurde, ergeben sich weitere Interaktionsmöglichkeiten. Somit wird aus einem einfachen Drehregler ein sogenannter Drehdrücksteller (in den Marketingunterlagen der Firmen werden hierfür die interessantesten Namen kreiert).



Abbildung 220 : Links "Click Wheel" des iPod (aus 2004); Rechts: Ergo-Commander im BMW X5

Mittels Drehdrücksteller sind sehr viele Interaktionsformen denkbar. Beispielsweise unterscheidet der iPod zwischen langsamen und schnellem „Drehen“. Wenn der Finger schnell, kreisförmig über das Click-Wheel geführt wird, dann erscheint eine sogenannte ABC-Liste für eine schnellere Navigation in einer Liste.

Betrachtung der Interaktionsformen des Drehdrückstellers

Bezüglich der Interaktionsformen ist die Verwendung des Drehdrückstellers für die Navigation durch Listen gut zu verwenden, auch wenn die Auge-Hand-Koordination dabei für ungeübte Benutzer eine Eingewöhnungszeit erfordert. Die kreisförmige Fingerbewegung hat eine lineare Listenbewegung (horizontal oder vertikal) zur Folge.

Schwierig wird es, wenn Affordance und Mapping nicht klar ersichtlich sind. Dies ist beispielsweise der Fall, wenn ein Drehdrücksteller zur Steuerung eines Navigationssystems verwendet wird, indem durch eine Drehbewegung in Karten gezoomt wird.

Übung 10.5.1-Ü1

Überlegen Sie das von Ihnen erwartete Verhalten, wenn Sie in eine Karte hineinzoomen möchten. Würden Sie nach links oder nach rechts drehen?

Lösung

Empirische Umfragen des Autors haben ergeben, dass einige Benutzer im Uhrzeigersinn drehen würden, andere entgegen des Uhrzeigersinns.

Bei der Verwendung neuer Interaktionsformen zur Steuerung immer komplexerer Anwendungen ist die Beachtung von Affordance und Mapping sowie die Auge-Hand-Koordination sehr wichtig.

10.5.2 Dialoge für Touch und Grafiktablett

Touchanwendungen und die Verwendung von Grafiktablets befinden sich noch in der aktuellen Entwicklung, sodass hier keine (abschließenden) Ergebnisse präsentiert werden können. Im Rahmen der Interaktionsformen gibt es bei der Entwicklung von Systemen mit absoluter Positionierung jedoch einige Dinge zu beachten.

Regeln für die Interaktionselemente von Touchdisplays:

- Die Interaktionselemente müssen **groß genug** und **genügend voneinander abgetrennt** sein, damit die Interaktion mit den Fingern problemlos möglich ist.

- Die Interaktionselemente müssen so angeordnet sein, dass der **Arm** bei der Eingabe **den Bildschirm nicht verdeckt**.
 - Daher wird eine Touch-Tastatur immer am unteren Bildschirmrand eingeblendet.
 - Daher eignet sich die obere Steuerungsleiste nicht für eine Dialogbedienung mittels Touch oder Grafiktablett, auch wenn die Interaktionselemente wie beispielsweise Ribbons groß genug wären.
- Die Software muss eine **hohe Toleranz gegenüber kurzem und langem Tippen** haben. Am Besten wird nur das Tippen an sich und nicht die Zeit oder eine langsame Bewegung während des Tippens für eine Interaktion ausgewertet.
- **Schnelle Fingergesten** können **gut** ausgewertet werden.
- Aufgrund der fehlenden taktilen Rückmeldung ist eine visuelle und je nach Umgebung und Anwendung auch eine akustische Rückmeldung bei Touchsystemen besonders wichtig. Die taktile Rückmeldung ist jedoch wesentlich besser ist, als eine visuelle und akustische Rückmeldung. Deshalb finden derzeit auf dem Gebiet der taktilen Rückmeldung für Touchanwendungen viele Forschungsarbeiten statt.
- **Nicht sinnvoll umsetzbar ist Drag and Drop**, also das Ziehen eines Objektes mit dem Finger in ein anderes Objekt.

Warum ist Drag and Drop nicht sinnvoll umsetzbar?

Sowohl bei der Bedienung mit dem Finger als auch bei der Bedienung mit einem Stift gibt es gegenüber der Bedienung mit der Maus einen gravierenden Unterschied. Die Maus bewegt sich in 2 Dimensionen. Finger und Stifte bewegen sich in 3 Dimensionen, wobei eine Änderung in der Höhe, also eine Fortbewegung vom Bildschirm, zu einem Verlust der Positionierung und damit zu einem Abbruch bei Drag and Drop führt.

Ist WIMP für Touchanwendungen geeignet?

Im Unterkapitel „[Direkte Manipulation und WIMP](#)“ wurden die Haupteigenschaften genannt, wie ein Benutzer mit Objekten einer WIMP-Anwendung unter Verwendung einer Maus oder einem anderen Zeigegerät mit relativer Positionierung arbeiten kann. Hier sollen in einer stark vereinfachten Sicht (ja/nein) diese Haupteigenschaften bezüglich der Eignung für Touchanwendungen betrachtet werden:

- Selektieren (ja)
- Bewegen (nein)
- Öffnen (ja)
- Löschen (nein)
- Erzeugen (nein)
- Kopieren (nein)
- Ablegen (nein)
- Drucken (ja)

Immer wenn es sich um eine einfache Auswahl (selektieren, öffnen etc.) handelt, dann können diese Manipulationen bei einer Interaktion mit dem Finger gut umgesetzt werden. Jedoch baut das WIMP-Konzept insbesondere auf die Manipulation mittels Drag and Drop.

Fazit:

- **Touchanwendungen** eignen sich heutzutage sehr **gut für** Systeme, an denen **kurze Interaktionen** stattfinden (Mobile Geräte, Kiosksysteme).
- Das bisherige **WIMP-Konzept** ist für Touch-Anwendungen **nicht sinnvoll verwendbar**.
- Die bisherigen Softwarelösungen sind auf eine Bedienung mit Maus und Tastatur ausgelegt und für eine kontinuierliche Bedienung mit dem Finger nicht gut geeignet.

Somit ist bei der vermehrten Verfügbarkeit von Touchbildschirmen eine komplette Neuerstellung von entsprechenden Softwareanwendungen notwendig.

10.5.3 Dialoge für Spracheingabe

Die Interaktionssteuerung mittels Sprache eignet sich sehr gut, wenn:

- Der Benutzer sich an einem **Platz ohne weitere Sprecher** (Mitarbeiter, Telefongespräche, Radio) befindet
- Das System die Sprache des Benutzers **trainieren** kann

- **Einzelne Anweisungen** gesprochen werden (beispielweise Adressen für ein Navigationssystem)

Somit ist die Sprachsteuerung geeignet für telefonische Anwendungen (z.B. Callcenter-Steuerung, telefonische Auskunftssysteme), elektronische Hilfesysteme, Übersetzungssystem, Wahl von Telefonnummern oder Eingabe von Adressdaten.

Eine gute Übersicht über den Einsatz von Sprachdialogen findet sich bei Dahm[Dahm06].

Beispiel

Es gibt Mobiltelefone, die sich über Spracheingaben steuern lassen. So kann beispielsweise das Nokia 6300 mit Bluetooth über ein Autoradio als Freisprecheinrichtung verwendet werden. Die Anwahl des Gesprächspartners erfolgt über eine Spracheingabe.

Schade nur, dass die Aussprache der Namen der Gesprächspartner nicht selbst aufgenommen werden kann. So muss man beim Sprechen des Namens die Betonung der synthetischen, englischsprachigen Stimme genau treffen, wenn man nicht mit jemand ganz anderem verbunden werden möchte.

Eine Rückmeldung, ob man die richtige Betonung getroffen hat, erfährt man erst, wenn der Gesprächspartner ans Telefon gegangen ist.

10.6 Zusammenfassung

- Sie kennen die Begriffe Fitts' Law, Hick's Law und GOMS und können die Kernaussagen benennen.
- Sie kennen die möglichen Interaktionsformen mittels Tastatur und können die Vor- und Nachteile von Kommandosystemen, Funktionstasten, Tastenkombinationen sowie mittels Cursorblock und Pfeiltasten benennen.
- Sie kennen die Möglichkeiten und Grenzen der Interaktion mit der Maus und anderen relativ positionierenden Zeigegeräten.
- Sie wissen, was man unter den Begriffen „Direkte Manipulation“, „Absolute Manipulation“ und „WIMP“ versteht.
- Sie können die Vorteile und Grenzen des WIMP-Konzeptes benennen.
- Sie können die Vor- und Nachteile von Fenstersystemen benennen und kennen die unterschiedlichen Gestaltungsmöglichkeiten von Dialogfenstern.
- Sie wissen um die Vor- und Nachteile von Menüs und Icons und in welchen Fällen der Einsatz von Ribbons angebracht ist.
- Sie kennen unterschiedliche Arten der Ausprägung eines Kontextmenüs.
- Sie kennen die Möglichkeit Mausgesten einzusetzen und können diese Möglichkeit bewerten.
- Sie kennen die Möglichkeiten und Grenzen des Einsatzes von Drehdrückstellern.
- Sie können bewerten, wann sich der Einsatz von Touchscreen-Anwendungen anbietet und wissen warum im Bereich der Desktop-Anwendungen noch Entwicklungsaufwand notwendig ist.
- Sie können eindeutige Beispiele benennen, wann sich ein Dialog für die Spracheingabe eignet.
- Sie wissen, wann Sie welche Interaktionsform einsetzen sollten und können den Einsatz anhand vorhandener Softwareanwendungen bewerten.

10.6.1 Selbsttests

Frage 10.6.1 - 1 - 1: Welches „Gesetz“ beschreibt die Zeit zur Positionierung in Abhängigkeit von der Entfernung und der Zielgröße?

- ☐ Fitts' Law.
- ☐ Hicks' Law.
- ☐ GOMS

Ok Zurücksetzen

Frage 10.6.1 - 2 - 1: Womit lassen sich die typischen Interaktionszeiten für den Wechsel von Maus und Tastatur abschätzen?

- ☐ Fitts' Law.
- ☐ Hicks' Law.
- ☐ GOMS

Ok Zurücksetzen

Frage 10.6.1 - 3 - 1: Worin wird beschrieben, wie sich die Zeiten für die Erkennung und Auswahl von der Anzahl der Menüpunkte ergeben?

- ☐ Fitts' Law.
- ☐ Hicks' Law.
- ☐ GOMS

Ok Zurücksetzen

Frage 10.6.1 - 4 - 1: Bei der Erstellung eines **Eingabedialogs für eine Internetanwendung** wird normalerweise eine Eingabe per Maus betrachtet. Welche Alternativen sollten angeboten werden?

- ☐ Eingabe über Kommandosystem.
- ☐ Eingabe über Funktionstasten
- ☐ Eingabe über Tastenkombinationen.
- ☐ Eingabe über Cursorblock und Pfeiltasten

Ok Zurücksetzen

Frage 10.6.1 - 5 - 1: Welche Interaktionsformen sind für Touch-Systeme sehr gut geeignet?

- ☐ Eingabe über Kommandosysteme.
- ☐ Eingabe über Funktionstasten.
- ☐ Eingabe über Tastenkombinationen.
- ☐ Eingabe über Cursorblock und Pfeiltasten
- ☐ Interaktionen über Icons
- ☐ Interaktionen über Kontextmenüs
- ☐ Modale Fenster
- ☐ Nichtmodale Fenster
- ☐ Spracheingabe

Ok Zurücksetzen

10.6.2 Übungen

Übung 10.6.2-Ü1

Wann sollten modale Fenster eingesetzt werden?

Lösung

Der Einsatz eines modalen Fensters bedeutet, dass zuerst der Dialog im Fenster geschlossen werden muss, bevor man mit der Hauptanwendung weiter arbeiten kann.

Modale Fenster sind immer dann notwendig, wenn eine zwingende Eingabe oder eine Eingabebestätigung erforderlich ist. Man sollte sich immer fragen, was passiert, wenn das Fenster einfach geöffnet bleibt und sich der Benutzer „später“ darum kümmert (z.B. weil er gerade angerufen wurde und noch schnell etwas anderes nachsehen möchte).

Ein typisch sinnvolles modales Fenster ist z.B. die Eingabe des Passworts beim Download einer kostenpflichtigen App.

10.6.3 Zusatzaufgaben

Diese Zusatzaufgaben können vom standortlokalen Betreuer als Einsendeaufgaben verwendet werden.

Aufgabe 10.6.3-A1

Schreiben Sie einen kleinen Erfahrungsbericht über den Umgang mit Mausgesten. Suchen und installieren Sie sich hierfür entsprechende Programme oder Browser-Plugins. Selbstverständlich ist das verwendete System, die Eingabegeräte, die Software für die Mausgesten etc. zu nennen.

[Der zeitliche Umfang dieser Einsendeaufgabe wird mit 4 Stunden abgeschätzt]

10.6.4 Rückmeldung

Zur ständigen Verbesserung des Moduls und zur realistischen Abschätzung des Arbeitsaufwands ist eine Rückmeldung gerade bei Online-Modulen sehr wichtig. Deshalb würde ich mich über eine kurze Angabe sehr freuen!

Ihre Daten werden an den Autor anonym übermittelt.

Frage 10.6.4 - 1 - 1: **Welche Bewertung geben Sie dieser Lerneinheit?**

- ☐ Sehr gut

- ☐ Gut
- ☐ Nicht so gut
- ☐ Schlecht
- ☐ Sehr Schlecht

Frage 10.6.4 - 1 - 2: **Wie lange haben Sie für die Bearbeitung dieser Lerneinheit insgesamt benötigt** (mit Sekundärliteratur, Aufgaben und Erstellung eigener Notizen, jedoch ohne die Einsendeaufgaben)?

Frage 10.6.4 - 1 - 3:

Was hat Ihnen in diesem Kapitel besonders gut gefallen?

Frage 10.6.4 - 1 - 4: **Was hat Ihnen in diesem Kapitel nicht gefallen und muss verbessert werden?**

Frage 10.6.4 - 1 - 5:

Weitere Anregungen, Ideen, Links, Glossareinträge usw.

Ok Zurücksetzen

Ergebnisse anonym senden an: jt@imut.de

Vielen Dank für Ihre Rückmeldung!