

# Mensch-Computer-Interaktion

Professor: Weber, Wintersemester 21/22

## Inhaltsverzeichnis

1	Einführung .....	1
1.1	Human-Computer Interaction .....	1
1.1.1	Kontext von Computern .....	1
1.1.2	Fähigkeiten von Menschen: .....	1
1.1.3	Evaluation .....	2
1.1.4	Architektur.....	2
1.2	Aufgabenanalyse .....	2
1.3	Benutzer verstehen: Mentale Modelle .....	3
1.4	Evaluation durch Analyse der Handlungen .....	4
2	Evaluationsmethoden .....	4
2.1	Usability .....	4
2.2	Evaluation von Zeigehandlungen .....	5
2.3	Evaluationsmethoden .....	6
3	Texteingabesysteme.....	6
4	Fenstersysteme .....	7
4.1	Interaktion und Paradigmen .....	7
4.2	Komponenten einer GUI.....	7
5	Formale Modelle und Zeit .....	9
5.1	Übergangsdiagramme .....	9
5.2	Ereignisse in GUIs .....	9
5.3	Event Response Language (ERL).....	10
5.4	Zeit und Interaktion.....	10
6	Adaptierung und Adaptivität.....	12
7	Navigationssysteme.....	15
7.1	Navigation mit mobilen Endgeräten .....	15
7.2	Fußgängernavigation.....	16
7.3	Kollaboratives Benutzermodell einer Gruppe durch Annotation .....	16
7.4	Navigation in Gebäuden.....	18
8	Entwicklungswerkzeuge von Benutzeroberflächen .....	19
8.1	Modelbasierte Transformation .....	19
8.2	Visuelles Programmieren .....	21
8.3	Ausblick.....	21

# 1 Einführung

## 1.1 Human-Computer Interaction

**Definition:** HCI is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them."

### HCI untersucht:

- Kontext von Computern
- Fähigkeiten des Menschen
- Entwicklungsprozess
- Architektur von Schnittstellen

#### 1.1.1 Kontext von Computern

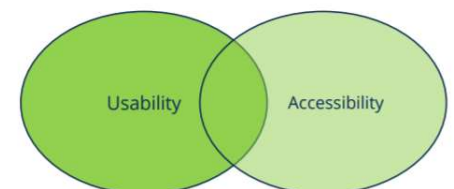
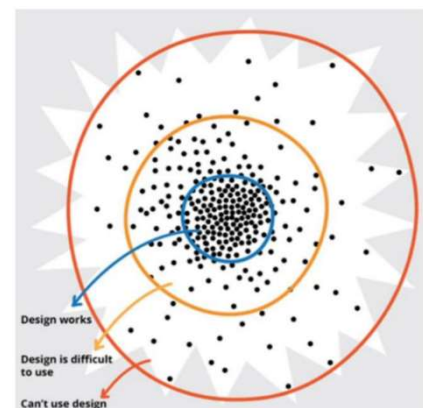
- Arbeitsplatz, Auto, Haushalt, Schule, öffentliche Plätze

### Contextual Design (nach Holzblatt):

- Systemdesign muss die Arbeitsweise des Benutzers unterstützen und erweitern
- Gutes Design erfordert partnerschaftliches Vorgehen und Partizipation der Benutzer
- Gutes Design basiert auf Kohärenz
- Menschen sind Experten, können das aber schlecht artikulieren
- Design basiert auf explizierter Repräsentation

#### 1.1.2 Fähigkeiten von Menschen:

- Menschliche Informationsverarbeitung
  - o Visual overload
  - o Motorische Fertigkeiten
  - o Akzeptanz
- Kommunikation
  - o Soziale Regeln
- Ergonomie
  - o Individuelles Vermögen und Grenzen
- Benutzer im weitesten Sinne:
  - o Kinder
  - o Ältere Benutzer
  - o Behinderte Menschen →
    - Körperbehinderungen
      - Rollstuhlfahrer, Gehstock etc.
      - Zittern, Spastiken
    - Sensorische Behinderungen
      - Blindheit, Sehbehinderungen
      - Gehörlosigkeit, Hörbehinderung
      - Taubblindheit
    - Kognitive Behinderungen
      - Aphasie (Verlust des Sprechvermögens)
      - Autismus
      - Demenz



- Dyslexie (Lese-Rechtschreib-Störung)

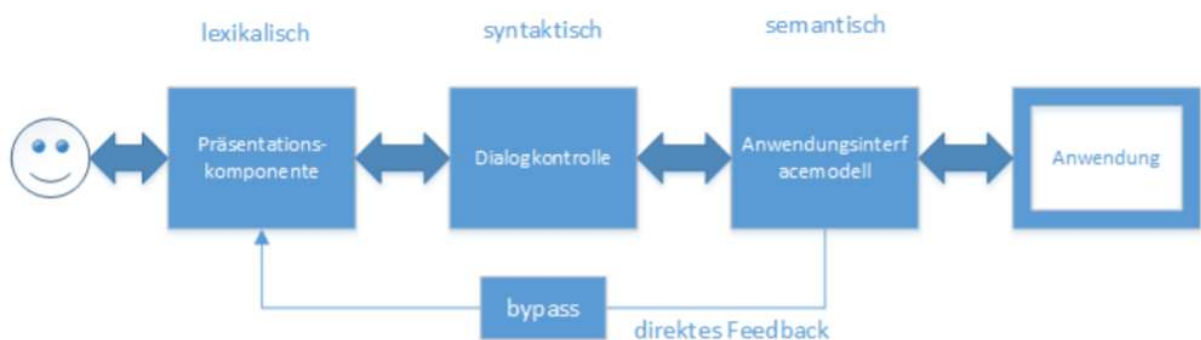
### 1.1.3 Evaluation

- Analysieren der Probleme der Benutzer mit Software
  - Sicht der User
  - Expertensicht
- 5-7 Nutzer reichen aus, um mind. 80% der Entwurfsfehler zu finden

### 1.1.4 Architektur

- Graphische Benutzeroberfläche (GUI)
- Multimedial UI
- Multimodale Benutzeroberflächen

#### Seeheim Modell zur Architektur des UI

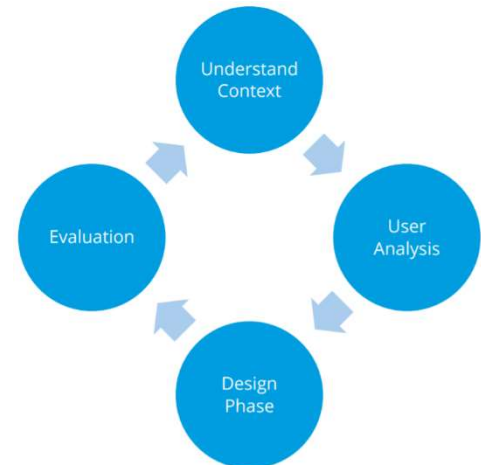


### Designprozesse

**User Centered Design:** ist ein iterativer Prozess, der den Benutzer in den Mittelpunkt stellt

Ziel des Prozesses:

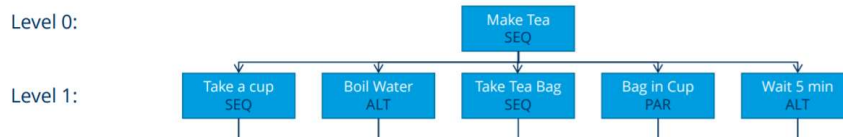
1. Gebrauchstauglichkeit
  - Effektivität
  - Effizienz
  - Zufriedenstellung
2. Akzeptanz



### 1.2 Aufgabenanalyse

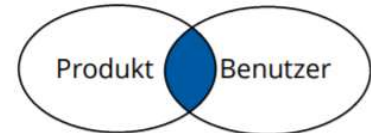
- Ziel (goal): Erwünschter Systemzustand
- Aufgabe (task): die Folge von Aktionen, um das Ziel zu erreichen
- Aufgaben, Ziele und Aktionen sind für unterschiedliche Benutzer verschieden
- Ähnlich der Modellierung im Softwareengineering, sind nicht alle Aktionen Teil der Interaktion
- ➔ Die Identifikation von Zielen der Benutzer bereitet die Evaluation vor
- ➔ Hierarchische Aufgabenanalyse benötigt Stoppregeln

## Hierarchical Task Analysis (HTA)



Grenzen der HTA:

- Nicht sehr exakt z.B.: „warten“
- Keine Interaktion der Benutzer untereinander
- **Kontext** wird nicht wirklich modelliert



## Benutzer verstehen: Personas

- Beschreibung eines beispielhaften Benutzers nicht notwendiger Weise einer realen Person
- Verwendung als Stellvertreter
- Details machen Personas relevant (Identität, Kontext, Präferenz)

## Szenarien

- Was werden Benutzer tun wollen?
- Lineare Abfolge notwendig – ohne Alternativen

### 1.3 Benutzer verstehen: Mentale Modelle

- ein mentales Modell ist eine subjektive Vorstellung eines Benutzers vom interaktiven System
  - ein konzeptionelles Modell wird vom Designer verfolgt und soll objektiv nachvollziehbar sein
    - o basiert auf Beschreibung von Aufgaben, d.h. Trennung von Objekten und Operationen
    - o Metaphern sind hilfreich, um konzeptionelle Modelle zu entwickeln
  - Fehler entstehen beim Erheben von Mentalen Modellen und sind bedingt durch:
    - o Fehlerhafte Wiedergabe/Verbalisierung durch den Benutzer
    - o Fehlerhafte Interpretation durch den Designer
- ➔ Benutzer kann man beschreiben durch: Persona (Persönlichkeit), Szenarien (Verhalten), mentales Modell (Denkweise)

## Regeln fürs Design

- Regelwerke sind Heuristiken
  - o Nichtbeweisbare Erfahrungen
  - o Barrierefreie-Informationstechnik-Verordnung hat indirekt Gesetzeskraft bekommen
- ISO 9241-110 benennt Gestaltungskriterien
  1. Aufgabenangemessenheit
  2. Selbstbeschreibungsfähigkeit
  3. Erwartungskonformität
  4. Steuerbarkeit
  5. Robustheit gegen Benutzungsfehler
  6. Erlernbarkeit
  7. Benutzerbindung

## 1.4 Evaluation durch Analyse der Handlungen

- GOMS ist eine Experten-basierte Evaluationsmethode
  - o Goals: Zielzustand, den der Benutzer erreichen will
  - o Operators: Handlungen
  - o Methods: Reihenfolge der Operatoren für ein (Teil)Ziel
  - o Selection rules: Anwendungsauswahl bei mehreren Operatoren, Kontrollstrukturen
- (Beispiel zur GOMS-Analyse: Keystroke Level Model)
- Ziel ist die Effizienz zu verbessern
- Handlungs- und Denkabläufe werden zu Operatoren

## 2 Evaluationsmethoden

### 2.1 Usability

- Ergonomie: Wechselwirkungen zwischen menschlichen und anderen Elementen eines Systems, die bei der Gestaltung des Arbeitssystems beachtet werden, um das Wohlbefinden und die Leistung zu steigern
- Usability (Gebrauchstauglichkeit): Ziele effektiv, effizient und zufriedenstellen erreichen
- Evaluation: formativ (während des Designprozesses oder summativ (nach dem Design)

#### Usability-Engineering Lifecycle:



#### Testverfahren

- Befragungen: Interviews, Umfragen, Fragebogen, usw.
- Beobachtung: Thinking Aloud, Cognitive Walkthroughs, Formale Usability Inspektion, usw.
- Kreative Methoden:
  - o Fokusgruppen und Workshops
  - o Theater-basierte Evaluation
- Analyse der Software
  - o Modell-basierte Methoden: GOMS Analyse, KLM
  - o Automatische Evaluation: Tastenbelegung eindeutig, Farbkontraste

## 2.2 Evaluation von Zeigehandlungen

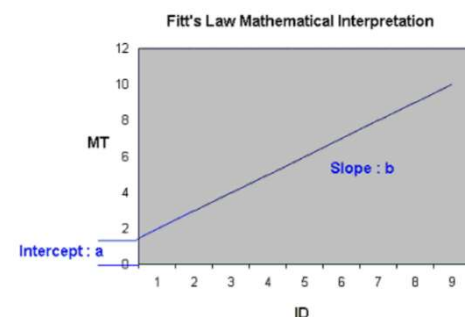
Auswertung von	Dimensionen							
	1		2				3	
Position	Rotations-scheibe	Schieber	Tablet&Puck	Tablet&Stift	Lichtgriffel	Floating Joystick	3D Joystick	Mechanisch
			Touch Tablet		Touch Screen		3D Trackball	Touch
Bewegung	Kontinuierliche Scheibe	Laufband	Maus	Laufteiler		Trackball	Mobiltelefon	Mechanisch
						X/Y Pad		Touch
							ToF Kamera	Visuell
Kraft	Verdrehungs-sensor		Pantograph				Force-Feedback: Falcon	Mechanisch

- Temporale Eigenschaften:
  - o Multimedial: mehrere Medien werden zur Darstellung benutzt
  - o Multimodalität: mehrere Eingabemethoden führen zum selben Zustand im Dialog
  - o Intermodalität: Koordination mehrerer Wahrnehmungsmöglichkeiten und Übernahme von primären und sekundären Funktionen

**Fitts'sches Gesetz:** „Zeit zum Erreichen des Ziels ist abhängig von Entfernung und Größe des Ziels“

$$MT = a + b \log_2 (2A/W + c)$$

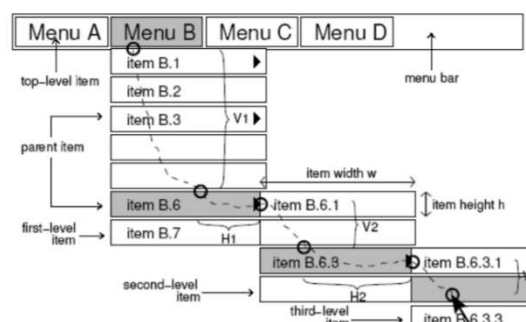
- $\log_2 (2A/W + c)$  = index of difficulty (ID): Schwierigkeit der motorischen Steuerung
- $1/b$  index of performance (IP): Informationsverarbeitungskapazität des menschlichen Bewegungsapparats
- ➔ Große Ziele in der Nähe werden schneller erreicht als kleine weit entfernte Ziele
- Erweiterung MacKenzie & Buxton zur Interpretation von W in 2D:
  - o Status Quo (nur Breite) und Summenmodell (Höhe + Breite) korrelieren signifikant
  - o Status Quo und Flächenmodell (Höhe \* Breite) korrelieren
  - o „Smaller of“ Modell (min (Höhe, Breite)) und W' Modell (Strecke im Ziel zum Mittelpunkt (geeignet für Kreise und Dreiecke)) korrelieren ebenfalls
- Fitts'sches Gesetz „überlisten“
  - o A verringern: pie menus / mit Proxyobjekt temporäre Zielbewegung
  - o W erhöhen: Flächencursor
  - o Beides zusammen: Semantik der Widgets einbeziehen z.B. bei scroll bar



**Steering Law:**  $MT = a + b(A/W)$ , nur für geradlinige Begrenzungen verifiziert!

Menüs mit Fitts' und Steering Law: Steuerung wechselt zwischen vertikal (W) und horizontal (H)

➔ In kaskadierenden Menüs: Fitt's (V1, V2, V3) und Steering (H1, H2)



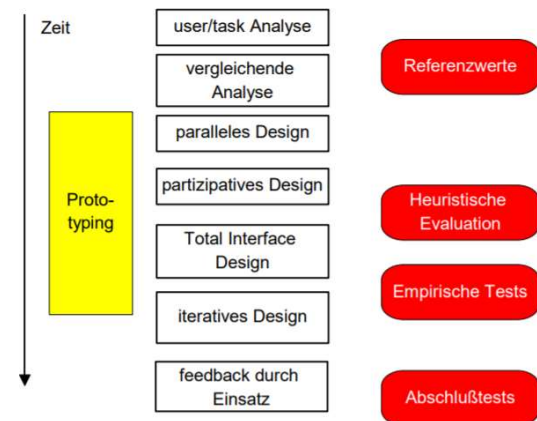
## 2.3 Evaluationsmethoden

### Heuristische Evaluation

- Beispiele für Richtlinien:
  - o Gib Feedback an User
  - o Hilfmeneü immer rechts
- Gruppe von Evaluatoren prüft, ob Programm diese erfüllt
- Bei nur 5 Evaluatoren noch über 30% Varianz

### Cognitive Walktrough

- Für unerfahrene Benutzer werden Aufgaben festgelegt und Beschreibungen angefertigt
- Usability Experten diskutieren Leitfragen:
  1. Werden Benutzer versuchen den gewünschten Effekt zu erzielen?
  2. Werden B erkennen, dass korrekte Handlung ausgeführt werden kann?
  3. Werden B erkennen, dass korrekte H zu gewünschtem Effekt führen wird?
  4. Werden B den Fortschritt erkennen, wenn sie die korrekte H ausgeführt haben?



### Prototypen

- Problem in frühen Phasen: B können wenig mit abstrakten GUI Spezifikationen anfangen
- Schnelle und billige Realisierung mit Prototypen

### Qualitative Methoden

- Thinking Aloud Methode
- Grounded Theory
- Eyetracking
- Fragebögen
  - o Nasa Task Load Index (TLX) als standardisierter Fragebogen (MD, PD, TD, OP, EF, FR)
    1. Pair-wise comparisons of factors
    2. Rating scales für jeweiligen Task
    3. Bewertung:  $\text{rating} \times \text{weight} = \text{product}$ ,  $\text{sum}(\text{product}) / \text{total weight} = \text{Score}$

## 3 Texteingabesysteme

### Texteingabe - Bewertung

- Geschwindigkeit
- Fehlerrate
- Ergonomie
- Lernbarkeit

### Texteingabe - Methoden:

- Sprache und Handschrift (im HCI-Kontext nicht immer geeignet)
  - o Sprache: 100 wpm möglich, voice user interface (VUI) schwer zu entwickeln
- Touch Typing
  - o Hardware Keyboards
    - QWERTY über 150 wpm möglich, 50-70 für Geübte
    - Dvorak (eng., häufige Buchstaben in Ausgangszeile, Alternierung links/rechts)
    - Neo (optimiert für deutsche Sprache)
  - o 12-key Keyboard (Probleme: Auflösen von Mehrdeutigkeit, Word Prediction)

- Virtuelle Tastaturen
  - Unnatürlich aber Potenzial zur Verbesserung durch alternative Layouts
  - Viele verschiedene Ausführungen
- Dynamische Selektion
  - Stroke-based Text Entry
    - T-cube: kaskadierende Pie-Menüs
    - VirHKey: Aufteilung über Pentagonale Kacheln
    - Weitere Nennungen: Cirrin, Quikwriting, Dasher
- Shape Writing
- Gesten
  - Ideen: Abstrakter (robuster, effizienter), näher an lat. Buchstaben (intuitiver)
  - Varianten: MDITIM, EdgeWrite, Unistroke, etc
- Alternative Tastaturen
  - Chording Glove, Thumbcode, per Gyroskope an Fingern, etc.

## 4 Fenstersysteme

### 4.1 Interaktion und Paradigmen

- **Interaktionsmodelle:** helfen dabei die Handhabung von Systemen durch Benutzer zu verstehen
- **Interaktionstechniken:** werden durch die Art des Systems beeinflusst
- Unstimmigkeiten zw. Benutzer und System
  - Ausführungslücke (gulf of execution): formulierte Aktion ~ erlaubte Aktionen
  - Bewertungslücke (gulf of evaluation): physische Präsentation des Systemzustands ~ Erwartungen des Benutzers
- **Interaktionstechniken:**
  - Kommandozeile
  - Menü: Benutzer wird stark geführt, kann rein textbasiert implementiert werden
  - Natürliche Sprache
  - Frage/Antwort Dialog: Verlässlichkeit nicht gegeben
  - Formulare und Tabellenkalkulation: modale Dialoge (Abschluss erzwingen) nicht modale Dialoge (parallel zur Bearbeitung)
  - Window, Icon, Menu, Pointer (WIMP)
  - Zeigen, Klicken, „magnetische Grids“
  - 3D Benutzungsoberfläche

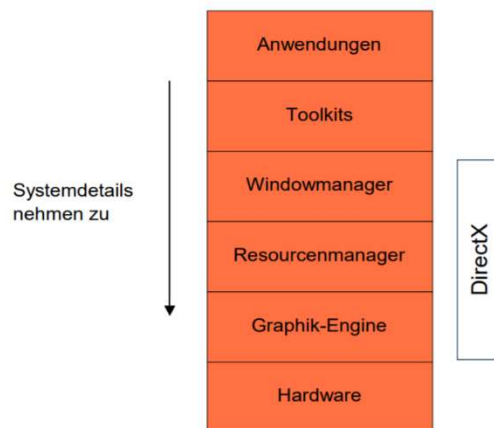
### 4.2 Komponenten einer GUI

#### Ausgewählte Kriterien der Architektur von GUIs:

- Verfügbar
- Einfache Programmierung
- Betriebssystem verbergen
- Grafik-Engine (Raster vs. Vektor)
- Stile (fixiert vs. flexibel)
- Anpassbarkeit (nach Benutzer, nach Land/Kultur, zeitlich)
- Erweiterbarkeit (nicht erweiterbar, Quellcode ändern, dynamisch erweiterbar)
- Parallele Verarbeitung (extern: für Benutzer, intern: Programme könne gleichzeitig laufen)
- Modularisierung von Ressourcen

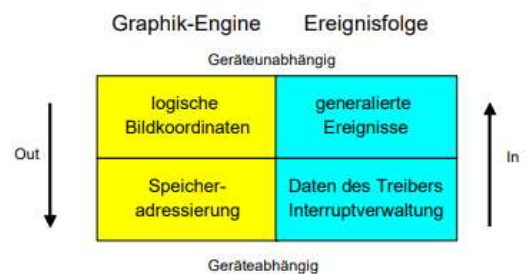


## Komponenten eines Fenstersystems:



### Grafik-Engine:

- Eingabefunktionen: Ereignisse/Events
- Basisoperationen
- Ausgabenfunktionen
- Objekte: Canvas, Font, Pen (Zeichenmodell für Vektorenmodell im Background, Icon)
- ➔ Alle Objekte werden in **Grafikkontext** zusammengefasst



### Ressourcenverwaltung:

- Multiplexing
- Schriftarteneinsatz
- Grafikkontexte
- Farbtabelle
- Remote Zugriff



## Betriebssystem und Fenstersystem

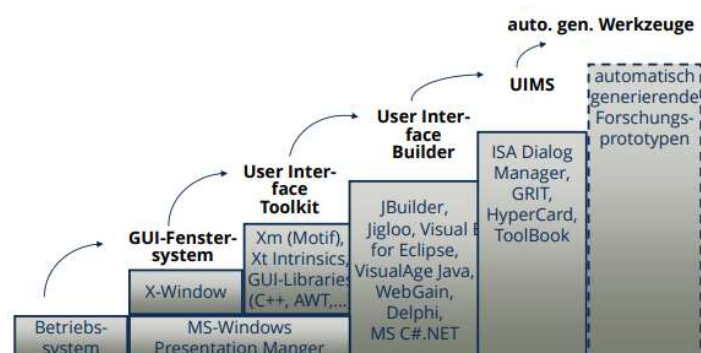
- Adresse und Prozesse

### Fensterverwalter (FM)

- Verantwortlichkeit für Fokus und Verdeckung
- Logischer Bildschirm umfasst evtl. Mehrere Monitore
- Anwendung „denkt“ in Fenstern
- Menütechniken evtl. ohne FM implementiert
- Drag and drop behandelt FM

## Hauptkategorien von GUI-Werkzeugen

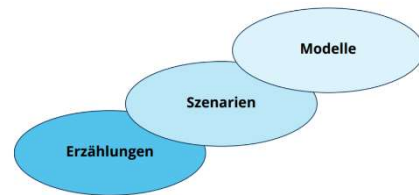
- UIMS: enthält Dialogbeschreibungssprache
- Interface Builder: enthält Oberflächenbeschreibungssprache
- Toolkits: bereitstellung von Oberflächenbaukästen



## 5 Formale Modelle und Zeit

### Ziele formaler Modelle

- Analyse der Benutzeroberfläche auf bestimmte Eigenschaften
- Anwenden von Modellen menschlicher Performance (GOMS)
- Automatische Kritik zum Bildschirmdesign
- Beweis sicherheitskritischer Aspekte



### 5.1 Übergangsdiagramme

- Anwendungsbeispiele: Fahrkartenautomat, Mobiltelefon, Heizungssteuerung, etc.

#### Anwendung: Starke Verbundenheit

- Verbundenheit: zwischen zwei Zuständen besteht ein gerichteter Übergang
- Starke Verbundenheit: von jedem Zustand kann jeder andere Zustand erreicht werden

#### Weitere Analysen

- Charakteristische Pfadlänge (geometrisches Mittel aller kürzesten Wege)
- Exzentrizität: Abstand zum am weitesten entfernten Zustand (von A zu D)
- Durchmesser des Pfads: der Abstand zwischen A und D
- Die kleinste Exzentrizität
- Extremwerte sind genauer zu hinterfragen

#### Effizienzanalyse (Fitts Law)

- $MT = c + d \log((D/W) + 1)$ 
  - o W: Tasten mit Radius r
  - o D Entfernung der Tastenmittelpunkte (x,y)
- Nicht anwendbar für 3D Geräte (Kameras)

#### Analyse durch Simulation

### 5.2 Ereignisse in GUIs

#### Nebenläufigkeit

- **Problem:** Teilautomaten stehen in loser Beziehung zueinander
- **Anson:** die Architektur von Events in GUIs muss als mehrere unabhängige State-Charts konstruiert werden, eine Überführung in ein einzelnes State-Chart ist nicht möglich

#### Kontrollfluss

- Zur Beschreibung wird Interpreter benötigt (kennt alle Übergangsdiagramme (ÜD))
- Falls eine Bearbeitung des Diagramms unterbrochen wurden muss, bestimmt Interpreter ein neues ÜD

#### Interpreter für ÜD

- FROM: Liste von anderen Interaktionsobjekte, von der diese Methoden Variablen erbt
- IVARS: Liste von zu instanziiierenden Variablen und ihre Vorbelegung
- METHODS Vereinbarung von Prozeduren dieses Interaktionsobjekts
- TOKENS: Definition jedes Ein- und Ausgabeelements, das im Syntaxdiagramm für dieses Element verwendet wird
- SYNATX: ÜD, dass die Abfolge der Verarbeitung von Tokens und andere Aktionen festlegt
- SUBS: zusätzlich verwendete ÜD
- STATES: zusätzliche Beschreibung von Übergängen

### 5.3 Event Response Language (ERL)

- **Event Response System (ERS)**: kann auch Nebenläufigkeit intrinsisch beschreiben
- ➔ Sprache für ERS wird als **ERL** bezeichnet

### 5.4 Zeit und Interaktion

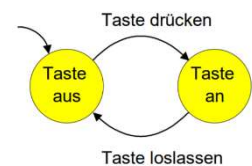
- Verschiedene Phasen bestimmen das Erleben der Interaktion
  - o Eingabezeit, Antwortzeit, Ausgabezeit, Denkzeit
- Fokuskonzept zur Serialisierung paralleler Aufgaben

#### Ereignis Intervall Regelsystem (EIRS)

- **Zeitintervall**: Anfangszeitpunkt  $t_-$ , Endzeitpunkt  $t_+$
- **EIRS** ist Tupel mit  $(F, I, R, \Omega, S, f_f)$ 
  - o **F Bedingungsvariablen**, die bewertet werden
  - o **I Ereignisse** und **Intervalle**, die den Verlauf beschreiben
  - o **R Regeln**
  - o  $\Omega$  **Wurzelintervall** aus allen Zeitpunkten  $t$ , das damit beginnt, dass die Zustände in  $S$  wahr werden und das damit endet, dass  $f_f$  als „wahr“ bewertet wird
  - o  $S \subseteq F$  **Bedingungsvariablen**, die initial „wahr“ sind
  - o  $f_f \in F$  ist die letzte **Bedingungsvariable**, die „wahr“ wird, und das Ende des Systems angibt

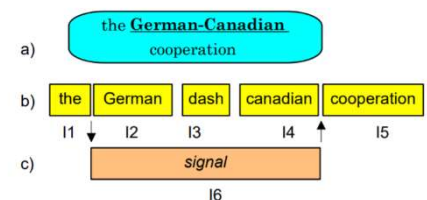
#### Temporale Modelle

- Temporale Modelle einer Benutzeroberfläche sind solche, die
  - a) die zeitliche **Strategie des Benutzers** während der Interaktion beschreiben,
  - b) Bedingungen für zeitliche **Beschränkungen** während der Interaktion angeben oder
  - c) die **Nebenläufigkeit** von Interaktionsschritten beschreiben



#### Intervalldiagramme

- Intervall-Modell beschreibt durch temporale Intervalle die Lebensdauer eines Objekts oder die Ausführbarkeit von Methoden

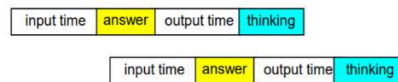


#### Ereignisse und Intervalle

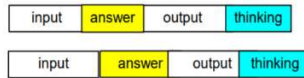
- Ereignisse sind Kontaktpunkte
- Ereignisse sind selbst Kontaktpunkte (KP)
- Direkte KP benötigen einen Bezug in beiden Medien
- Indirekte KP stellen einen Bezug auf einer vorhergehenden Referenz in anderen Medien her

## Interaktion: Parallele Modelle

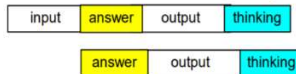
Parallelität durch erneute Eingabe



Parallelität durch zwei Eingabegeräte

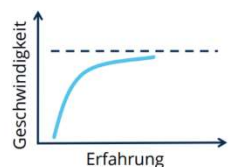
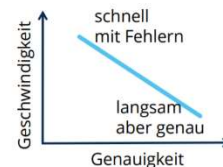


Parallelität durch zwei Ausgabegeräte



### Gestaltung der Eingabezeit

- Geschwindigkeit vs. Genauigkeit vs. Erfahrung
- Hick-Hyman Law, 1952: je mehr Entscheidungen, um so langsamer die Reaktionszeit RT
- $RT = a + b \cdot H$ , wobei H die Entropie (Information in Bits) sowie a und b Konstanten



### Zeitliche Wahrnehmung vs. Toleranz

- Benutzer empfindet Ausgabezeit oft als inakzeptabel
  - o Management der Operationen
  - o Management der Wahrnehmung
  - o Management der Toleranz
- Die Festlegung von Obergrenzen für die Ausgabezeit kann durch vielfältige Beeinträchtigungen bei der Wahrnehmung und Erwartungen des Benutzers hinfällig werden, z.B. durch
  - o aktive vs. passive Vorgehensweise bei der Wahrnehmung
  - o Ansprechbarkeit

### Ansprechbarkeit

- **Sofortheit:** 0,1 – 0,2 s (öffnen des Menüs)
- **Unmittelbarkeit:** 0,5 – 1s (positive Bestätigung einer Eingabe)
- **Kontinuität:** 2 – 5s (Rückmeldung eines Verarbeitungsschritts)
- **Beachtung:** 5 – 10s (max. Aufmerksamkeitsspanne zur Verfolgung einer Aufgabe)

### Bewertung einzelner Intervalle <30s

- Verallgemeinerung: erst Veränderungen zeitlicher Intervalle um mehr als 20% werden wahrgenommen

### Vergleich zweier Intervalle

- erfolgt nicht auf Basis des arithmetischen, sondern des geometrischen Mittels

### Zeiteinteilung

- **Prospektive Angaben**
- **Echtzeitangaben**
- **Retrospektive Angaben**

### Management der Wahrnehmung

- Vorzeitiger Stark
- Frühes Ende/ unsichtbares Beenden

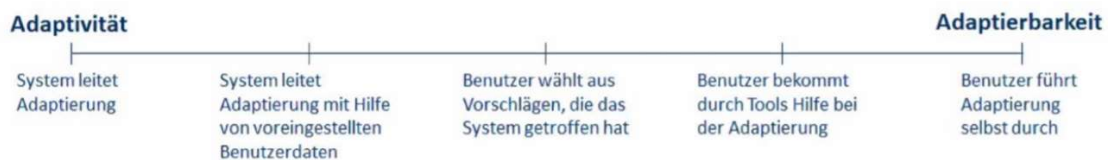
- Abnehmende Intervalle/ nicht-lineare Fortschrittsanzeige
- Sinnvolle Ablenkung

### Management der Toleranz

- Übererfüllung bei vorsichtiger Planung
- Benennen der Wert der Wartezeit (Anzeige der geprüften Flüge)
- Puffer und progressiver Aufbau des Ergebnisses
- Einmal-Effekt (einmalige Aktionen entsprechend benennen)
- Benchmark-Funktion
- sekundengenaues Beenden in der Schlussphase

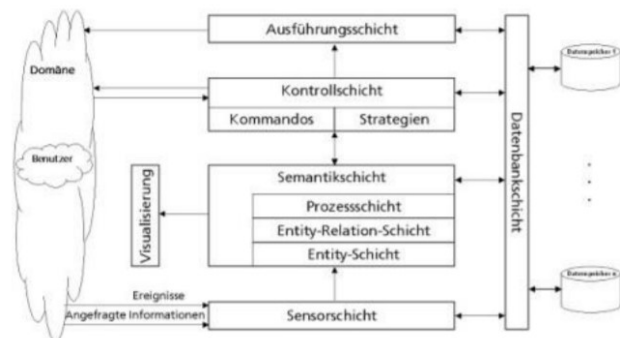
## 6 Adaptierung und Adaptivität

- **Adaptivität** beschreibt die **selbständige** automatische Anpassung des Systems an Eingaben oder auch von Ausgaben
- Von **Adaptierbarkeit** wird gesprochen, wenn durch Anpassungen Einstellungen des Programms verändert werden können



### Adaptivität in der Interaktion

- **Sensorschicht** ermittelt
  - o Veränderung des Kontext
  - o Interaktion des Benutzers mit dem Kontext
- **Semantiksicht**
  - o Identität des Benutzers
  - o Lokation
  - o Zeit
  - o Beziehungen zu anderen Objekten
- **Kontrollschicht**
  - o Basissteuerung mittels Regeln
  - o Befehlsstrukturen für komplexe Anpassungen
- **Ausführungsschicht**
  - o Domänenabhängige Implementierung der Steuerung

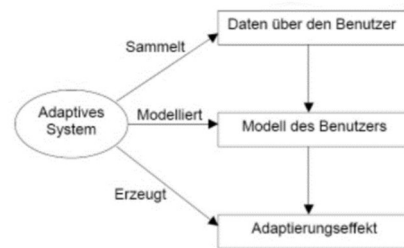


### Methodik adaptiver Systeme

1. Afferenz (Beobachtung und Sammlung von Informationen)
2. Inferenz (Auswertung der gesammelten Informationen)
3. Efferenz (Anpassung des Systems)

## Benutzermodell

- Wertebereiche: binär, statisch, symbolisch
- Daten über den Benutzer
  - o Vorhandenes/ fehlendes Wissen
  - o Pläne und Ziele des Benutzers
  - o Präferenzen
  - o Fähigkeiten
  - o Missverständnisse



## Afferenz: Daten des Benutzers

- Benutzermerkmale
- Aufgabenmerkmale
- Umgebungsmerkmale
- Technikmerkmale

## Inferenz: Schlussfolgerungen

- Steuerung muss mit unsicheren/ mehrdeutigen Indikatoren umgehen)
  - o Wissensrepräsentation
  - o Inferenzmechanismen
  - o Psychologische Strategien

## Efferenz: Ausgaben (Umsetzung der Ergebnisse der Inferenz)

- Für verschiedene Leistungen
  - o Inhalt
  - o Ausgabemedium
  - o Benutzerschnittstellen
  - o Empfehlungen
  - o Navigationshilfe

## Adaptive Hypermedien

- Hypermedien unterstützen räumliche und temporale Verweise in zeitabhängigen und zeitunabhängigen Medien
  - o Tutorielle Systeme
  - o Onlinehilfesysteme etc.
- Statische Hypermedien haben Einschränkungen
  - o Keine Anwendungsziele oder Lernziele
  - o Interessen nicht ermittelbar
  - o Vorkenntnisse unbekannt etc.

## Tutorielle Systeme

- Durch Adaptivität kann eine höhere Lernleistung erzielt werden
- **Voraussetzungen**
  - o Klare Strukturierung des Wissensgebiets erforderlich
  - o Lernsteuerung verhindert Wiederholungen und ermöglicht Vertiefungen, Nachholen von Vorwissen
- **Probleme** im aktuellen System
  - o Vorwissen sehr verschieden
  - o Andere Lernstrategien werden nicht berücksichtigt

## Information Retrieval

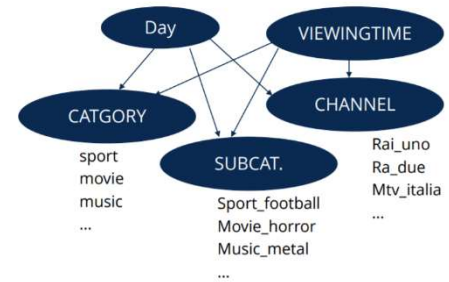
- **Adaptive Führung** (auszeichnen aller relevanten Verweise einer Seite)
- **Adaptive Annotation** (generieren von Hinweisen)
- **Adaptive Empfehlung**

## Stereotypisiertes Benutzerprofil

- Mit **Klassifikationsdaten** und **Vorhersagen**
- Verwaltung: Benutzerkriterien werden mit Stereotypen verglichen

## Dynamisches Benutzermodell

- Bayes'sches Netzwerk von Annahmen verwendet z.B. Netica
- Basiert auf Kontextvariablen (Tag, Uhrzeit)
- Wunschvariablen (Kategorien Unterkategorie, Kanal)



## Kombinieren von Benutzermodellen

- Es muss eine Entscheidung für eine Gruppe von Nutzer gefunden werden
  - o Likert-Skalen (Wertung von 1 – 10)
- Abstimmungsmethoden
  - o **Pareto Regel** (Verbesserung, ohne jemanden schlechter zu stellen)
  - o **Anonymität** (Vermeidung von Diktatoren)
  - o **Positive Assoziation** (x vor y, auch noch nach Erhöhung von x)
  - o **Condorcet-Gewinner** (x ist Condorcet-Gewinner, falls dieser vor allen anderen Alternativen bevorzugt wird)

## Kollaborative adaptive Empfehlungssysteme

- Gleichgesinnte ermitteln eine Empfehlung
- Das System adaptiert Verweise selbstständig

## Das Bewertungsproblem

- Ermitteln der Bewertung, die der Benutzer nicht kennt
- Ansätze
  - o Inhalts-basiert: Objekte mit ähnlichen Eigenschaften werden ähnlich bewertet
  - o Kollaborativ: mit Hilfe von Clustern

	Inhaltsbasiert	Kollaborativ
<b>Vorteile</b>	<ul style="list-style-type: none"> <li>▪ Empfehlung unbewerteter Objekte möglich</li> <li>▪ Unabhängig von der Benutzerzahl</li> <li>▪ Außergewöhnliche Präferenzen werden berücksichtigt</li> </ul>	<ul style="list-style-type: none"> <li>▪ Unabhängig von den Objekten für die Empfehlung</li> <li>▪ Unabhängig von früheren Empfehlungen</li> </ul>
<b>Nachteile</b>	<ul style="list-style-type: none"> <li>▪ Objektbeschreibung ist notwendig</li> <li>▪ Mindestzahl von Bewertung von einem neuen Nutzer ist notwendig</li> <li>▪ keine subjektiven Kriterien</li> <li>▪ keine Berücksichtigung der Erkenntnisse andere Benutzer</li> </ul>	<ul style="list-style-type: none"> <li>▪ Kaltstart (neues Benutzer, neues Objekt unsicher)</li> <li>▪ Bei schwach besetzter Matrix -&gt; niedrige Empfehlungsqualität</li> <li>▪ Popularitätsausrichtung</li> </ul>

## Hybride Empfehlungssysteme

- Vorgehensweise
  - o Kombination: Ergebnisse getrennt ermitteln, bestes Ergebnis mit bester Konfidenz wählen
  - o Inhaltsbasierte in kollaborative Technik integrieren
  - o Kollaborative in inhalts-basierte Technik integrieren



## Erweiterungen

- **Weitere Anforderungen:** Skalierbarkeit, Nachvollziehbarkeit, Präferenzermittlung optimieren, Dialogbasiertes Empfehlen
- **Empfehlungsqualität:** Qualitätsmetriken, Kontext berücksichtigen, Diversifikation der Themen
- **Sicherheit:** Datenschutz, Manipulationsschutz

## 7 Navigationssysteme

### 7.1 Navigation mit mobilen Endgeräten

#### Mobile Endgeräte

- Größe: Miniaturisierung
- Reichenleistung nimmt zu, Preis sinkt kontinuierlich
- Anwendungen: Mobile Computing wird zunehmend realisiert
- Eingabe: Tastatur, Touchscreen (GUI, Gesten, Handschrift), Sprache, Haptik
- Ausgabe: Bildschirm, Sprache, akustische Signale, Musik, Klänge, Vibration
- Anwendungen haben nur sehr kleinen Ausgabebereich zur Verfügung
  - o Websites automatisch transformieren?
  - o Welche Interaktionstechniken realisieren?
  - o Welche Anwendungen überhaupt sinnvoll?

#### Systeme

- GPS(USA), GLONASS(Russland), GALILEO(EU), COMPASS(China)
  - o Verwendung mehrerer Systeme parallel erhöht Genauigkeit

#### Satellitennavigation

- Systeme bestehen aus 3 Segmenten
  - o Weltraumsegment
  - o Kontrollsegment (Bodenstation)
  - o Benutzersegment (Anwender)

#### Positionsbestimmung

- Auswertungsalternativen: **Trägerphasendifferenzen** der Wellen
- Verfahren zur Erhöhung der Genauigkeit
  - o Assisted GSP: zusätzliche Bestimmung durch Mobilfunkmasten (recht ungenau)
  - o Differential GPS: Einbezug der terrestrischen Referenzstationen (kostenpflichtig)

#### NMEA-Nachrichtenformat

- National Marine Electronics Association, aktuell Standard NMEA-0183
- Laut Standard: nur ein Sender und mehrere Empfänger
- Daten werden ASCII-codiert übertragen, jeder Datensatz beginnt mit \$
- Verschiedene Datensätze u.a. GPRMC, GPGLA, ...

#### Anwendungen

- Verkehres- und Seefahrtüberwachung
- Überwachung von Notfalleinsätze
- Landwirtschaft

#### Fahrzeugnavigation

- Berechnung schnellsten/kürzester Route
- Dynamischer Einbezug von Verkehrsinformationen



## 7.2 Fußgängernavigation

- Mikronavigation („halten der Spur“): Entfernungen <10m
- Makronavigation (Navigation zum nächsten Wegpunkt): Entfernungen >50m

### Räumliche Vorstellung

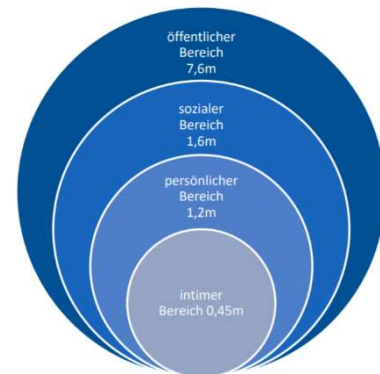
- Egozentrisches (Kurzzeitgedächtnis) vs. allozentrisches (Langzeitgedächtnis)
- Geometrisches Verständnis
  - o Auch ohne visuellen Eindruck
  - o Einfluss auf ego-/allozentrisches Modell je nachdem ob (geburts-)blind, Augen verb.
- Projektion 3D Strukturen auf Flächen
- Makro-Effekte wie Stern-Relation, Kultur, Vegetation und Klima werden berücksichtigt
- Integration räumlicher und zeitlicher Relation

### Kognitive Verarbeitung

- Wissensaufbau erfolgt unterschiedlich und parallel
  - o Landmark, point of interest
- Lernen von Karten vs. lernen von Routen

### Proximität

- Menschen haben Raumverständnis mit unterschiedlichen Distanzzonen
- Je nach Kulturen variieren Entfernungen



### Fußgängernavigation

- **Zusätzliche Anforderungen für Routenfindung**
  - o Auflösung/Detaillierungsgrad muss höher sein
  - o Genauere Positionsbestimmung notwendig
  - o Multimodalität der Fortbewegung berücksichtigen
- **Mobilitätseingeschränkte Fußgänger**
  - o Motorisch Behinderte
  - o Senioren Sehgeschädigte
  - o Gehörlose Personen
  - o Dyslexika
  - o Situationsbedingte Einschränkungen (z.B. Kinderwagen)
- **Zusätzliche Anforderungen mobilitätseingeschränkter Fußgänger**
  - o Anderes Kartenmaterial notwendig
  - o Zusätzliche geografische Informationen, komplexere Routenberechnung
  - o Differenzierte Anforderungen an Ein- und Ausgabesystem
  - o Einbezug multimodaler Fortbewegung wichtig

### Mobilitätshilfsmittel

- Hindernisvermeidung: Blindenlangstock etc.
- Strukturelle Installationen: Talking Signs, Talking Braille
- Navigationssysteme: Trekker, Kapten

## 7.3 Kollaboratives Benutzermodell einer Gruppe durch Annotation

### Multimodale Annotation

- **Lösungsansatz:**
  - o Verwendung einer Methodik für semi-automatische Annotation geographischer Daten durch die eigentlichen Benutzer
  - o Verwendung adaptiver Navigationsalgorithmen für Berechnung von Routen optimiert für verschiedene Benutzergruppen

- Verwendung von Methoden aus dem Bereich des Multicriteria Decision Making für die Auswahl einer optimalen Route
- Generierung von umfangreichen und personalisierten Navigationsanweisungen auf Basis der Annotation

### **Annotationen**

- (persönliche) Attribuierung geographischer Objekte
  - Points of interest
  - Manuell durch Benutzer erbrachte Informationen (Wegbeschaffenheit)
  - Automatisch erfassbare Informationen (Dauer einer Strecke)

### **Annotiertes Wegenetz**

- **Verschiedene Gewichte an den Kanten des Navigationsgraphs**
  - Gewichte für: Durchschnittszeit, Nutzerbewertung, Orientierungsmerkmalen

### **Mehrkriteriale Routingverfahren**

- Berechnung des Weges:
  - Für jede Kante müssen Kriterien berücksichtigt werden
  - Gruppen- und Nutzerprofile definieren Kriterien und Gewichtung
  - Unter Verwendung von MCDM entsteht konsolidiertes Gewicht je Kante

### **Multicriteria Decision Making (MCDM)**

- Oberbegriff für Entscheidungsproblem mit mehreren Zielen
  - Ziele oft im Konflikt zueinander
- Werden in Multi-Attribute DM und Multi-Object DM unterteilt

### **Routenberechnung mittels MCDM**

- Lösungsraum für Routenberechnung ist diskret → Multi-Attribute
- Schlechte und gute Bewertungen können sich ausgleichen
- Für Vergleichbarkeit muss Normierung durchgeführt werden

### **Entscheidungsregeln**

- Ohne Berücksichtigung von Präferenzen
  - **Dominanz-Strategie**
  - **Maximin-Strategie**
  - **Maximax-Strategie**
- Mit Berücksichtigung von Präferenzen
  - **Konjunktive Strategie**: Festlegung von Mindestgrenzen
  - **Disjunktive Strategien**: Festlegung von Mindestgrenzen, aber mit Alternativen, die bestimmte Anzahl von Grenzen nicht erreichen
  - **Lexikographische Strategie**: Anwendung konjunktive Strategie, mit Reihenfolge bis nur eine Alternative übrigbleibt
- unter Berücksichtigung kardinaler Attributinformationen
  - **einfache additive Gewichtung**:  $\text{Attribut1} * \text{Gewichtung1} + \text{Attribut2} * \text{Gewichtung2} + \dots = \text{Wert}$

### **Wege und Attribute**

- beste Wege benötigen Attribute
- Erstellen durch kollaborativen Ansatz: „Gesetz der großen Zahlen“

### Routenberechnung für multiple Kriterien

- Dijkstra Algorithmus
- Likert Skala
- Normalisierte Likert Skala
- Gewichtete Likert Skala

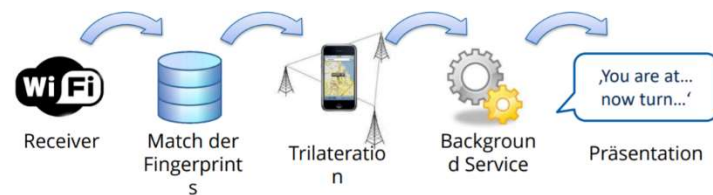
## 7.4 Navigation in Gebäuden

### Unabhängigkeit bei Wegen

- Unbekannte und komplexe (teilweise öffentliche) Gebäude
- Web-App für Desktop Webbrowser
- Barrierefreie Smartphone App

### Indoor-Lokalisierung

- GPS ohne Empfang, dafür W-LAN
- Probleme: Fingerprints schwanken
- Korrekturen/ Feiltern nötig



### Wegbeschreibungen erzeugen

- Fußgänger bewegen sich nicht auf vordefinierten Bahnen
  - o Schritt für Schritt Ansagen vermeiden
  - o Ergänzen der Mobilitätsfähigkeiten
  - o Räumliche Informationen sollen die Gebäudestruktur erläutern
  - o Beschreibung auch als statisches Hörbuch

### Explore by Touch

- Erkunden, selektieren, swipe

### Informationsdarstellung

- Grafisch (Wegeplan)
- Statische Beschreibung (annotierter Text)
- Strukturierte Informationen (annotierte Attribute)
  - o Typ des Elements
  - o Etageebene
  - o Warnung vor Gefahren

### Kartenerstellung und Verteilung

- Annotation für Gebäudeplan (SVG + XML)
  - o Ersteller: sehende Sachverständige für Bauten/ Dritte
  - o Speichern auf Webserver, Download auf Client
- Strukturierung kann variieren und ist vom Annotierenden abhängig
  - o Vorbereitete Beschreibungstexte erforderlich
  - o Attribute näher beschreiben

### Andere Lokalisierungstechniken

- Aktiv:
  - o RFID Tags im Boden
  - o QR Code schwarzweiß/farbig
  - o Markercode
- Passiv:
  - o Kamera

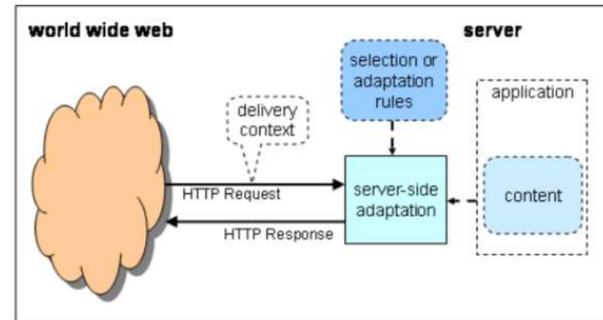
- Deckenlampen identifizieren
- Magnetische Positionierung
- Bluetooth Beacons

## 8 Entwicklungswerkzeuge von Benutzeroberflächen

### 8.1 Modelbasierte Transformation

#### User Interface Description Language (UIDL)

- Eine UIDL ist eine formale Sprache für die Zwecke der MCI, um eine konkrete Benutzungsoberfläche zu beschreiben
- UIDL sind abhängig von der verwendeten Implementierungstechnologien
- Zu einer UIDL gehören:
  - Semantik: UML2, OWL2, ...
  - Abstrakte Syntax: unabhängig von jeglicher Repräsentationssprache
  - Konkrete Syntax: XML, RDF, ...
  - Stilistische Elemente: Grafik, ...



#### State of the Art der IDEs

- **Verschiedene IDEs für skalierbare UI:** Eclipse4SL, Android Studio, NetBeans JavaFX
- **Grenzen der Skalierbarkeit:**
  - Aufgaben, die einen Gerätewechsel erfordern/zulassen
  - Lösungsansatz: Cloud-based computing und geeignete UI

#### Kontext der Benutzung

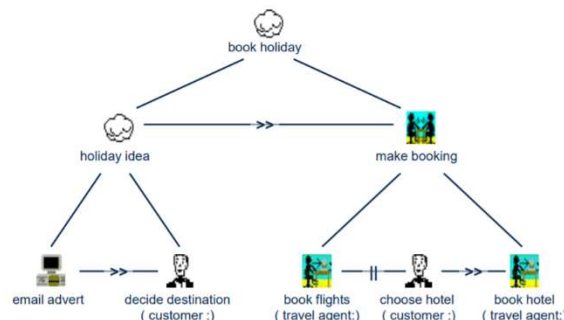
- Der Kontext der Benutzung eines interaktiven Systems ist ein dynamische, strukturierter Informationsraums mit den folgenden Entitäten:
  - Modell von **Benutzer U**
  - Modell von **Plattform P**
  - Soziales und physisches Modell **Umgebung E** in der die Interaktion stattfindet
- Beispiele:
  - **Multi-target UI:** mehrere U, P und E
  - **Adaptive UI:** UI berücksichtigt den Kontext der Benutzung und kann sich an Kontextänderungen anpassen
  - **Adaptable UI:** UI kann durch Menschen angepasst werden
  - **Plastizide UI:** multi-target UI die ihre Gebrauchstauglichkeit über multiple Zielplattformen erhält

#### Formale Aufgabenmodelle

- ConcurTaskTrees (CCT)

#### CCT entwerfen

- Aufgaben hierarchisch zerlegen: Aktion + Objekt
- Identifizieren der temporalen Relationen zw. Tasks auf derselben Ebene
- Objekte die manipuliert werden: Element, Liste, Tabelle, Kollektion, Verbund, ...
- Aktionen darauf ausführe: rendern, modifizieren, ableiten, berechnen, ...



## CCT

- **Zwei Arten von Objekten**
  - o Wahrnehmbare Objekte: Interaktion mit dem UI
  - o Anwendungen: Interaktion mit dem Kern
- **Vereinfachen des UI durch:**
  - o Filtern der relevanten Tasks
  - o Abbilden der Aufgaben auf Objekte und deren Kommunikation
- **Abbilden der Aufgaben auf Presentation Task Sets (PTS)**
  - o PTS entsprechen Aufgaben, die zur selben Zeit aktiv sind
  - o PTS werden durch Zustandsübergänge aus Basis einer Heuristik verbunden

## Groupware Task Analysis (GTA)

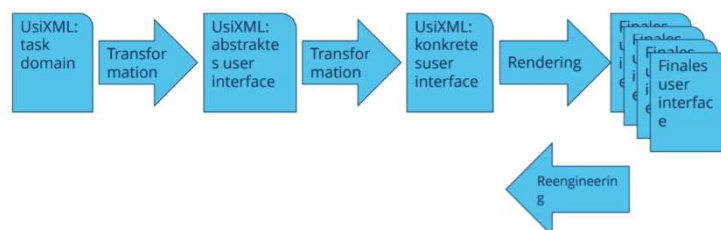
- o Groupware, Workflows, etc. erfordern noch mehrere Benutzer in die Aufgabenanalyse einzubinden
- o Eigener konzeptioneller Rahmen, Werkzeuge, Ermittlungstechniken
- **Reichhaltige Modelle der Aufgaben (tasks)**
- **Reichhaltige Ontologie**
  - o Diverse Rollen in der Kollaboration physischer und elektronischer Objekte

## UsiXML

- Beschreibung des Modells der Benutzeroberfläche in einer gemeinsamen XML-Sprache
  - o Modular erweiterbar
- **User Interface extensible Markup Language beschreibt**
  - o Character UI
  - o Graphical UI
  - o Auditory UI
  - o Multimodal UI
- **Mittels einer UIDL mit den Zielen**
  - o Geräteunabhängigkeit
  - o Plattformunabhängigkeit
  - o Unabhängigkeit von Modalitäten
  - o Wiederverwendung von Komponenten

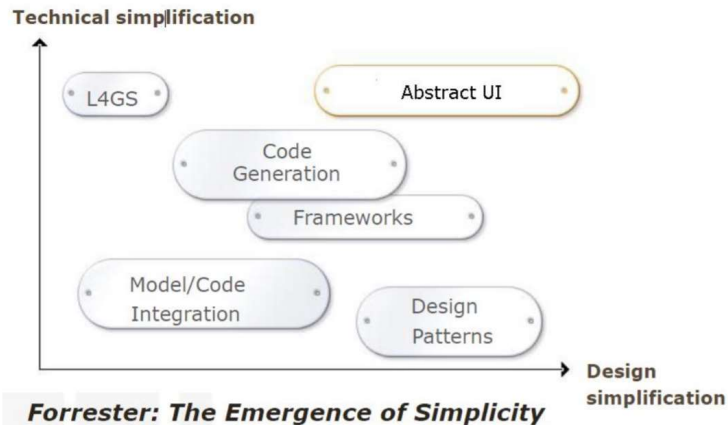
## UI Engineering mit UsiXML

- **Model-based Engineering**
  - o Transformationen
- **Rendering** (von Code wie z.B. HTML)
- **Model-driven Engineering**
  - o Transformation wird selbst in einem formalen Modell beschrieben



## 8.2 Visuelles Programmieren

### Vereinfachung der Entwicklung



### Beispiele für Visuelles Programmieren:

- AgentSheet, Scratch, Labview
- ➔ „Verbannen aller Textelemente“, Editieren durch Drag&Drop

### Visuelle Programmiersprachen

- Visual Interaction Language:
  - o ER-Diagramme
  - o Kontrollflussdiagramme
- (Iconic) Visual Information Processing Language:
  - o Bildverarbeitungssprache
  - o NC-Programmiersprache
- **Probleme:**
  - o Fehleranfälligkeit
  - o geringe Komplexität der mentalen Modelle
  - o Abhängigkeiten zw. Diagrammen

## 8.3 Ausblick

### Zukünftige Interaktion

- Parallel, hoch-interaktiv
- Kontinuierlich (+diskret)
- Multimodale Benutzeroberflächen (Sprache, Gesten, Körperbewegungen)
- Etc.

### Auswirkungen auf Software

Heute	Zukunft
Single thread I/O	Parallel, asynchrone Dialoge; mit Querbezügen
diskrete Tokens	kontinuierliche Eingaben und Antworten (plus diskrete)
präzise Tokens	Probabilistische Eingabe, nicht leicht in Blobs einteilbar
Sequenz, nicht Zeit	Real-time, Deadline-basiert
kurzfristig	Mittel und langfristiges Beobachten der Benutzer

## Schlussfolgerungen

- Hintergrund für aktuelle Forschung zu Software für Benutzungsoberflächen
  - Interaktionsstile
  - Unabhängige Dialoge
  - Barrierefreiheit
  - Vier Designebenen
  - User interface Management System (UIMS)
  - User Interface Description Language (UIDL)
  - Ziel: Erzeugen der UI von abstrakten Beschreibungen
  - Mehr Softwarewerkzeuge
- Was kommt in der Praxis an?
  - Trennung der Benutzungsoberfläche von der Anwendung
  - Model-View-Controller Architektur
  - Evaluation
  - Spezielle Tools
  - Spezielle Sprachen für portable Anwendungen