

Patterns für User Interface Design

- Bis jetzt: Techniken zum Entwurf einer sauberen SW-Architektur
- Aber: Wie baut man „gute“ Benutzungsoberflächen?
- Im folgenden: Einige Patterns aus
 - J. Tidwell: Designing Interfaces, O'Reilly, 2005.
- Weitere Literatur:
 - The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques von [Wilbert O. Galitz](#), 3rd Edition, 2007



Elemente der Präsentationsschicht

- Eine Bedienoberfläche muss folgende Elemente haben:
 - Eingabe von Information
 - Ausgabe von Information
 - Führung/Hilfe für die Benutzer
 - Anpassung an die Bedürfnisse der Benutzer
- Wichtige Fragen, die beim Entwurf zu stellen sind:
 - Wie können Benutzer ihre Aufgaben am effektivsten erfüllen?
 - Welche Komponenten sind am besten für die Interaktion geeignet?
 - Wie kann man Information am besten visuell repräsentieren?
 - Wie kann der Benutzer möglichst flexibel handeln?
 - Was können Benutzer alles falsch machen?

Unterschiedliche User-Profile

- Der User als Experte
 - Beispiele: Photoshop, Eclipse, Administrationswerkzeuge
 - Komplexe Funktionalität, wenig Anleitung der Nutzer
 - Optimierungen für effiziente Bedienung
 - oft dokumentenzentriert oder listenorientiert
- Unerfahrene User
 - Beispiele: Ticketverkauf, Terminal im Touristenzentrum, eShops, Geldautomat
 - Wenig Funktionalität, Anleitung der Nutzer
 - Einfache Oberflächen
- Die meisten Anwendungen liegen in der Mitte!

Patterns für User-Verhalten

- Safe Exploration
 - „Let me explore without getting lost or getting into trouble.“
 - Unterstützung des Users, das System auszuprobieren
 - Beispiele: Undo, Back Button, Warnung vor entstehenden Kosten
- Instant Gratification
 - „I want to accomplish something now, not later.“
 - User soll Aufgaben möglichst ohne Umwege erledigen können
 - Beispiele: Funktionalität nicht hinter Einführungen/Registrierungen/Werbung verstecken

Patterns für User-Verhalten /2

- Satisficing (Satisfying + Sufficing)
 - „This is good enough. I don't want to spend more time learning to do it better.“
 - User akzeptieren das „gut genug“ anstatt „das Beste“, wenn das „gut genug“ leicht zu bekommen ist
 - Beispiele: Übersichtliche Menüs, Beschriftungen, Pfade
- Changes in midstream
 - „I changed my mind about what I was doing.“
 - User ändern manchmal ihre Ziele
 - Beispiele: Wieder-Eintreten in Vorgänge, gute Wahlmöglichkeit von Funktionalität, Speichern von eingegebenen Daten

Patterns für User-Verhalten /3

- Deferred Choices
 - „I don't want to answer that now; just let me finish.“
 - Beispiele: Trennung von notwendigen und optionalen Informationen, Ausblenden von weniger wichtigen Fragen/Informationen, default-Werte
- Incremental Construction
 - „Let me change this. That doesn't look right; let me change it again. That's better.“
 - Unterstützung von inkrementellen Prozessen
 - Beispiele: gute Previews, möglichst kurze „Compile“-Zeiten

Patterns für User-Verhalten /4

- Habituation
 - „That gesture works everywhere else; why doesn't it work here, too?“
 - Beispiel: Ctrl-S für Speichern
 - Wichtig: Konsistenz der Bedienung innerhalb einer Anwendung
 - Problem: Bestätigung-Dialoge werden oft nicht gelesen („Wollen Sie die Datei XY löschen?“)
- Spatial Memory
 - „I swear that button was here a minute ago. Where did it go?“
 - User merkt sich oft mehr den Ort als die Beschriftung (z.B. von Dokumenten)
 - Vorsichtiger Umgang mit automatischer Umgruppierung, Schaffen von Bereichen, in denen User Informationen selbst anordnen können

Patterns für User-Verhalten /5

- Prospective Memory
 - „I'm putting this here to remind myself to deal with it later.“
 - Unterstützung von Notizzetteln, ToDo-Listen, geöffneten Fenstern, Bookmarks etc.
- Streamlined Repetition
 - „I have to repeat this how many times?“
 - Unterstützung von wiederkehrenden Aufgaben
 - Beispiele: Find&Replace, Definition von Macros

Patterns für User-Verhalten /6

- Keyboard Only
 - „Please don't make me use the mouse.“
 - Es gibt viele Nutzer, die keine Maus bedienen wollen/können
 - Unterstützung der Möglichkeit, Anwendungen ohne Maus zu bedienen
- Other People's Advice
 - „What did everyone else say about this?“
 - Information über andere Meinungen, Empfehlungen
 - Beispiele: User Bewertungen, Empfehlungen (was haben andere Nutzer gemacht?)

Organisation von Anwendungen

- Listen von Objekten
 - z.B. E-Mails, Fotos, Dokumente, Lieder
 - Linear (sortiert), 2D-Tabellen, Hierarchien, Anordnung im Raum (z.B. Karten, Charts, Desktop)
- Listen von Aktionen/Aufgaben
 - Beispiel: Steuererklärung
- Listen von Kategorien
 - Beispiel: Lieder (Künstler, Album, Stil)
- Listen von Werkzeugen
 - Beispiel: Handy (Rechner, Mail, Spiele etc.)

User Input

• Auswahl zwischen 2 Alternativen

– Checkbox

- +: einfach, platzsparend
- -: die Bedeutung von „Nicht-Anklicken“ kann unklar sein



– Zwei Radio Buttons

- +: beide Alternativen sind explizit formuliert
- -: braucht mehr Platz



– Two-choice dropdown list

- +: wie oben, weniger Platzverbrauch, leicht erweiterbar auf mehrere Alternativen
- -: nur eine Alternative ist sichtbar, schwieriger zu bedienen



Auswahl zwischen N Alternativen (kleines N) / 1

• N Radio buttons

- +: Jedes ausgewählte Item ist jederzeit sichtbar
- -: hoher Platzverbrauch



• N-item dropdown list

- +: wenig Platzverbrauch
- -: nur eine Alternative sichtbar, außer das Menu ist geöffnet, schwierigere Bedienung



• N-item set of mutually exclusive iconic toggle buttons

- +: wenig Platzverbrauch, die Auswahl ist sichtbar
- -: evtl. kryptisch (Tooltip erforderlich), evtl. unklar, dass nur ein Item ausgewählt werden kann



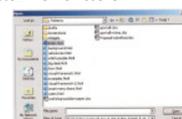
Auswahl zwischen N Alternativen (kleines N) /2

- Menu of N radio-button menu items
 - +: wenig Platzverbrauch (Teil von menu bars oder pop-up menus), Auswahl sichtbar
 - -: Popup Menu evtl. schwierig zu finden, schwer zu bedienen
- Spinner
 - +: wenig Platzverbrauch
 - -: nur eine Alternative sichtbar, schwierigere Bedienung



Auswahl zwischen N Alternativen (großes N)

- N-item dropdown list, evtl. mit Scrolling
 - +: wenig Platzverbrauch
 - -: nur eine Alternative sichtbar, außer das Menu ist geöffnet, schwierigere Bedienung
- Cascaded List
 - +: mehrere gewählte Items sind sichtbar, in vielen Fällen wird das Finden von Information erleichtert
 - -: evtl. unklar für unerfahrene Benutzer, hoher Platzverbrauch, schwierigere Bedienung
- Browser
 - +: Unterstützt die Suche verfügbarer Information
 - -: evtl. unklar für unerfahrene Benutzer, schwierigeres Design, erfordert separates Fenster



User-erstellte Listen /1

- Liste oder Tabelle mit „Add“ oder „New“ Button

- +: Buttons sind sichtbar und verständlich
- -: hoher Platzverbrauch, „visual clutter“



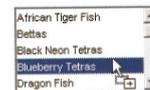
- Letzter Eintrag in der Liste/Tabelle kann frei definiert werden

- +: weniger Platzverbrauch, Editieren direkt in der Tabelle
- -: „add“-Aktion evtl. unklar



- Liste mit Drag-und-Drop erweiterbar

- +: visuell elegant und platzsparend, intuitiv
- -: „add“-Aktion evtl. unklar



Zahlen-Input /1

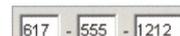
- Text-Feld

- Evtl. mit Forgiving Format (intelligente Korrektur von Fehlern)
- +: erlaubt viele Formate bzw. Datentypen
- -: Eingabeformat nicht explizit, sorgfältige Validierung des Inputs erforderlich



- Text-Feld mit strukturiertem Format

- +: Format explizit
- -: höherer Platzverbrauch, kein Abweichen vom Format möglich



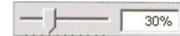
Zahlen-Input /2

- Spin-Box



- +: übersichtlich für ganze Zahlen
- -: evtl. ungeeignet für unerfahrenen Benutzer

- Slider (evtl. mit Text-Feld)



- +: klare Metapher, keine Eingabe außerhalb des angezeigten Bereichs möglich
- -: höherer Platzverbrauch