

Inhalt

1.	DEFINITION BENUTZERSCHNITTSTELLE	3
2.	ARTEN VON BENUTZERSCHNITTSTELLEN	3
2.1	EDITOR (ZEILENEDITOR)	3
2.2	ZEICHENORIENTIERTE BENUTZERÖBERFLÄCHEN	3
2.2.1	<i>Bildschirmmasken, Bildschirmformulare</i>	<i>3</i>
2.2.2	<i>Menüstrukturen</i>	<i>4</i>
2.3	GRAFISCHE BENUTZERÖBERFLÄCHE (GUI – GRAPHIC USER INTERFACE)	4
3.	NAVIGATION MIT MENÜS	4
4.	SOFTWARE ERGONOMIE	5
4.1	GENERELLES	5
4.2	ENTWURF VON BENUTZERÖBERFLÄCHEN	5
4.3	GRUNDSÄTZE	6
4.4	MASKENLAYOUT	6
4.4.1	<i>Fluchtlinien</i>	<i>6</i>
4.4.2	<i>Abstände</i>	<i>7</i>
4.5	GRUPPIERUNG VON INFORMATIONEN	7
4.6	FARBEN	8
4.7	SCHRIFT	8
4.8	DIALOGGESTALTUNG	9
4.8.1	<i>Defaults</i>	<i>9</i>
4.9	MENÜS	9
4.10	TABWEGE	10
4.11	WENIGER KOMFORTABLE GUI'S	11
4.11.1	<i>Dolphin Plus</i>	<i>11</i>
4.11.2	<i>Microsoft's Visual Studio</i>	<i>11</i>
4.11.3	<i>Microsoft's Visual Source Safe</i>	<i>11</i>
5.	BASIS GUI-ELEMENTE	12
5.1	DEFINITION	12
5.2	EINGABEFELD	12
5.2.1	<i>Verwendung</i>	<i>12</i>
5.2.2	<i>Beispiel</i>	<i>12</i>
5.3	DROP-DOWN-LISTE	12
5.3.1	<i>Definition</i>	<i>12</i>
5.3.2	<i>Verwendung</i>	<i>13</i>
5.4	CHECKBOX	13
5.4.1	<i>Definition</i>	<i>13</i>
5.4.2	<i>Verwendung</i>	<i>13</i>
5.4.3	<i>Beispiel</i>	<i>14</i>
5.5	AKTIONSTASTEN – PUSH-BUTTON	14
5.5.1	<i>Definition</i>	<i>14</i>
5.5.2	<i>Verwendung</i>	<i>14</i>
5.5.3	<i>Beispiel</i>	<i>14</i>
6.	DIALOGGESTALTUNG	15
6.1	BEISPIEL	15
6.2	MODAL / NICHT-MODAL	15
6.2.1	<i>Beispiel</i>	<i>15</i>

6.3	MDI (MULTIPLE DOCUMENT INTERFACE)	16
7.	DESIGN PATTERNS	16
7.1	MVC DESIGN-PATTERN	16
8.	ÜBUNGEN.....	18
8.1	18

1. Definition Benutzerschnittstelle

Unter dem Begriff Benutzerschnittstelle (engl. interface) ist der Übergang zwischen zwei Systemen zum Informationsaustausch zu verstehen. Die Benutzeroberfläche stellt die Schnittstelle zwischen Mensch und Maschine dar. Jede Anwendersoftware (Betriebssystem oder ein anderes Anwendungsprogramm) besitzt eine spezifische Benutzeroberfläche zu welcher die Gestaltung des Bildschirms, die Belegung der Tastatur, die Mausunterstützung und die Hilfefunktionen zählen. Während ältere Benutzerschnittstellen oft noch text- und kommandoorientiert (Kommunikation mit der Software über Text – Kommandos), sind moderne Benutzeroberflächen (die des Windows) grafikorientiert.

2. Arten von Benutzerschnittstellen

2.1 Editor (Zeileneditor)

Wer sich schon mit DOS-Befehlen am PC befasst hat weiss, dass im MS-DOS jeder Befehl eingetippt, anschliessend vom System ausgeführt, und im besten Fall in irgendeiner Form quittiert wird. Früher wurden auch Texte (z.B. Source-Programme, Dokumentationen) mit Zeileneditors erfasst. Um einen Fehler zu korrigieren, musste man erst die entsprechende Zeile aufrufen, anschliessend konnte die Korrektur eingegeben werden

2.2 Zeichenorientierte Benutzeroberflächen

2.2.1 Bildschirmmasken, Bildschirmformulare

Bildschirmmasken sind elektronische Formulare. Beim Ausfüllen springt der Cursor von Feld zu Feld. Mit diversen Tasten können Felder gezielt angesprungen oder übersprungen werden. Wichtig ist die benutzerfreundliche Anordnung der Felder (z.B. gleich wie auf der Vorlage).

Vorname 1	<input type="text" value="Uno"/>	Vorname 2	<input type="text"/>
Nachname	<input type="text" value="Acht"/>	Geschlecht	<input type="text" value="Männlich"/>
Geburtsname	<input type="text"/>	Geburtsdatum	<input type="text" value="07.07.1955"/>
Straßenname	<input type="text" value="Hellersbergstr."/>	Hausnummer	<input type="text" value="14"/>
Postleitzahl	<input type="text" value="41460"/>	Ort	<input type="text" value="Neuss"/>
<input type="button" value="Absenden"/>		<input type="button" value="Löschen"/>	

2.2.2 Menüstrukturen

Jede Online-Transaktion ruft mindestens ein Menü auf. Der Aufbau des Menüs richtet sich nach den Bedürfnissen der Applikation.



2.3 Grafische Benutzeroberfläche (GUI – Graphic User Interface)

Zu einer modernen Benutzeroberfläche gehören:

- Bitmapbildschirme (anstelle von textorientierten)
- Fenstertechnik
- Sinnbilder (ICONS)
- Maus
- WYSIWYG (What you see is what you get)

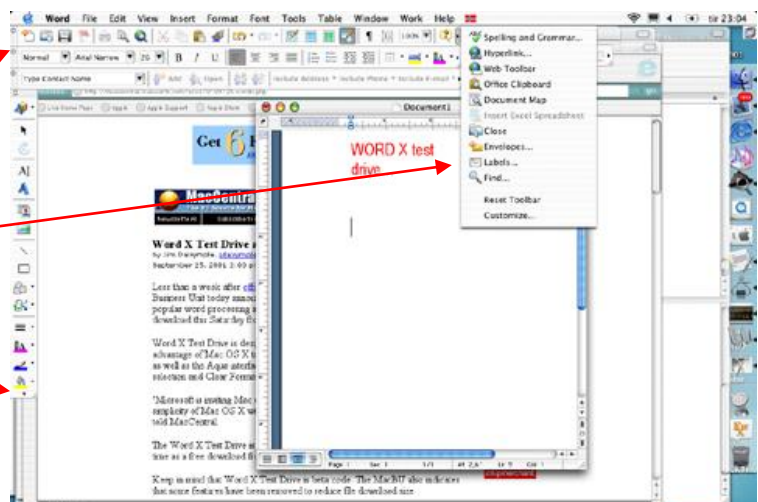
Beispiele: Windows XP, OS X, ...



3. Navigation mit Menüs

In moderneren Applikationen treffen wir folgende Komponente oft an:

- **Menübalken**
- **Aufklappmenüs**
- **Icons**



Mittels Cursor wird die jeweils gewünschte Funktion oder das gewünschte Untermenü ausgewählt.

„Negative“ oder auf eine andere Art gekennzeichnete Funktionen können zum momentanen Zeitpunkt nicht ausgeführt werden.

4. Software Ergonomie

4.1 Generelles

Unabhängig von ergonomischen Aspekten im Hardwarebereich (Bildschirmarbeitsplätze, Beleuchtung, etc.) kennen wir auch die sogenannte Software-Ergonomie. Ziel ist die Anpassung der Eigenschaften von Software an die psychischen Eigenschaften der damit arbeitenden Menschen. Unzureichende software-ergonomische Gestaltung führt zu erhöhten psychischen Belastungen, Kopfschmerzen, Augenflimmern, Stress und längerfristig auch zu körperlichen Beschwerden. Während der Entwicklung eines Werkzeugs beginnt man immer mit dem technischen Ansatz, d.h. das Werkzeug steht im Zentrum der Aufmerksamkeit und der Benutzer. Der Entwickler konzentriert sich vorerst auf die Funktionalität.

Wenn die technischen Probleme gelöst sind, formuliert der Benutzer seine Wünsche. Das Werkzeug wird entsprechend angepasst und mit den entsprechenden Benutzeroberflächen ausgestattet.

4.2 Entwurf von Benutzeroberflächen

Der Entwurf einer Oberfläche läuft in mehreren Phasen ab:

Aufgabenanalyse

- Beschreibung des Problems / System
- Use-Cases für zentrale Aufgaben

Aufgabenmodellierung

- Sammeln der Anforderungen an die Kommunikation
- Ergonomie-Ebenen (Aufgaben, semantische, syntaktisch, Interaktion)

Dialogspezifikation

- Interaktionen des Systems beschreiben mit Szenario (Story Bord)

Prototypkonstruktion

- Erstellen eines Prototyps (Beschreibung der Dialoge)

Auswertung

- Konfrontation des Benutzers mit dem Prototyp und Feedback.

Die Softwareergonomie setzt wissenschaftliche Erkenntnisse ein, um Arbeitsaufgaben, Arbeitsumgebung und Produkte an die physischen und mentalen Fähigkeiten und Grenzen von Menschen anzupassen und deren Gesundheit, Sicherheit, Wohlbefinden und Leistungsvermögen zu steigern.

Die Leitlinien zur Gestaltung von Software sind international festgelegt.

4.3 Grundsätze

- **Aufgabenangemessenheit**
Die Belastung des Benutzers zur Beherrschung der Oberfläche soll nicht grösser sein als nötig.
- **Selbsterklärungsfähigkeit**
- **Erwartungskonformität**
Der „erfahrene“ Benutzer erwartet, dass sich das neue System ähnlich verhält wie jenes, das er bereits kennt.
- **Fehlerrobustheit**
Auf Fehler des Benutzers soll das System mit veränderlichen Fehlermeldungen reagieren.
- **Erlernbarkeit**
Die Bedienung des Systems soll schnell und bequem gelernt werden können.

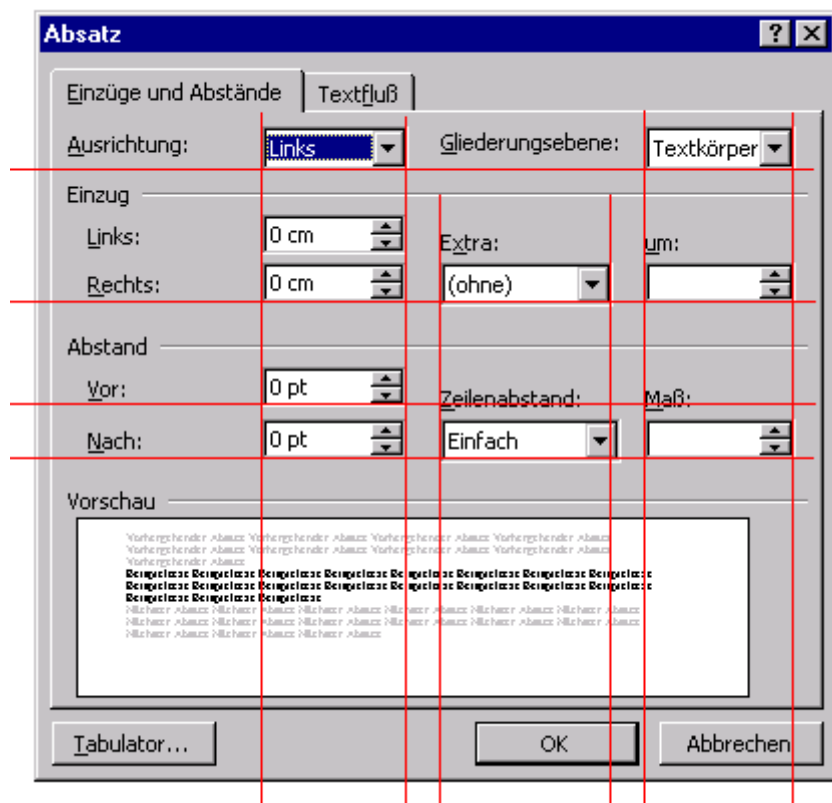
4.4 Maskenlayout

4.4.1 Fluchtlinien

Das Ausrichten der Bildelemente an WENIGEN Fluchtlinien verringert die Komplexität für das Auge. Es findet die gewünschten Informationen leichter, da es sich an optischen Strukturlinien orientieren kann.

Horizontale Fluchtlinien

Bei horizontalen Fluchtlinien steht in vielen Fällen die Aligierung von Texten (nicht von Kanten) im Vordergrund.



Vertikale Fluchtlinien

4.4.2 Abstände

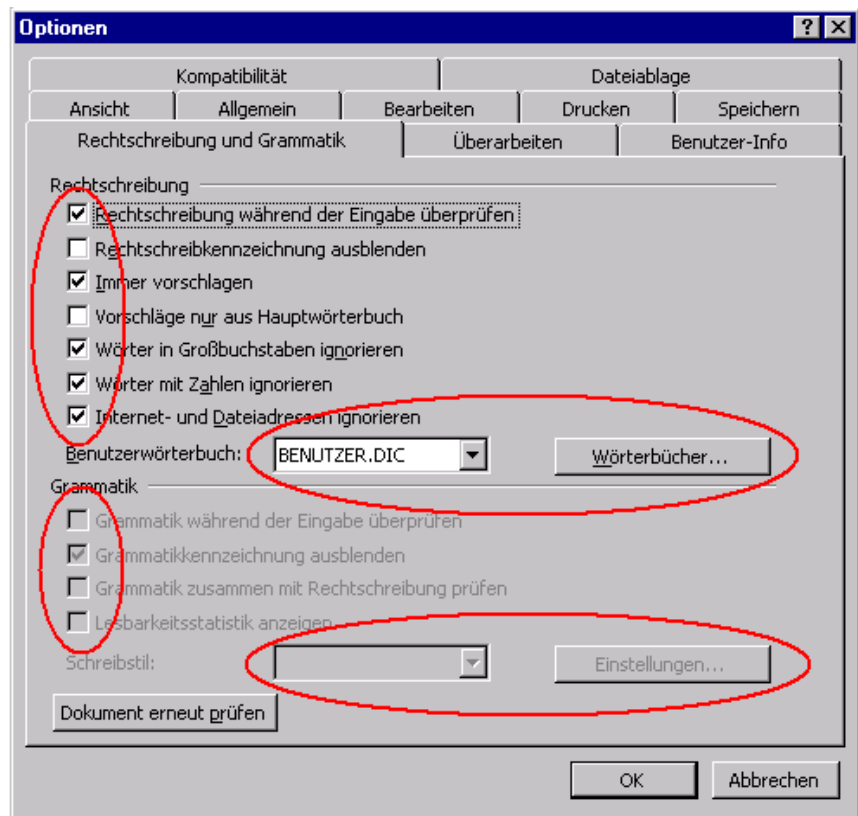
Die Abstände helfen mit, dass zusammengehörige Informationen als solche erkannt werden und der Bildschirm als ganzes Harmonie ausstrahlt. Sowohl die horizontalen als auch die vertikalen Abstände sollen innerhalb einer Applikation standardisiert sein. Einige generelle Regeln sind:

- Abstände zum Fensterrand sollten immer gleich sein
- Die Breiten von Schaltflächen sollten immer gleich sein.
- Abstand zwischen Gruppenfeldern und Führungstext (Titel) sollte innerhalb eines Fensters immer gleich sein (wenn möglich auch bei allen Fenstern). Die Gruppenfelder sollten eingerückt zum Titel sein.
- Wenn möglich sollten Listen- oder andere Gruppeneinträge der grösstmöglichen Eingabe entsprechen.

4.5 Gruppierung von Informationen

Die räumliche Anordnung und Gruppierung zeigt die Bedeutung der Felder und Schaltflächen auf. Man spricht von „intuitiven“ Oberflächen. Die Gruppierung wird über die Optimierung der Abstände zwischen den Bildschirm-elementen erreicht. Die Benutzer sehen sofort, welche Felder zusammengehören.

Wichtig dabei ist auch die fachliche Gestaltung. Fachlich zusammengehörige Elemente sollen auf dem Bildschirm zusammen dargestellt werden.



4.6 Farben

Farben erlauben das Hervorheben von Elementen und erleichtern das Erkennen der Bedeutung von Informationen. Farben sollen aber sparsam eingesetzt werden.

Farbkombinationen: Farbkombinationen, bei denen das Auge Probleme mit der Fokussierung hat:

- BLAU / ROT
- ROT / GRÜN
- GRÜN / BLAU

Vordergrundfarben: Sollten sich von Hintergrund abheben (z.B. grosser Kontrast).

Kontraste: Beispiele:



Reine Farben: „Muster“ vernebeln das Bild..



Psychologische Wirkung Informationen lassen sich verstärken. Die konventionelle Wirkung von Farben und Formen sollte berücksichtigt werden, z.B.:

ROT = Gefahr, Stop
GELB = Vorsicht
GRÜN = Sicherheit, keine Gefahr, Fluchtweg → Hilfe

4.7 Schrift

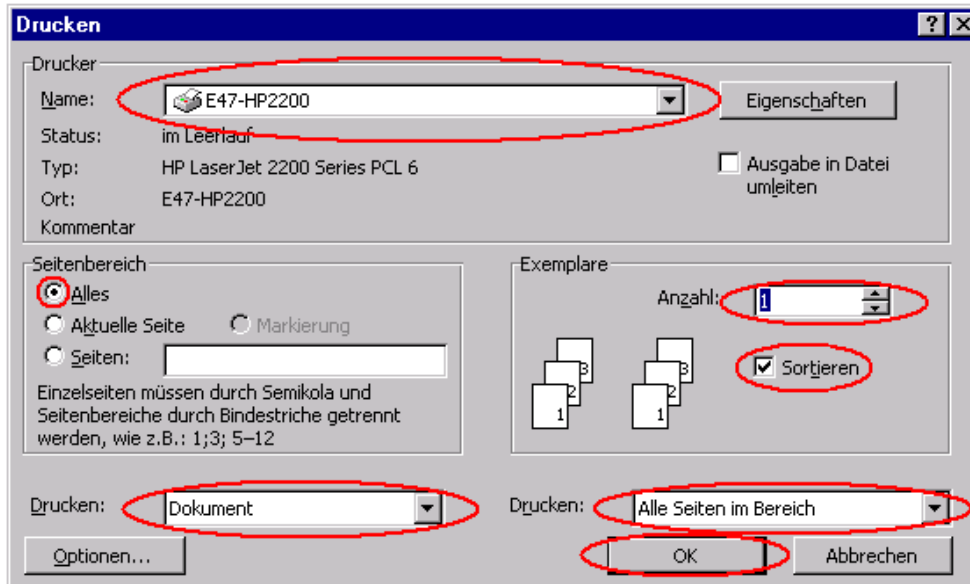
In der Regel werden in Programmen Standard-Schriftarten verwendet (z.B. MS Sans Serif). Einige Überlegungen zu den ergonomischen Aspekten:

- Man sollte auf kursive Schrift verzichten, da sie auf dem Bildschirmen mit geringer Auflösung gerne als „gezackte“ Linien erscheinen.
- Aus demselben Grund eignen sich Schriften mit Serifen auch nicht.
- GROSSSCHREIBUNG schadet der Lesbarkeit.
- Schriften mit Zeicheneindeutigkeit sind vorzuziehen: l (1) oder 1 (eins) oder I (I)
- Je nach Bildschirmgröße und –auflösung sollte man die Schriftgröße anpassen.

4.8 Dialoggestaltung

4.8.1 Defaults

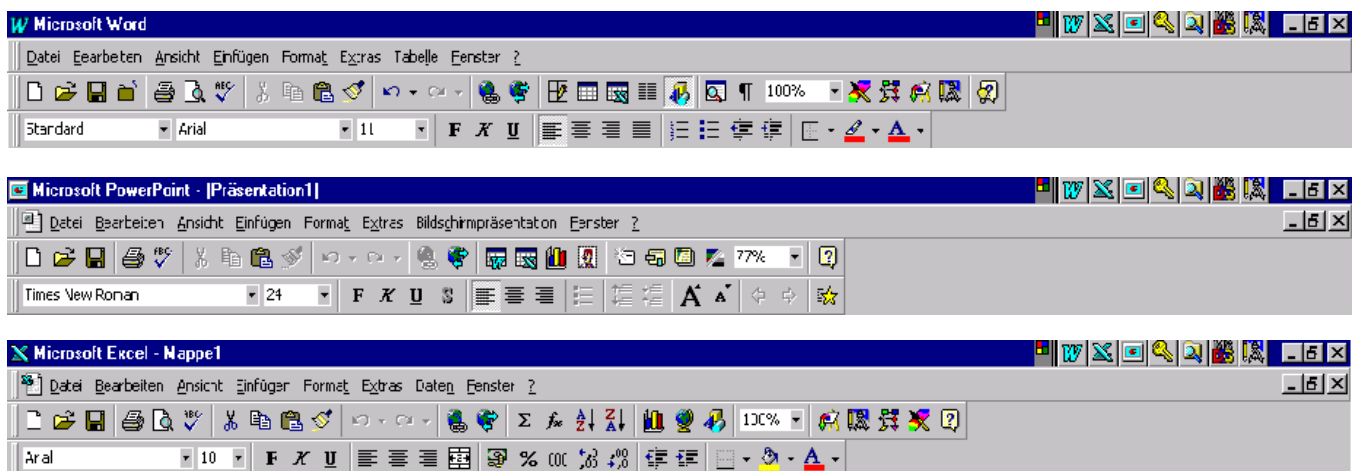
Defaults steht für Vorgaben, Standard-werte. Sie repräsentieren die Häufigkeit von Informationen und reduzieren Dialog-aufwand, indem sie die meistgewählte resp. Wahrscheinlichste Eingabe vorgeben.



4.9 Menüs

Menüs sollten verständlich – möglichst selbsterklärend, durch sinnvoll gewählte Titel oder Icons – sein und den Bedienungsaufwand minimal halten. Die Einträge der Menüleisten rufen entweder ein weiteres Menü (z.B. Dialogmenü) auf oder lösen eine Aktion aus, z.B. Aufruf einer anderen Funktion (drucken, andere Transaktionen).

- Häufig benutzte Einträge sollten möglichst weit oben hingesezt werden.
- Zu viele Ebenen schaden der Transparenz (nur 1-2 verschachtelte Menüs, nicht 10).
- Der Aufbau des Menüs innerhalb einer Applikation sollte immer ähnlich sein. Als Beispiel dienen die Office Produkte:



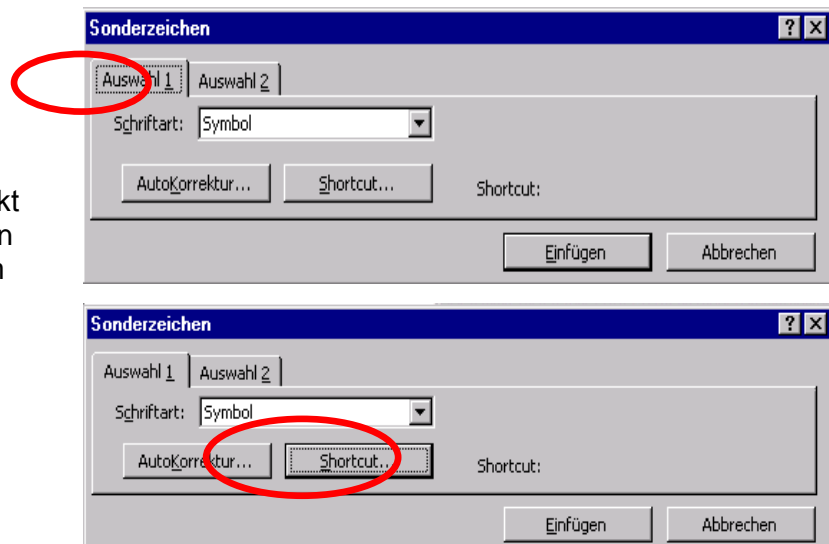
An diesem Beispiel sieht man, dass die Grundstruktur des Menüs, ob bei Word, Excel oder PowerPoint, stets beibehalten wurde.

4.10 Tabwege

Die Tabwege sollten so minimal wie möglich gehalten werden und immer einer einheitlichen Richtung folgen:

- Er beginnt links oben
- Er geht innerhalb einer einspaltigen Feldergruppe vertikal, resp. Horizontal bei nebeneinander liegenden Feldern, zum Ende dieser Gruppe.
- Er springt am Ende einer Gruppe zum obersten Eintrag der Gruppe rechts neben (resp. unterhalb) der bearbeiteten Gruppe.

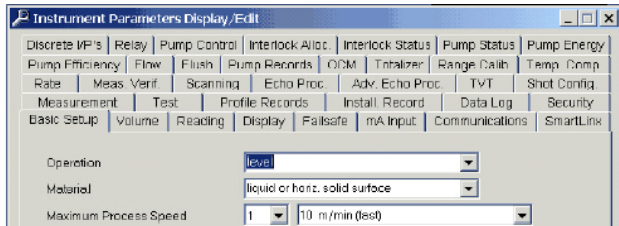
„Auswahl 1“ ist ausgewählt. Drückt man <<TAB>>, so wird der Button „Autokorrektur“ ausgewählt. Beim zweiten <<TAB>> wird „Shortcut“ selektiert, usw. immer von Links oben nach Rechts unten.



Wissenswertes...

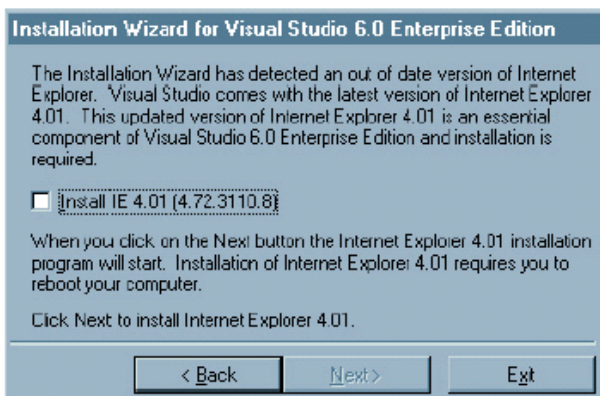
- Der Bildschirm (Masken, Menüs) sollte zumindest für die gesamte Applikation einheitlich sein (z.B. Fehlermeldungen immer auf der untersten Zeile, Hilfe immer via Taste F1, Kundennummer immer links oben)
- Um unerwünschtes Abschneiden der Eingabe zu verhindern, sollte die Länge des Dialogfeldes sichtbar sein.
- Menüpunkte und Steuertasten sollten beschrieben sein.
- Kommentare und Fehlermeldungen sollten in einer speziellen Zeile angezeigt werden.

4.11 Weniger komfortable GUI's



4.11.1 Dolphin Plus

Hier bei handelt es sich um ein Konfigurationspaket für industriellgenutzte Sensoren und Aktoren. Diese Beispiel sollte sich selbst erklären. Übersicht?!

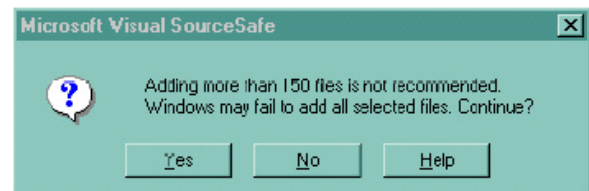


4.11.2 Microsoft's Visual Studio

Wie verbreitet man "unauffällig" oder dreist seine SW. Schauen sie sich das Fenster an! Der Benutzer freut sich, dass er die Wahl hat, ob er den IE installieren will oder nicht (Checkbox an oder abwählen). Der zweite Blick bringt die Ernüchterung mit sich, denn der Dialog lässt sich nur fortsetzen, wenn man den IE installiert. Man sollte eigentlich denken das die Qualität den Kunden überzeugen sollte und mit der ist es beim IE ja gar nicht so schlecht gestellt.

4.11.3 Microsoft's Visual Source Safe

Ein Programm zur Sicherung von Quellcode erzeugt bei dem Versuch eine weitere Datei abzuspeichern dargestellte Meldung. Was soll diese Meldung bezwecken? Will der Hersteller die Verantwortung auf den Benutzer abgeben, nur weil er nicht sorgfältig genug geplant hat. Das Programm ist nicht vertrauensenerweckend. Die Aufgabe dieser SW ist es Vertrauen zu schaffen, somit ist sie wohl eher ein Paradoxon.



5. Basis GUI-Elemente

5.1 Definition

Basis GUI-Elementen bilden die Grundbausteine des interaktiven Dialogs zwischen Software und Mensch. Hier werden wir auf einige davon genauer eingehen.

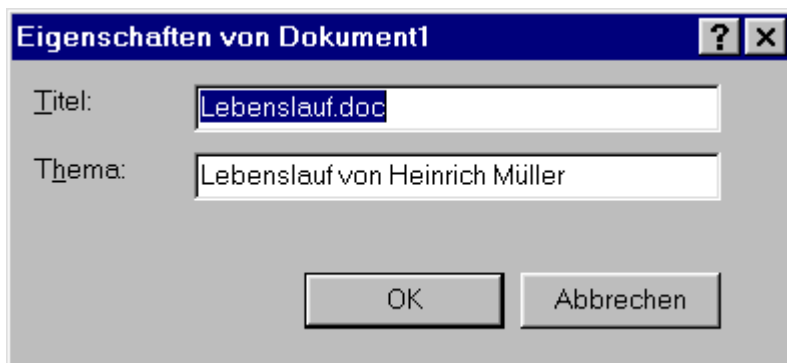
5.2 Eingabefeld

Eingabefelder sind ein- oder mehrzeilige Bereiche, in die der Benutzer Daten eingeben kann. Mehrzeilige Bereiche werden verwendet, wenn die genaue Länge der möglichen Eingaben nicht vorhersehbar ist, wie z.B. bei der Eingabe von Namen. Mehrzeilige Bereiche können auch mit Bildlaufleisten dargestellt werden.

5.2.1 Verwendung

Textuelle Eingabe von Information.

5.2.2 Beispiel



5.3 Drop-Down-Liste

5.3.1 Definition

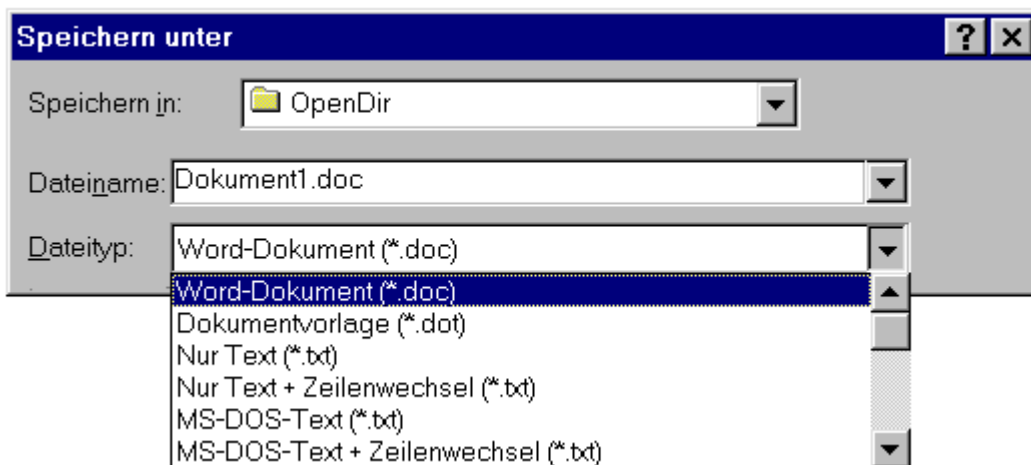
Eine Drop-Down-Liste ist eine temporär sichtbare Eifachauswahlliste. Der Benutzer fordert bei Bedarf die Auswahlliste an.

5.3.2 Verwendung

Eine Drop-Down-Liste sollte verwendet werden, wenn:

- die Auswahlliste als Eingabehilfe nicht immer sichtbar sein soll,
- alle Auswahleinträge dem System bekannt sind,
- die Anzahl der Wahlmöglichkeiten so groß ist, daß sie im Arbeitsbereich nicht übersichtlich abgebildet werden können.

Beispiel



5.4 Checkbox

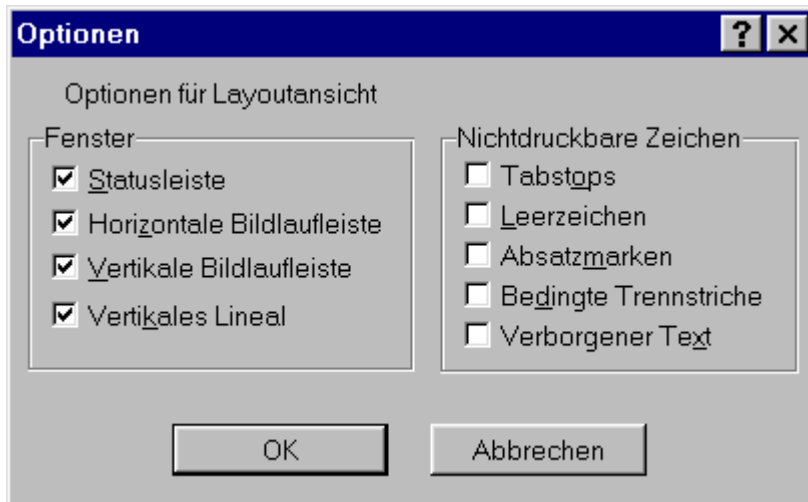
5.4.1 Definition

Das Kontrollfeld bietet die Möglichkeit Optionen an- bzw. auszuschalten. Das Kontrollfeld kann allein oder in einer Gruppe stehen.

5.4.2 Verwendung

- Mit Kontrollfeld werden Einstellungen für Objekte oder Aktionen vorgenommen. Sie lösen nie Aktionen aus.
- Kontrollfelder können in allen Fensterarten vorkommen.

5.4.3 Beispiel



5.5 Aktionstasten – Push-Button

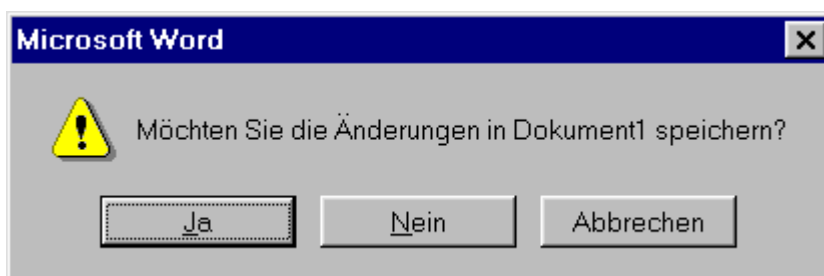
5.5.1 Definition

Aktionstasten sind sensitive Bereiche, die der Benutzer mit der Maus oder der Tastatur aktivieren kann. Dadurch wird die Aktion, auf die die Tastenbeschriftung hinweist, direkt ausgelöst.

5.5.2 Verwendung

- Durch Aktionstasten wird dem Benutzer die direkte Selektion der wichtigsten Aktionen ermöglicht.
- Im Hauptfenster können sie gelegentlich eingesetzt werden, um für häufig benutzte Funktionen, den Weg über die Menüleiste zu umgehen. In diesem Fall sollten aber nur Funktionen auf Aktionstasten gelegt werden, die auch in der Menüleiste vorkommen.
- Aktionstasten ersetzen z.T. die Funktionstastenzeile herkömmlicher alphanumerischer Benutzeroberflächen.

5.5.3 Beispiel



6. Dialoggestaltung

Ein Dialog ermöglicht die Interaktion zwischen einem Benutzer und einem Dialogsystem, um ein bestimmtes Ziel zu erreichen.

Aus Sicht der Benutzer lassen sich Primär- und Sekundärdialoge unterscheiden. Die Arbeitsschritte, die unmittelbar zur Erfüllung der Hauptaufgabe dienen, bilden den Primärdialog. Er ist dann beendet, wenn die zu bearbeitende Aufgabe erfüllt ist. Benötigt der Benutzer zusätzliche Informationen oder muss er Teilaufgaben bearbeiten, dann werden diese Hilfsdienste durch Sekundärdialoge erledigt. Ist der Sekundärdialog beendet, wird der Primärdialog fortgesetzt.

6.1 Beispiel

Bearbeitet man in einem Tabellenkalkulationsprogramm ein Dokument, so wird ein Primärdialog ausgeführt. Wenn man das Dokument drucken möchte, wird der Sekundärdialog „Drucken“ mit den jeweiligen Einstellungen ausgeführt. Erst wenn dieser Dialog beendet wird, kann mit der Bearbeitung des ursprünglichen Dokumentes fortgefahren werden.

Primär- und Sekundär-Dialoge müssen häufig synchronisiert werden.
Ein GUI-System muss daher die folgenden Anforderungen erfüllen:

- Sicherstellung der korrekten Abfolge der Arbeitsschritte
- Schließen aller Fenster beim normalen oder abnormalen Ende einer Anwendung
- Synchronisierbarkeit von Primär- und Sekundär-Dialogen
- Funktionen, die auf ein Fenster wirken, wirken auch auf alle hierarchisch untergeordneten Fenster

6.2 Modal / Nicht-Modal

Aus technischer Sicht lassen sich folgende Dialogmodi unterscheiden:

- Ein modaler Dialog (modal dialog) muss beendet sein, bevor eine andere Aufgabe der Anwendung durchgeführt werden kann.
- Ein nicht-modaler Dialog (modeless dialog) ermöglicht es dem Benutzer, den aktuellen Dialog zu unterbrechen, d. h. andere Aktionen durchzuführen, während das ursprüngliche Fenster aktiv bleibt. Bei dieser Dialogform wird also kein bestimmter Arbeitsmodus vorgeschrieben. Das Ziel der Dialoggestaltung sollte es sein, möglichst viele nicht-modale Dialoge zu verwenden, da dadurch die Handlungsflexibilität optimiert wird. In bestimmten Situationen sollte die Flexibilität eingeschränkt werden. Wenn z. B. ein Fehler auftaucht, kann erst nach dessen Behebung weitergearbeitet werden.

6.2.1 Beispiel

Wenn man in einer Textverarbeitung während der Bearbeitung eines Dokumentes den Dialog „Suchen“ startet, so kann man mit der Texterstellung fortfahren, ohne den Dialog zu beenden (nicht-modaler Dialog). Dagegen handelt es sich bei der „Seitenansicht“ um einen modalen Dialog. Erst wenn dieser Dialog beendet wurde, kann mit der Texterstellung fortgefahren werden.

6.3 MDI (Multiple Document Interface)

Wir differenzieren zwischen SDI- und MDI-Anwendungen. Eine SDI-Anwendung (Single Document Interface) ermöglicht es dem Benutzer, zu einem Zeitpunkt nur ein einziges Dokument zu öffnen und zu bearbeiten (z. B. WordPad). Bei einer MDI-Anwendung können zu einem Zeitpunkt beliebig viele Dokumente geöffnet sein (z. B. Microsoft Word/Excel).

7. Design Patterns

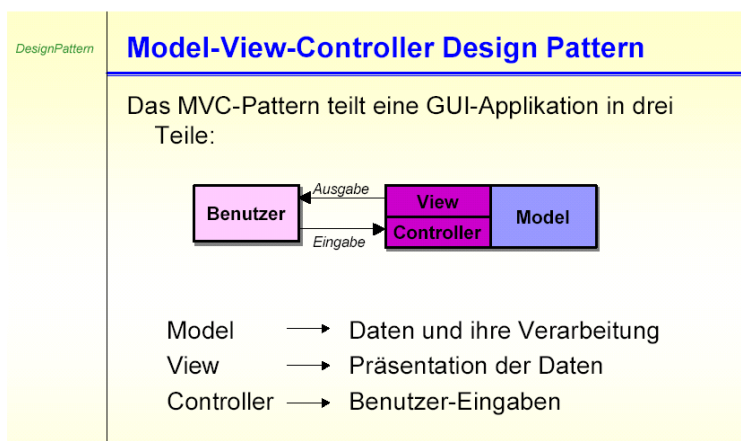
Design Patterns sind Programmiersprachenunabhängige Modulbeschreibungen, welche für die Definition von viel verwendeten Klassen benutzt werden. Aufgabe eines Design Pattern ist es, dem Betrachter einen vereinfachten Überblick der Objekte und Funktionen zu vermitteln. Da die Darstellung meist unabhängig von der Programmiersprache geschieht, soll das Modell später in jede Applikation integriert werden können. Design Patterns werden hauptsächlich für Objektorientierte Sprachen verwendet, da diese die optimalen Voraussetzungen betreffend Vererbung zwischen Objekten und der Implementation von Interfaces mitbringen.

Ein Pattern beschreibt ein wiederkehrendes Problem und die grundsätzliche Lösung dieses Problems. Diese Lösung ist 'millionenfach', wiederverwendbar ohne dass sich zwei Lösungen gleichen.

Hinweis: Design Patterns haben generell nichts mit GUI-Design zu tun. Es gibt jedoch einige Vertreter von Design Patterns, welche sich intensiv mit der GUI-Programmierung beschäftigen z.B MVC.

7.1 MVC Design-Pattern

Das MVC-Design beschreibt die Trennung zwischen User-Interface und den Daten sowie deren Verarbeitung. Es ist, wie die Abkürzung erraten lässt, in drei Teile aufgeteilt.

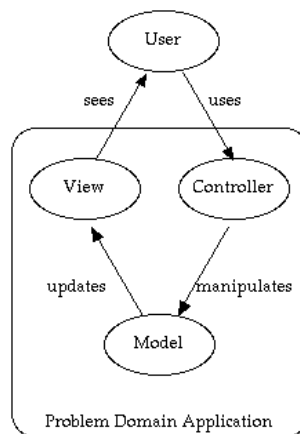


Das MVC Pattern hat zwei Vorteile:

- Darstellung der gleichen Daten in mehreren Views (Fenstern); wobei alle Views geändert werden, sollten sich die Daten ändern
- Wiederverwendung eines Views durch Schreiben eines neuen Models für neue Daten. Beispiel: ListView mit ListModel, ArrayModel, EmployeeModel etc.

Wieso das MVC-Pattern benutzen?

- MVC macht es einfach verschiedene Displays mit der selben Information zu haben
- Zum Beispiel eine Graphik und eine Tabelle, haben die Möglichkeit, die selben Daten zu zeigen und zu editieren
- Grundsätzlich bringt es bessere Kontrolle über das User-Interface und sein Verhalten mit sich.



Eines der besten Beispiele für MVC ist das pluggable “Look and Feel” in Swing. Hier kann ein Wechsel des Look and Feels das Aussehen der gesamten Applikation ändern.

8. Übungen

8.1

Übung 1:

Folgende beiden GUIs sollen verglichen werden. Welches ist benutzerfreundlicher? Begründe deine Entscheidung.

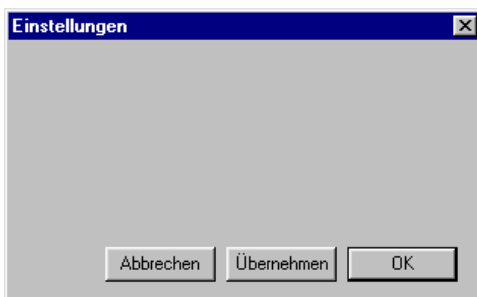


Abb. 1



Abb. 2

Übung 2:

Erkläre den Unterschied zwischen Modal und Nicht-Modal.

Übung 3:

Beschreibe die Ziele der Software Ergonomie.

Übung 4:

Beschreiben sie je einen Vor- und Nachteil von Design – Patterns

Übung 5:

Menüs sollten verständlich – möglichst selbsterklärend und sinnvoll beschrieben werden. Welche Eigenschaften müssen im Zusammenhang mit Verschachtelungen beachtet werden?

Übung 6:

An welcher Position werden die am häufigsten verwendeten Menüpositionen gespeichert, oben oder unten?