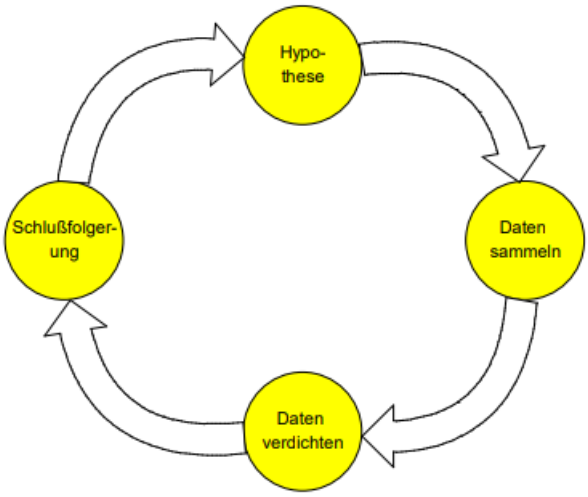
1 Einführung

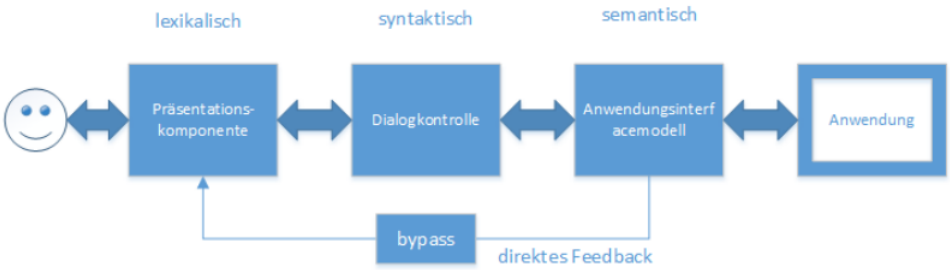
1.1 Benutzerschnittstellen

- **Human-Computer Interaction** untersucht
  - 1) Kontext von Computern
  - 2) Fähigkeiten des Menschen
  - 3) Entwicklungsprozess/Evaluation
  - 4) Architektur der Schnittstellen
- **1) Kontext von Computern:**
  - Arbeitsplatz
  - Schule
  - Auto
  - Haus
  - öffentliche Plätze
  - **Contextual Design**
    - muss Arbeitsweise des Benutzers unterstützen und erweitern
    - Partnerschaftliches Vorgehen und Partizipation der Benutzer
    - basiert auf Kohärenz (bzgl. Funktion, Struktur, Layout und Fluss im System)
    - Menschen sind Experten in dem, was sie tun, können es aber nicht artikulieren (Interviews vor Ort nötig)
    - basiert auf expliziter Repräsentation (Zeichnungen oder Modelle)
- **2) Fähigkeiten des Menschen:**
  - Menschliche Informationsverarbeitung
  - Kommunikation
  - Ergonomie (individuelles Vermögen und Grenzen)
    - **Behinderungen** (Körperbehinderungen, Sensorisch, Kognitiv)
    - Usability vs Accessibility
- **3) Evaluation**
  - Analyse der Probleme der Benutzer mit Software
  - Usersicht vs Expertensicht
  - Sammeln von Daten, beobachten, befragen, Kommentare
  - 5-7 Nutzer reichen für min. 80% der Entwurfsfehler

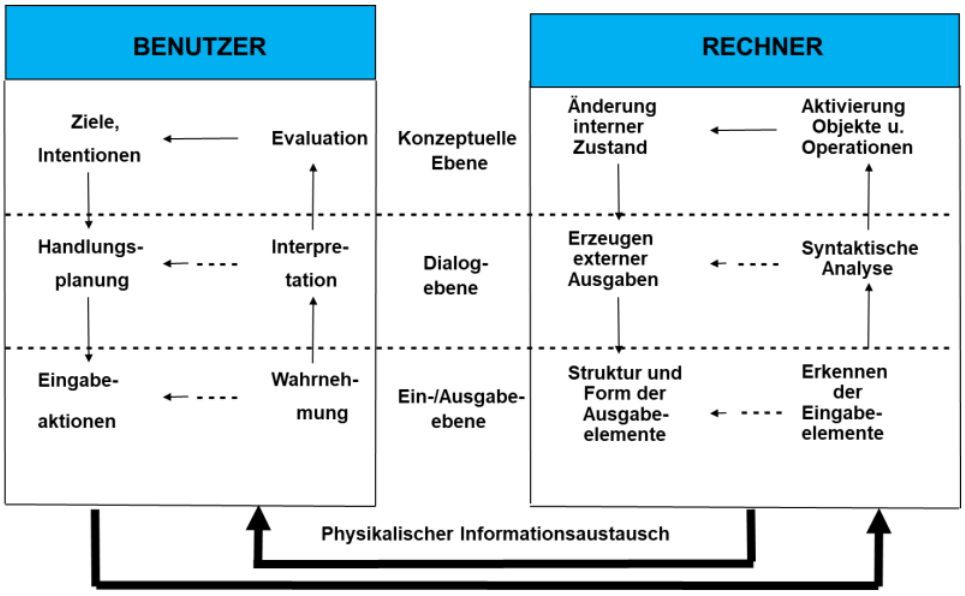


Heruntergeladen von Studydrive

- **4) Architektur**
  - Graphische Benutzungsoberfläche(GUI, Maus basiert, Desktop Computer)
  - Multimediale UI (Games 3D, Mobile mit Stimme und Stift)
  - Multimodale Benutzungsoberfläche
- **Seeheim Modell**
  - Anwendung kann ohne Eingabe Präsentation bzw. Feedback erzeugen (z.B. Fehlermeldungen, zeitabhängige Darstellungen)



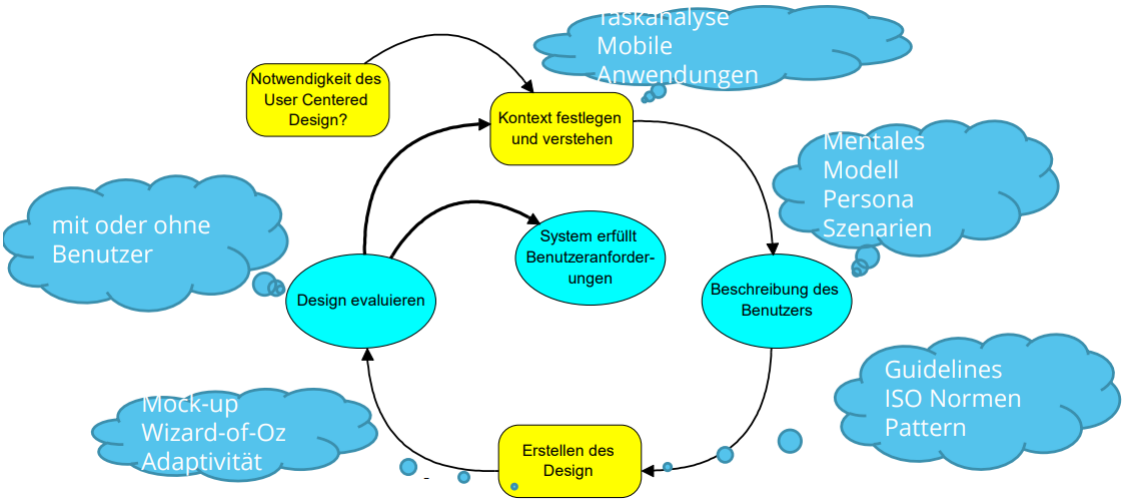
- **Nicht-visuelle Medien**
  - Hören (Sprach -wiedergabe, -synthese, Sprachassistent)
  - Sprechen (Spracherkennung, Diktierprogramme)
  - Fühlen (Brailleanzeigen)
  - Zeigen (mit/ohne Handschuh, immersive Systeme)
  - Ziehen (Kraftrückmeldung)
  - Kinästhesie (Erkennung Gesten)
  - Geruch (odor TV)
- **Schichtenmodell**



Heruntergeladen von Studydrive

1.2 Designprozesse

- **Ucer Centered Design** Lebenszyklus der Benutzeroberfläche



- Ziel ist Gebrauchstauglichkeit (**Usability**)
- Kriterien:
  - **Effektivität** (Ich komme zum Ziel)
  - **Effizienz** (Mit wie viel Aufwand komme ich zum Ziel?)
  - **Zufriedenstellung** (Wie angenehm war der Weg zum Ziel?)

1.3 Aufgabenanalyse

- Benutzer verfolgt Ziel und benötigt Rückmeldung
- Ziel: gewünschter Systemzustand
- Aufgabe: Folge von Aktivitäten, zur Zielerreichung
- Erhöhung Verständnis existierender Systeme und Arbeitsabläufe
- **hierarchische Aufgabenzerlegung**
  - Welche Aufgaben
  - Welche Reihenfolge
  - Welche Kenntnisse
- **Hierarchical Task Analyses (HTA)**
  - **SEQ**...sequenzielle Aufgaben
  - **PAR**...parallele Aufgaben
  - **ALT**...Alternativen

Heruntergeladen von Studydrive

  - 1) Bestimmung Benutzergruppen, Vertreter, Hauptaufgaben
  - 2) Vorbereitung und Durchführung Datensammlung (Ziele, Aktionen, Quellen)
  - 3) Datenanalyse, beachten der Stoppregeln (z.B. Wahrscheinlichkeit und Kosten eines Fehlers überschreiten Grenze)
  - 4) allgemeines Aufgabenmodell
  - 5) Abstimmung Modell mit Benutzer
  - 6) Iterationen planen
  - **Grenzen** der HTA
    - nicht sehr exakt
    - ungeeignet für Interaktionen unter Benutzern
    - Kontext wenig modelliert

1.4 Persona

- Beschreibung beispielhafter Benutzer
- Verwendung als Stellvertreter (Was würde Herr/Frau ... tun?)
- Sichtweisen auf Personas
  - **Ziel-gerichtet**
    - Vielzahl von Personas
    - in jedem Projekt neu entwickelt
  - **Rollen-basiert**
    - umfangreiche Fakten ausgewertet
    - mehrfach wiederverwendet
    - relevant für Marktforschung
  - **Einsatzorientiert**
    - Einsicht und Einbeziehung dargestellt
    - sozialer und psychologischer Hintergrund deutlich
  - **Fiktions-basiert**
    - Extreme Charaktere (z.B. behinderte Menschen)
- **Personas entwickeln**
  - Onsite Befragungen (auf Website)
  - Fokusgruppen (Workshops)
  - Tiefeninterviews (Einzelgespräche)
  - Nutzerbeobachtung vor Ort
  - Bildung von Clustern
  - Fotos und Namen verwenden

1.5 Szenarien

- Was werden Benutzer tun wollen? (lineare Abfolge)
- Schritt für Schritt Anleitung
  - Was können Sie sehen (z.B. Bildschirminhalt)
  - Was können Sie tun (z.B. mit der Maus)
  - Was denken Sie

Heruntergeladen von Studydrive

1.6 Mentale Modelle

- **mentales Model:**
  - subjektive Vorstellung eines Benutzers vom interaktiven System
- **konzeptionelles Model:**
  - von Designer
  - objektiv nachvollziehbar
  - Beschreibung von Aufgaben, Trennung von Objekten und Operationen
- **Analyse konzeptionelles Model:**
  - Welche Konzepte wird der Benutzer anwenden?
  - Gibt es andere vergleichbare Systeme?
  - Welche Art von Metapher (Spraydose als Bild in Paint)
  - Welche Interaktionstechniken sind relevant?
  - Woher können Fehler kommen?
- **Konzepte basieren auf**
  - Aktivitäten (Anweisungen erteilen, Unterhalten, Handhaben und navigieren, Erkunden und browsen)
  - Prozessen (ein oder mehrere Benutzer)
- Fehler im mentalen Modell durch fehlerhaftes Verständnis des Benutzers möglich

1.7 Regeln fürs Design

- **Entwurfsprinzipien:**
  - Kenntnis potenzielle Benutzer und ihre Aufgaben
  - Reduktion der kognitiven Belastung
  - Konsistenz
  - Abbruch und Rückgängig machen von Aktionen
  - Berücksichtigung von Fehlern
  - Adaptierbarkeit der Schnittstelle
- **Normen** z.B. ISO
  - Aufgabenangemessenheit
  - Selbstbeschreibungsfähigkeit
  - Erwartungskonformität
  - Steuerbarkeit
  - Robustheit gegen Benutzungsfehler
  - Erlernbarkeit
  - Benutzerbindung

Heruntergeladen von Studydrive

1.8 Evaluation durch Analyse der Handlungen

- **Modell des Gedächtnis**
  - Langzeit
  - visueller Bildspeicher
  - akustischer Speicher
  - kognitiver Prozessor
  - motorischer Prozessor
- **GOMS**
  - Evaluationsmethode um Effizienz zu verbessern
  - **Goals:** Zielzustand, den Benutzer erreichen will
  - **Operators:** Handlungen (Taste drücken)
  - **Methods:** Reihenfolge der Operatoren für Ziel
  - **Selection Rules:** Anwendungsauswahl bei mehrern Operatoren
  - **Keystroke Level Model (KLM):** Vergleich der Zeiten der Aufgaben
  - Nachteile:
    - nur Experten als Benutzer sinnvoll
    - Modelle werden zu groß
    - Fehler nicht leicht einzubeziehen

2 Evaluationsmethoden

2.1 Usability

- Ziele effektiv, effizient und zufriedenstellend erreichen
- **1) Analysephase**
  - Arbeits-, Prozess- und Systemanalyse
  - Erhebung der Nutzeranforderungen
- **2) Konzeptphase**
  - Arbeitsgestaltung und Prozessdefinition
  - Entscheidung über Systemfunktionalitäten
  - Konzepterstellung: High/Low-Fidelity Mock-ups
- **3) Entwicklungsphase**
  - Entwicklung von Prototypen
  - Systemintegration
- **4) Einführungsphase**
  - Piloteinsatz
  - Allgemeine Einführung
- **Partizipation**
  - Wer? (Nutzer oder Nutzervertreter)
  - Wie? (Passiv: Entwickler entscheiden, Aktive Mitwirkung: gemeinsam, Aktive Partizipation: Benutzer gestalten direkt)

Heruntergeladen von Studydrive

  - Wann? (dauern oder festgelegte Zeitpunkte)
  - Woran? (Definition, Gestaltung Funktionen...)
- **Evaluierung durch Benutzer:**
  - verschiedene Nutzergruppen
  - bestimmt Qualität der Software

2.2 Testverfahren

- Befragung (Interviews, Umfragen, Fragebogen, Eigenbericht, Aufzeichnungen...)
- Beobachtung (Thinking Aloud, heuristische Evaluation, Kontrolle, Messungen)
- Kreative Methoden (Fokusgruppen, Workshops)
- Testen der Software (z.B. GOMS)
- **Kontrolliertes Experiment**
  - Unabhängige/Abhängige Variablen bestimmen/messen
  - Within-group (mehrere Variablenwerte je Proband)
  - Between-group (ein Variablenwert je Gruppe)
  - statistische Auswertung und Schlussfolgerung

2.3 Evaluation von Zeigehandlungen (Performanztest)

- verschiedenste Zeigegeräte mit Dimensionen
- **Position** (Schieber, Tablett&Stift, 3D Joystick)
- **Bewegung** (Laufband, Maus, Mobiltelefon)
- **Kraft**
- **Temporale Eigenschaften**
  - Multimedia (mehrere Medien zur Darstellung benutzt)
  - Multimodalität (mehrere Eingabemethoden führen zum gleichen Ziel)
  - Intermodalität (Koordination mehrerer Wahrnehmungsmöglichkeiten)
- Empirische Tests **Fitts Law**
  - Zeit für das Erreichen eines Ziels ist abhängig von der Entfernung und der Größe des Ziels

$$\mathbf{MT = a + b * log_2(2A/W + c)}$$

    * MT...Bewegungszeit
    * a und b...empirisch bestimmte, geräteabhängige Konstanten
    * c...Konstante von 0, 0.5, 1
    * A...Entfernung der Bewegung (von Start zu Ziel)
    * W...Breite des Ziels
    * Term $log_2(2A/W + c)$ index of difficulty (ID) beschreibt Schwierigkeit der motorischen Steuerung
  - große nahe Zeile schneller erreichbar
  - ID erhöht sich um 1 je Verdoppelung der Amplitude und Halbierung der Breite
  - Verschiedene Modelle (Breite, Breite + Höhe, Breite * Höhe, ...)

Heruntergeladen von Studydrive

  - Standardabweichung (S) des Endpunkts einer Bewegung erhöht sich mit der Entfernung (D) des Endpunkts und verringert sich mit der Dauer (T) einer Bewegung, k...Konstante:
  - $\mathbf{S = k * (\frac{D}{T})}$
  - Optimierung:
    * Entfernung D verringern
    * Breite W erhöhen

- **Steering Law**
  - neben Zeigehandlungen auch andere Arbeiten von Bewegungen zu beobachten
  - $\mathbf{MT = a + b * \frac{A}{W}}$

2.4 Evaluationsmethoden

- **Heuristische Evaluation**
  - Grundlegende Richtlinien (Feedback an User, Hilfsmenü immer rechts, Play Button Dreieckssymbol...)
  - Evaluatoren prüfen Einhaltung der Richtlinien (Prioritätsstufen: kosmetisch, kleines, großes Problem, Katastrophe)
- Heuristiken nach **Nielsen und Molich**
  - Sichtbarkeit Systemstatus
  - Benutzerkontrolle und Freiheit
  - Ästhetisches Design
  - Hilfe und Dokumentation

2.5 Cognitive Walkthrough

- zielt auf unerfahrene Benutzer
- Aufgaben festgelegt mit Tutorial
- Gruppe von Usability Experten diskutiert:
  - Werden Benutzer versuchen gewünschten Effekt zu erzielen?
  - Werden Benutzer erkennen, dass korrekte Handlung ausgeführt werden kann?
  - Werden Nutzer erkennen, dass korrekte Handlung zum gewünschten Effekt führen wird?
  - Werden Benutzer Fortschritt erkennen, wenn sie die korrekte Handlung ausgeführt haben?

2.6 Prototypen

- **Problem** früher Phasen:
  - Benutzer können wenig mit abstrakten GUIs anfangen
  - funktionsfähige GUI erst spät in Entwicklung
- **Lösung** durch Prototypen:
  - Prototyp so früh wie mgl. testen lassen
  - schnelle und billige Realisierung (eingeschränkte Funktionalität, Verzicht auf optimale Effizienz...)

Heruntergeladen von Studydrive

2.7 Qualitative Methoden

- **Thinking Aloud Methode**
  - leicht aufzuzeichnen
  - wenn Proband aufhört, Fragen stellen (z.B. Was denken Sie gerade?)
- **Grounded Theory**
  - Theoriebildung durch qualitative Analyse verbaler Äußerungen zur Bildung von Kategorien
- **Eyetracking**
  - Performanzanalyse und qualitative Auswertung der Aufmerksamkeit
- **Fragebögen**
  - Offene Fragen (qualitative Analye)
  - Likert-Skalen (quantitative Analyse)
  - anfangs allgemeine Fragen (Alter)
  - keine doppelte Verneinung
  - Standardisierte Fragebögen liegen vor
  - **Task Load Index (TLX)**
    - Mental Demand (MD): mental anspruchsvoll?
    - Physical Demand (PD): körperlich anstrengend?
    - Temporal Demand (TD): Tempo der Aufgabe gehetzt?
    - Performance (OP): Aufgabe erfolgreich ausgeführt?
    - Effort (EF): hart, Leistungsniveau zu erreichen?
    - Frustration (FR): stressig, frustrierend?
    - Paare bilden (z.B. PD/MD) und wichtigeres auswählen (zählen und dadurch Gewichtung)
    - Likert Skalen für jeden Wert
    - Endwert Summe aller Punkte (vorher Likert Wert * Gewichtung)
    - niedriger Wert ist gut

3 Texteingabesysteme

- Sprache und Handschrift (fehleranfällig)
- Touch Typing (Tastatur)
- dynamische Selektion
- Shape Writing
- Gesten
- andere z.B. Blicksteuerung
- **Bewertungskonzepte**
  - Geschwindigkeit
  - Fehlerrate
  - Ergonomie
  - Lernbarkeit

Heruntergeladen von Studydrive

- Hardware Tastaturen (QWERTY, Neo, 12key)
- virtuelle Tastaturen (FITALY, OPTI)
- Gestern Alphabete (Unistrokes, Graffiti)
- Alternativen (Visual Panel, Chording Glove)

4 Fenstersysteme

4.1 Interaktion und Paradigmen

- **Interaktionsmodelle** helfen, die Handhabung von Systemen durch Benutzer zu verstehen
- **Aufgabenanalyse:** Feststellung des Problemraums mit den Dimensionen Domäne, Ziel, Intentionen und Aufgaben
- **Norman´s** Modell der Interaktion:
  - ausarbeiten des Ziels
  - bestimmen der Intention
  - festlegen der Aktionssequenz
  - ausführen der Aktion
  - wahrnehmen des Systemzustands
  - interpretieren des Systemzustands
  - evaluieren des Systemzustands in Bezug auf Ziele und Intentionen
  - **Ausführungslücke...**Unterschied zwischen formulierten Aktionen, um Ziel zu erreichen und Aktionen die vom System erlaubt sind
  - **Bewertungslücke...**Unterschied zwischen physischen Präsentation des Systemzustandes und Erwartungen des Benutzers
- **Interaktionstechniken**
  - Kommandozeile (telnet, ssh)
  - Menü (ATM)
  - natürliche Sprache
  - Frage/Antwort Dialog
  - Formulare und Tabellenkalkulation
  - WIMP (Window, Icon, Menu, Pointer)
  - Zeigen, Klicken
  - 3D Benutzeroberfläche
- modale Dialoge (erzwingen Abschluss) vs nichtmodale Dialoge (erlauben parallele Bearbeitung von Aufgaben)
- **Direkte Manipulation** liegt vor wenn gegeben:
  - Sichtbarkeit aller relevanten Objekte
  - unmittelbare Rückmeldung auf Aktionen
  - Rücknahmemöglichkeit ALLER Aktionen
  - syntaktische Korrektheit aller Aktionen
  - Direkte Manipulation ermöglicht Drag and Drop

Heruntergeladen von Studydrive

4.2 Komponenten eines GUI

- **Fenstersystem**
  - Eingabeverwaltung
  - Verwaltung der Ausgaben
  - Fenstermanagement
- Kriterien der **Architektur von GUIs**
  - Verfügbarkeit
  - einfache Programmierung
  - Betriebssystem verbergen
  - Graphik Engine (Raster vs Vektor)
  - Struktur/Komfort API
  - Prozesskommunikation (Clipboard, Drag and Drop)
  - Anpassbarkeit (je Benutzer/Land/Kultur/Sprachen/zeitlich)
  - Erweiterbarkeit
  - Parallele Verarbeitung von Aufgaben
- **Komponenten** eines Fenstersystems
  - Anwendungen
  - Toolkits
    - Sammlung an Dialogtechniken
    - ergänzende Programmeinheiten (Installations-/Deployment Tookits, Netzwerküberwachung, Fehlerdiagnose)
  - Windowmanager
    - Fensterverwalter
  - Ressourcenmanager
    - Fenster (owner, Zustand)
    - Ereignisse
    - Queue (Warteschlange der empfangenen Ereignisse)
    - Prozess
    - Multiplexing (Queues richtig zuordnen)
    - Schriftarteneinsatz
    - Grafikkontexte (durch Anwendung, Graphische Attribute)
    - Farbtabelle
    - Remote Zugriff (verteilte Systeme)
  - Graphik Engine
    - Eingabe
    - Operation
    - Ausgabe
  - Hardware
- **Betriebssystem**
  - Adressräume (Kernel, Anwendung/User)
  - Ressourcenverwalter bietet Dienste, Anwendung nutzt sie
- **Fensterverwalter (FM)**

Heruntergeladen von Studydrive

  - Verantwortlichkeit für Fokus und Verdeckung
  - logischer Bildschirm umfasst evt. mehrere Monitore
  - Fensterrahmen
  - Drag and Drop
  - Icons (Mauszeiger)
- **Abstraktionsgrad**
  - Programmierung (Funktions- oder Klassenbibliotheken, Verwendung in allgemeiner Programmiersprache)
  - Textuelle Spezifikation (Spezielle Beschreibungssprache)
  - Strukturiertes Editieren (Graphische Spezifikation (direkte Manipulation), Syntax-sensitiver Editor)
  - Automatische Generierung (Automatische Auswahl und Attributierung der Oberflächenobjekte, Generierung von Dialogen mit einheitlicher Syntax)
- **GUI-Werkzeuge**
  - GUI-Fenstersystem
  - 1) UI-Toolkit (Oberflächenbaukästen)
  - 2) UI Builder (Oberflächenbeschreibungssprache und Editoren)
  - 3) UIMS (Dialogbeschreibungssprache und Simulationskomponente)
  - 4) auto. gen. Werkzeuge
- **1) UI-Toolkit**
  - stellen Bausteine für graphische Dialogobjekte bereit
  - Flexibilität (beliebige Erweiterungen)
  - Hohe Einlernzeit (Programmierkenntnisse)
  - Prototyping schlecht unterstützt
  - mangelnde Portabilität
- **Application Frameworks**
  - Rahmenprogrammierwerkzeuge (objektorientiert)
  - enthalten Klassen für Dialogobjekte und generische Klassen für Entwicklung der Anwendung
  - bietet sozusagen vorgefertigten Code für z.B. ein Dropdown Menu oder APIs für ein einfacheren Backend Zugriff
  - Mächtig, aber komplex!
- **2) UI Builder**
  - einfache Entwicklung von plattformunabhängige Benutzeroberflächen
  - Trennung Oberfläche und Restprogramm
  - muss transformiert werden für konkrete Schnittstellen für eine Programmiersprache
- **3) UIMS**
  - Gestaltung, Implementierung Bedienoberflächen
  - komplette Trennung von Dialog- und Anwendungsteil
  - Oberflächenentwicklung nach ergonomischen Gesichtspunkten durchführbar
- **4) Automatisch generierende Werkzeuge**
  - Dialogbeschreibungen aus abstrakten Spezifikationen generiert
  - automatische Umsetzung von durch Regelmenge und vordefinierter Bausteine
  - automatische Einhaltung ergonomischer Richtlinien
  - automatische Auswahl von Dialogobjekten

Heruntergeladen von Studydrive

5 Formale Modelle und Zeit

- **formale Modelle**
  - Analyse Benutzeroberfläche auf bestimmte Eigenschaften
  - Anwenden von Modellen menschlicher Performanz (GOMS)
  - Automatische Kritik zum Bildschirmdesign
  - Beweis sicherheitskritischer Aspekte

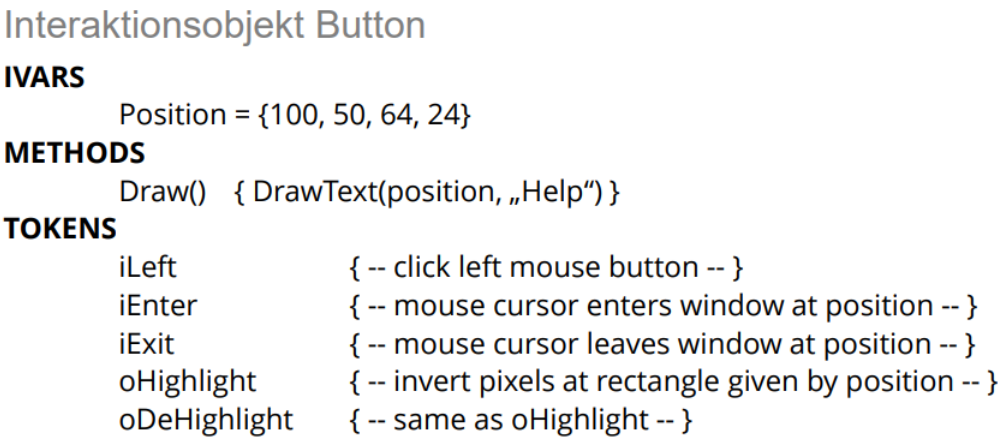
5.1 Übergangsdiagramme

- Zustandsdiagramm
- Bestimmung der kürzesten Wege (z.B. Floyd-Warshall Algorithmus)
- Starke Verbundenheit (von jedem Zustand kann jeder andere Zustand erreicht werden)
- Pfadlängen analysieren
- **Effizienzanalyse Fitts Law**
  - $\mathbf{MT = c + d * log((\frac{D}{W}) + 1)}$
  - W...Tasten mit Radius r
  - D...Entfernung zwischen den Tasten (Tastenmittelpunjt durch Koordinaten beschrieben)
- **Analyse durch Simulation**

5.2 Ereignisse in GUIs

- **Nebenläufigkeit**
  - mehrere Aufgaben laufen Parallel ab (z.B. Laden einer Datei im Hintergrund und Fortschrittsbalken im Vordergrund)
  - **Problem:** Teilautomaten stehen in loser Beziehung zueinander
  - Direkte Manipulation (Auswirkungen auf Objekte durch Zeigebewegung sofort sichtbar, vollständige Rückkopplung)
  - Sammlung von relativ kleinen Dialogen (Verhalten wie Koroutinen)
  - Koroutinen... Funktionen, die Ablauf unterbrechen und später wieder aufnehmen können
- **Kontrollfluss**
  - regelt zeitliche Abfolge der einzelnen Befehle
  - Interpreter benötigt
  - ordnet aktiven Übergangsdiagramm Eingabeereignis zu
  - **Interpreter:**
    - FROM: Liste von anderen Interaktionsobjekten, von der dieses Methoden, Variablen erbt
    - IVARS: Variablen mit Vorbelegung
    - METHODS: Interaktionsmethoden dieses Interaktionsobjekts
    - TOKENS: Ein- und Ausgabeelemente
    - SYNTAX: Übergangsdiagramm, legt Abfolge von Verarbeitung der TOKENS/Aktionen fest
    - SUBS: zusätzliche Übergangsdiagramme
    - STATES: zusätzliche Beschreibung von Zuständen

Heruntergeladen von Studydrive



5.3 Event Response Language (ERL)

- kann Nebenläufigkeit beschreiben
- Event Response System ERS: Tupel(F,$\sum$, R, S, f_f)
  - F... Bedingungsvariablen
  - $\sum$... Menge von Eingabeereignissen
  - R... Menge von Verarbeitungsregeln
  - S... Bedingungsvariablen (Teilmenge von F), die inital wahr sind
  - f_f... Bedingungsvariablen, die angibt, dass Eingabe akzeptiert ist
- Bsp.: $F_1 \rightarrow F_2$ (Aktion in F_2 wird "gefeuert" wenn Bedingung F_1 erfüllt ist, also wenn alle Bedingungsvariablen wahr sind)
- **Feuern einer Regel:**
  - Alle Zustände im Bedingungsteil einer Regel werden gelöscht
  - Alle Zustände in den Aktionsteilen einer Regel werden gesetzt (ohne Reihenfolge zu kennen)
  - Alle markierten Regeln werden als nicht markiert gekennzeichnet

5.4 Zeit und Interaktion

- Phasen der Interaktion
  - Eingabezeit
  - Antwortzeit
  - Ausgabezeit
  - Denkzeit
- Zeitintervall: Anfang t und Ende t+
- **Ereignis-Intervall Regel System (EIRS)** ist Tupel (F, I, R,Ω ,S ,f_f)
  - F... Bedingungsvariablen
  - I... Ereignisse und Intervalle für Verlauf
  - R... Regeln
  - Ω... Wurzelintervall

Heruntergeladen von Studydrive

  - S... Bedingungsvariablen (Teilmenge von F), die inital wahr sind
  - f_f... Bedingungsvariable, die "wahr" wird, und Ende vom System angibt
- **Temporale Modelle**
  - beschreiben zeitliche Strategie des Benutzers während der Interaktion
  - geben Bedingungen für zeitliche Beschränkungen während Interaktion an
  - beschreiben Nebenläufigkeit von Interaktionsschritten
- **Intervalldiagramm**
  - beschreibt durch temporale Intervalle die Lebensdauer eines Objekts oder die Ausführbarkeit von Operationen
- **Parallele Modelle**
  - Parallelität durch erneute Eingabe
  - Parallelität durch zwei Eingabegeräte
  - Parallelität durch zwei Ausgabegeräte
- **Ansprechbarkeit**
  - **Sofortheit** 0,1 s bis 0,2 s (Rückmeldung Tastenanschlag)
  - **Unmittelbarkeit** 0,5 s bis 1 s (positive Bestätigung des Eingangs von Eingabe)
  - **Kontinuität** 2 s bis 5 s (Verarbeitungsschritte nach 2 s, bei Fehler auch bis zu 5 s)
  - **Beachtung** 5 s bis 10 s (maximale Aufmerksamkeitsspanne zur Verfolgung einer Aufgabe)
- **Zeiteinteilung**
  - **Prospektive Angaben**: Wie lange wird es dauern? (vermeiden, dass Benutzer was anderes machen)
  - **Echtzeitangaben** Restzeit (bei nicht Nachvollziehbarkeit von Teilschritten)
  - **Retrospektive Angaben**
- **Managment Wahrnehmung**: vorzeitige Installation obwohl Benutzer noch auswählt oder sinnvolle Ablenkung während Installation
- **Managment Toleranz**: großzügige Planung (wenn 3 min dauert, 5 min angeben) oder Puffern bei Streaming

6 Adaptierung und Adaptivität

- **Adaptivität**... beschreibt die selbständige automatische Anpassung des Systems an Eingaben oder auch von Ausgaben
- **Adaptierbarkeit**... wenn durch Anpassungen Einstellungen des Programms verändert werden können
- **Bestandteile adaptives System**
  - Erwerbung von Informationen über den Benutzer des Systems
  - Repräsentierung dieser Information
  - Generierung von personalisierten Inhalten und angepassten Navigationsstrukturen
- **Sensorschicht:** Interaktion Benutzer mit Kontext
- **Semantikschicht:** Identität Benutzers, Beziehungen zu anderen Objekten

Heruntergeladen von Studydrive

- **Kontrollschicht:** Basissteuerung mittels Regeln
- **Ausführungsschicht:** Domänenabhängige Implementierung der Steuerung
- **Methodik**
  - Afferenz: Beobachtung und Sammlung von Informationen
  - Inferenz: Auswertung der gesammelten Informationen
  - Efferenz: Anpassung des Systems
- **Hypermedien:** nehmen Eigenschaften vom Benutzer auf und passen sich dementsprechend an
  - **Tutorielle Systeme:** Lernsystem, was sich den Antworten auf Fragen anpasst, klare Strukturierung und Lernsteuerung
- **adaptive Führung:** auszeichnen der relevanten Verweise einer Seite
- **adaptive Annotation:** Generierung von Hinweisen für Verweise
- **adaptive Empfehlung:** Zeile vom System bestimmt, nur relevante Verweise dargestellt
- **Stereotypisierte Benutzermodelle** für Vorhersagen
- Kombination von verschiedenen Benutzermodellen (Filmempfehlung für Gruppe von Zuschauern)
- **Kollaboratives Clustern**
  - "Mentor Nutzer" finden, dessen Bewertungen stark mit aktuellem Benutzer korrelieren
  - aggregierte Bewertung der Mentoren für ein Objekt gibt Vorhersage
- **Inhaltsbasiert vs Kollaborativ**

	Inhaltsbasiert	**Kollaborativ**
Vorteile	▪ Empfehlung unbewerteter Objekte möglich ▪ Unabhängig von der Benutzerzahl ▪ Außergewöhnliche Präferenzen werden berücksichtigt	▪ Unabhängig von den Objekten für die Empfehlung ▪ Unabhängig von früheren Empfehlungen
Nachteile	▪ Objektbeschreibung ist notwendig ▪ Mindestzahl von Bewertung von einem neuen Nutzer ist notwendig ▪ keine subjektiven Kriterien ▪ keine Berücksichtigung der Erkenntnisse andere Benutzer	▪ Kaltstart (neues Benutzer, neues Objekt unsicher ▪ Bei schwach besetzter Matrix -> niedrige Empfehlungsqualität ▪ Popularitätsausrichtung

- **Hybride Empfehlungssysteme**
  - Inhaltsbasierte in kollaborative Technik integrieren (Demographische Daten im Benutzerprofil erfassen)
  - Kollaborative in inhalts-basierte Technik integrieren (Gruppierung von inhalts-basierten Benutzerprofilen)

Heruntergeladen von Studydrive

7 Navigationssysteme

7.1 Navigation mit mobilen Endgeräten

- Preis sinkt, Leistung steigt
- Eingabe: Tastatur, Touch, Gesten, Sprache, Haktik
- Ausgabe: Bildschirm, Sprache, akustische Signale, Vibration
- Anwendungen müssen mit kleinem Ausgabebereich zurecht kommen
- **Ortung** (GPS, GALILEO, COMPASS)
  - 4 Satelliten senden: 3 x Ortung x, y, z (Überschneidungen) und 1 x Signallaufzeiten
  - NMEA-Nachrichtenformat (ASCII): Uhrzeit, Länge, Breite, Qualität, Anzahl Satelliten, Höhe über Meeresspiegel
- **Anwendung:** Verkehrs- Seefahrtsüberwachung, Landwirtschaft (Autonome Steuerung), touristische Anwendungen
- **Fahrzeugnavigation** (dynamischer Einbezug von Verkehrsinformationen)
- **Fußgängernavigation**
  - Mikronavigation ¡10m (Spur halten, Hindernisse)
  - Makronavigation ¿50m (Navigation nächster Wegpunkt)
  - Kognitive Verarbeitung (Wissenaufbbau erfolgt unterschiedlich und parallel)
  - Kognitive Karten (points of interests)
  - Art der Fortbewegung wichtig, genauere Positionsbestimmung (2-5m), zeitabhängige Faktoren (Baustellen vermeiden)
  - Berücksichtigung von Mobilitätseinschränkungen (Blinde, Rollstuhlfahrer, Senioren ...)
- **Proximität**
  - intimer Bereich 0,45m
  - persönlicher Bereich 1,2m
  - sozialer Bereich 1,6m
  - öffentlicher Bereich 7,6m

7.2 Kollaboratives Benutzermodell einer Gruppe durch Annotation

- **Multimodale Annotation**
  - semi-automatische Annotation geographischer Daten
  - Berechnung von Routen optimiert für verschiedene Benutzergruppen
  - Generierung von umfangreicheren und personalisierten Navigationsanweisungen auf Basis der Annotationen
- **Annotation**
  - persönliche Attributierung geographischer Objekte (Points of Interests)
  - z.B. Bewertung eines Weges durch Benutzer
- **Annotiertes Wegenetz**
  - Gewichtung der Kanten eines Navigationsgraphen (Durchschnittszeit, Bewertungen...)
- **Multicriteria Decision Making (MCDM)**

Heruntergeladen von Studydrive

- Normierung der Werte
- Berechnung über Zielerreichungsmatrix

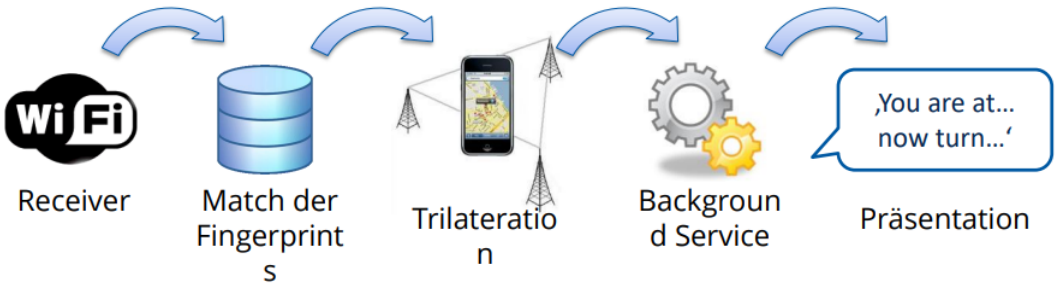
- **Entscheidungsregeln**
  - Dominanz (in allen Attributen überlegen oder min. gleichwertig)
  - Maximin (schlechtestes Attribut hat den besten Wert)
  - Maximax (maximaler Attributwert)
  - Konjunktiv (Mindestgrenzen von Attributen)
  - Disjunktiv (wie Konjunktiv + Alternativen zugelassen, die bei bestimmter Anzahl von Attributen Mindestwerte erreichen)
  - Lexikographisch (Konjunktiv in Reihenfolge der Wichtigkeit der Attribute bis nur 1 übrig bleibt)
  - additive Gewichtung (Berechnung mit Wichtigkeit und Summierung)

7.3 Personalisierung von elektronischen Fahrplananzeigen

- **UCD** nicht anwendbar, weil zu viele unterschiedliche Gruppen im öffentlichen Raum
- Personalisierung notwendig für verschiedene Nutzergruppen
- z.B. Bluetooth Bakes and Haltestellen

7.4 Navigation in Gebäuden

- Lokalisierung über Wifi:



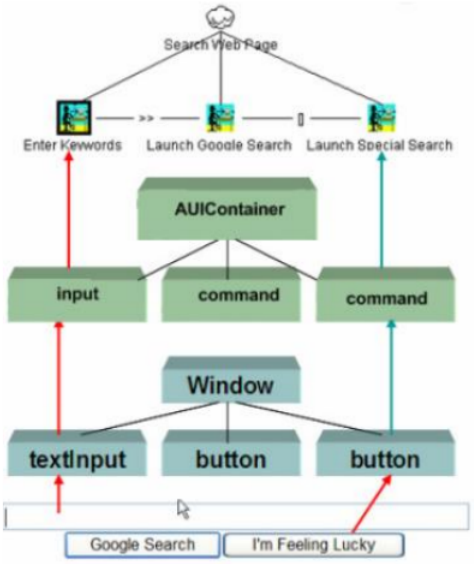
- Suche nach Übereinstimmung mit vorher bestimmten Fingerprints (Signalstärkenkarte)
- Korrekturen notwendig, da Schwankungen (Personen schirmen ab)

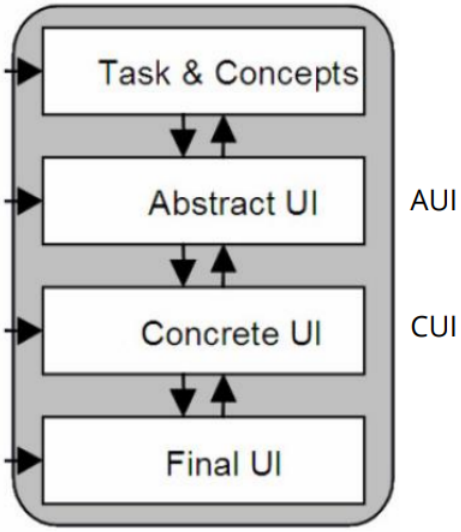
8 Entwicklungswerkzeuge von Benutzungsoberflächen

8.1 Modellbasierte Transformation

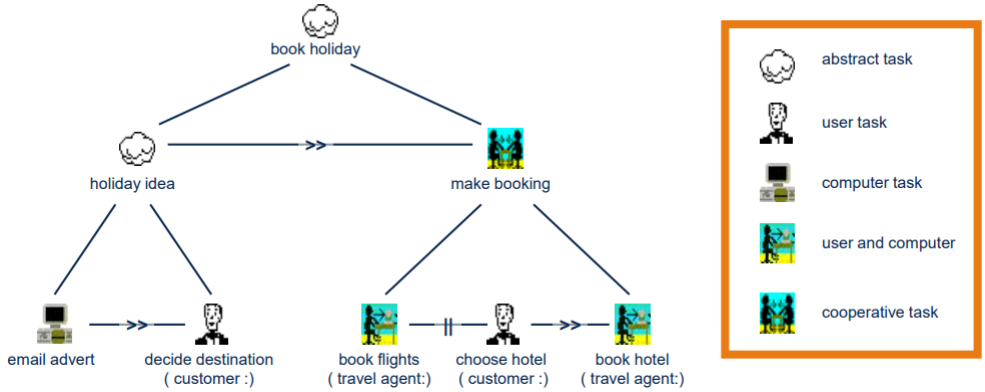
- **User Interface Description Language (UIDL)**... formale Sprache für Zwecke der MCI, um Benutzeroberfläche zu beschreiben

Heruntergeladen von Studydrive





- **Kontext der Benutzung**
  - eines interaktiven Systems ist dynamische strukturierter Informationsraum (Entitäten: Benutzer U, Plattform P, Umgebung E)
  - Kontext der Benutzung ist Tupel (U, P, E)
  - z.B. Multi-target UI (unterstützt mehre Benutzer, Plattformen und Umgebungen), Adaptable UI (kann angepasst werden)
  - Kontext: Produktion (Änderung Benutzungsoberfläche zur Laufzeit, je nach Standort und Benutzer)
- **ConcurTaskTrees (CTT)**



Heruntergeladen von Studydrive

- **Temporale Relationen**
  - T1 [] T2 - Auswahl
  - T1 ||| T2 - Überlappend
  - T1 |[]| T2 - Synchronisierung
  - T1 >> T2 - Ermöglichen
  - T1 []>> T2 – Ermöglichen und Weitergabe von Information
  - T1 [> T2 - Deaktivierung
  - T1* - Iteration
  - T1(n) - Finite Iteration
  - [T1] - Optional
  - T – Rekursion

- **Presentation Task Sets (PTS)**... entsprechen Aufgaben, die zur selben Zeit aktiv sind
- **UsiXML**
  - Beschreibung des Modells der Benutzungsoberfläche in einer gemeinsamen XML-Sprache (Modular erweiterbar)
  - beschreibt CUIs (Character), GUIs (Graphical), Auditory UIs, Multimodal UIs
  - Ziel: Geräteunabhängigkeit, Plattformunabhängigkeit, Wiederverwendung von Komponenten

9 Visuelles Programmieren

- Vereinfachung der Entwicklung
- Herausforderung besteht in der Abstimmung zwischen Spezifikation, Design, Umsetzung und eigentliches Ziel des Nutzers
- **Visuelle Programmiersprachen**
  - Er-Diagramme
  - Kontrollflussdiagramme

Heruntergeladen von Studydrive