

Ubiquitous Computing

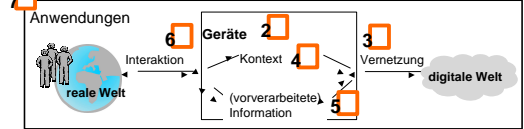
(Ubiquitäre Informationstechnologien)

Vorlesung im WS 03/04



Aufbau der Vorlesung

- 1 Grundlagen
- 2 Geräte allgemein
- 3 Vernetzung
- 4 Kontext
- 5 Information
- 6 Interaktion
- 7 Anwendungen



Wiederholung Grundlagen

- Ubiquitous Computing: M. Weiser, XeroxParc
- Werkzeuggedanke steht im Vordergrund, dadurch Einbettung des Computers in die Welt
- „Gute Technologie ist unsichtbar“, Mensch keine „Nutzermaschine“
- Erste Experimente: ParcTab, Augmented Environments
- Kontexte ermöglichen neue Möglichkeiten, Kontexte ausgezeichnetes Forschungsthema in Ubicomp
- Ansatz Kontexterkenkung: Umgebung oder Artefakt
→ Intelligente reaktive Umgebung oder smarte digitale Artefakte
- Probleme: Anwendung, HCI, Technologie

Appliances

Gestaltung: Affordances

Geräte

- D. Norman Information Appliance Konzept
- Geräteklassen
- Perönlich zugeordnete Geräte
- Geräte mit Prämerfunktion
- Digitale Artefakte

Benutzerschnittstelle

- Software
- Hardware

Gestaltung Affordance Konzept

Etwas Mensch-Maschine Interaction:

James Gibson, 1977: the theory of affordances

- Beeinflußt von Gestalt Theorie, aber „ecologischer“ (Entdecker-) statt konstruktivistischer Ansatz

Gestalt Theorie

- Menschen erkennen die Dinge/Abläufe im Zusammenhang und interpretieren diese innerhalb des aktuellen Kontextes
- Beschreibt, wie Dinge als zusammengehörig empfunden werden
- **Proximity:** Räumliche Nähe von Dingen
- **Similarity:** Ähnlichkeit im Aussehen, Bedienung, Kontext,...
- **Closure:** Im Zusammenhang nützlich
- **Simplicity:** Wenn sie zusammen ein vereinfachtes Bild abgeben

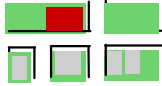
Gestaltung Affordance

Was ist das?



Gestaltung Affordance

Was ist das?



Gestaltung Affordance Konzept

Don Norman

- Human-centered design: "The Design of Everyday Things", 1988
- affordance refers to the perceived and actual properties of the thing



Idee

- Gestaltung von (alltäglichen Dingen) unter Ausnutzung der Affordance Eigenschaft

3 Prinzipien zur Gestaltung

- Mache nutzbare Eigenschaften sichtbar
- Benutze "natürliche" Assoziationen zur Verdeutlichung
- Gib "Feedback"



Gestaltung Affordances

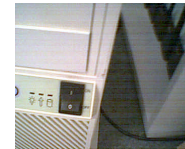
Gegenstände/Umgebungen legen ihre Benutzung/Funktion dem Menschen durch ihre Erscheinung nahe

- „Empfundene“ Eigenschaften
- Beispiel: Türklinke animiert zum Drücken, Tasse zum Trinken, Absperrungen zum nicht-Überschreiten
- Zum Teil anerzogene, zum Teil angeborenes Wissen über Funktion-zu-Erscheinungszuordnung

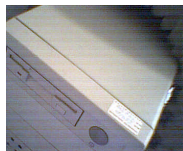
Rahmenbedingungen für Affordances

- Physikalische Eigenschaften
 - Große Dinge passen nicht in kleine Dinge
 - Sehr kleine Tasten müssen mit einem Hilfsmittel bedient werden
- Semantische Eigenschaften
 - Man fährt in Fahrtrichtung sitzend, also Lenkrad in Fahrtrichtung
- Kulturellen Bedeutungszuordnung
 - Rote Hervorhebung bedeutet Wichtigkeit
- Logische Eigenschaften
 - Die letzte Schraube ist für das letzte Loch gedacht

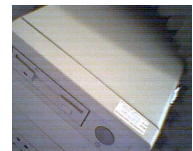
Gestaltung Affordance Beispiele I



Gestaltung Affordance Beispiele I

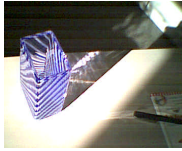


Gestaltung Affordance Beispiele II



- Bedienung offensichtlich ...
- ... falls Bedienung angelernt wurde
- Appliances müssen nicht auf Allgemeinwissen beruhen
- ... aber das Wissen kann in Allgemeinwissen diffundieren

Gestaltung Affordance Beispiel III



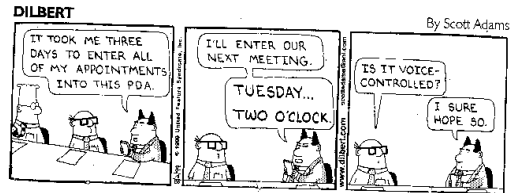
- Ist eigentlich eine Vase
- Manchmal Designer-Intention nicht Nutzer-Intention
- Einfach möglich in realer Welt
- Einfach möglich in virtueller Welt?



Ubiquitous Computing WS 03/04 Michael Beigl, TecO

3-13

Gestaltung Affordance



Ubiquitous Computing WS 03/04 Michael Beigl, TecO

3-14

Geräte

Gestaltung: Affordances

Geräte

- Information Appliance Konzept
- Geräteklassen
- Persönlich zugeordnete Geräte
- Geräte mit Primärfunktion
- Digitale Artefakte

Benutzerschnittstelle

- Software
- Hardware

Ubiquitous Computing WS 03/04 Michael Beigl, TecO

3-15

Geräte Appliance Konzept

Appliance

- „das Anwenden, die Anwendung, der Apparat, die Bedienung, das Gerät, das Hilfsmittel, die Verbrauchseinrichtung, die Vorrichtung“ (Quelle: dict.leo.org)
- „A device or instrument designed to perform a specific function, especially an electrical device“ (American Heritage Dic.)

Information Appliance

- D. Norman: „An appliance specializing in information. ... A distinguishing feature of inform. Appliances is the ability to share information among themselves“
- Beispiel: DigiCam, Taschenrechner
- Benutzer notwendig

3 Axiome für Information Appliances

1. **Einfachheit:** der Bedienung für Benutzer (so einfach wie dies die Aufgabe erlaubt!)
2. **Vielseitigkeit:** erlaubt und fördert Kreativität
3. **Vergnüglichkeit:** sollen erfreulich sein, Spaß machen



Geräte Appliance-Konzept

Appliance-Konzept

- Betonung des „Werkzeug“ Gedankens und der funktionalen Abgeschlossenheit
- Computer Sekundärartefakt, der Funktion und Bedienung untergeordnet
- „You don't want to use a computer“

PC / Application vs. Appliances

- PC: Universalrechner (Hardware) und Universalbedienung mit spezielle Anwendung (Application)
- ... führt zu einer komplizierteren Bedienung (Verstoß gegen Axiom 1)
- ... und meistens auch zu wenig Spaß (Axiom 3)

Tool vs. Appliances

- Tool: Werkzeug kann nicht mit anderen Werkzeugen kommunizieren
- ... führt zu weniger Kombinationsmöglichkeiten (Verstoß gegen Axiom 2)
- ... und so meistens auch zu wenig Spaß (Axiom 3)

Ubiquitous Computing WS 03/04 Michael Beigl, TecO

3-17

Geräte

Gestaltung: Affordances

Funktionszuordnung

Information Appliance Konzept

- Geräteklassen
- Persönlich zugeordnete Geräte
- Geräte mit Primärfunktion
- Digitale Artefakte

Benutzerschnittstelle

- Software
- Hardware

Ubiquitous Computing WS 03/04 Michael Beigl, TecO

3-18

Geräte

Geräte

- Gruppierung anhand verschiedener Kriterien
- aber: Axiome für Information Appliances gelten!

Klassen

- Anhand persönlicher Nutzung: Persönliche Geräte / Unpersönliche Geräte
- Anhand primärem Nutzungsbereich: Mobile Geräte
- Anhand Funktion: Digitale Artefakte

Geräte Persönliche Geräte

Charakteristik

- Sammlung sehr häufig gebrauchter Funktionen
- Vorhaltung persönlicher / sensibler Daten
- Intimer Gebrauch, immer dabei
- Als Eigentum betrachtet

Typische Anwendungen

- Kalender
- Adressen-, Telefonliste
- Notizen

Persönlicher Digitaler Assistent (PDA)

- Unterstützt Kalender, Adressenliste, Notizen
- Ist mit weiteren Anwendungen erweiterbar
- Kann Daten mit anderen Rechnersystemen austauschen und synchronisieren



Geräte Mobile Kommunikationsgeräte

Zentrieren sich um einen zentralen Verwendungsbereich

- Erweitern Nutzen um für den zentralen Verwendungsbereich interessanten Funktionen
 - Telefonliste
 - Notizen / SMS Speicher
- Sind (durch ihren Nutzen) extrem ansprechend, dadurch immer (ubiquitär) verfügbar
- Erweitern Nutzen mit Funktionen, die diesen Effekt ausnutzen
 - Wecker
 - Kalender
 - Voicerecorder / MP3 Player
 - Kamera
- **Wie ist hier die Affordance realisiert?**



Geräte Erweiterte Alltagsgegenstände

Erweiterung eines existierenden Gegenstandes

- eingebettetem Prozessor, Speicher, Sensorik und Netzwerkverbindung.
- Benutzerschnittstelle bleibt gleich oder ist wenig verändert
- Fähigkeit, digitale Information zu Erkennen und verarbeiten:
 - eigenständiger als einfache Objekte
 - nicht (notwendigerweise) „intelligent“ im KI-Sinne.
- Fähigkeit, Information mit anderen Geräten auszutauschen
 - nicht notwendigerweise online, aber zumindest gelegentliche Verbindung (Synchronisation)
 - Abgrenzung gegen abgeschlossene eingebettete Systeme

Geräte Beispiel: nicht digitale Objekte

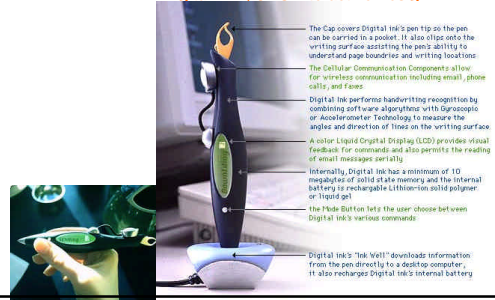
Integration nicht-digitaler Objekte

- Physischen Objekten eine Darstellung in der digitalen Welt geben
- Beispiel Mediacup:
 - Sensoren zur Erfassung des Zustands (Temperatur, Bewegung, Gewicht)
 - Prozessor zur Berechnung von Ereignissen („aufgefüllt“, „getrunken“, „abgekühlt“, ...)
 - Infrarot-Kommunikation, Broadcast in lokaler Umgebung
- **Wie wird hier Affordance genutzt?**



Beispiel Digitale Artefakte als Erweiterung bestehender Funktion

“Digital Ink”, CMU Interactive Design



Beispiel Digitale Artefakte als Ausgliederung von PC Funktionen

Eigenschaften

- In sich abgeschlossen, nur eigentliche Funktion wird nach außen kommuniziert
- Kombination bisheriger Einzelkomponenten
- Beispiel WebCam: Kamera und Rechner
- Beispiel DVD Player: DVD Laufwerk, Rechner, Audiokarte
- **Appliance Axiome?**



Ubiquitous Computing WS 03/04 Michael Beigl, TecO

3-25

Beispiel Digitale Photographie und Spiele

Digitale Kameras & Spiel



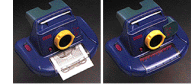
Bilderrahmen (am Internet)



Promera (Kamera+Projektor)



Creative Effects Camera s/w Bilder zum Ausmalen



Ubiquitous Computing WS 03/04 Michael Beigl, TecO

3-26

Geräte Appliances vs. non-Appliances

Kein Platz für den PC?

- Allroundtools manchmal nützlich, da physikalische Beschränkung
- Bsp. D. Norman: Bsp. Taschenmesser



- Bsp: Mikrowelle mit Internet?

- Mehr Infos: Want & Borriello: "Survey on Information Appliances"



Ubiquitous Computing WS 03/04 Michael Beigl, TecO

3-27

Geräte

Gestaltung: Affordances

Geräte

Software

Benutzerschnittstellen

Operating Systems

Hardware

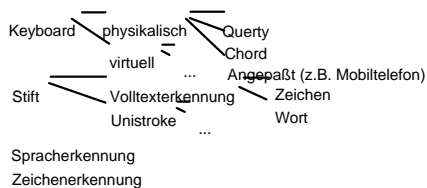
Ubiquitous Computing WS 03/04 Michael Beigl, TecO

3-28

Software Texteingabe

Eingabemöglichkeiten für Text in Geräte

- „Problem“ vieler Appliances: Klein & mobil
- Beispielhafter Auszug der wichtigsten Eingabemöglichkeiten:



Ubiquitous Computing WS 03/04 Michael Beigl, TecO

3-29

Software

Grob 2 Klassen

- Anwendungen
- Betriebssystem / Bibliotheken

Anwendungen

- sehr vielseitig von persönlichen Anwendungen
- ... über Standardanwendungen
- ... zu speziellen Anwendungen, insbesondere digitale Artefakte

Betriebssystem

- Vermittler zwischen Anwendungen und Hardware
- spezielle Betriebssysteme für spezielle Zwecke, z.B. persönliche Geräte, digitale Artefakte, Mobiltelefone
- Kriterium „am besten geeignet, statt „am umfangreichsten“ (PC)
 - klein und effizient statt vielfältig
 - harte Echtzeit statt „Performance“
- zu unterstützen: Benutzerschnittstelle, Kommunikation, Sensorik/Kontext, Energieverwaltung

Software Betriebssysteme

Microkernel oder API

- Unterstützt Kommunikation, Zugriff auf Sensorik, z.T. Scheduling
- Sehr Klein
 - Keine Prozesse, manchmal Threads
 - Wenige Variablen
 - Interrupt-gesteuert
 - Oft Kooperatives Scheduling
 - Problem: Hauptspeicher!

Betriebssystem- Beispiele

- Windows CE
- Palm OS
- EPOC
- Kommerzielle Real Time OS'e (RTOS), z.B. CMX für PIC, AVR...
- TinyOS (Berkeley) speziell für ad-hoc Netze
- BasicStamp ...

Ubiquitous Computing WS 03/04 Michael Beigl, TeCo

3-31

Betriebssystem

Aufgaben (vereinfacht)

- Task / Process / Thread Management
 - Inter-Process Communicaton
 - Memory Management
 - I/O Mangement (Netzwerk, Disk,...)
 - Device Management
 - Benutzerschnittstelle
 - „Middleware“ Dienste
 - Bootloading
- Handheld / Mobiltelefonbetriebssysteme**
- Windows Mobile, EPOC, PalmOS 5+, Embedded/Handheld Linux

Ubiquitous Computing WS 03/04 Michael Beigl, TeCo

3-32

Betriebssystem

Aufgaben (vereinfacht)

- Task / Process / Thread Management
- Inter-Process Communicaton
- Memory Management
- I/O Mangement (Netzwerk, Disk,...)
- Device Management
- Benutzerschnittstelle
- „Middleware“ Dienste
- Bootloading

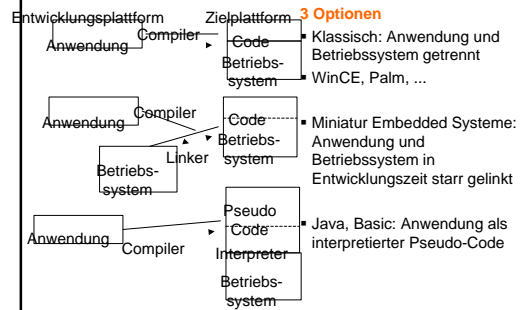
Microprozessoren

- keine oder minimal, oft co-operativ
- Semaphore, Queues (selten)
- oft dem Microprozessor überlassen
- kabellose Kommunikation, seriell, keine Disk etc!
- VIELE verschiedene Sensoren, viele Treiber
- wenig (LCD Treiber)
- „Middleware“ Dienste oft Perzeptionsalgorithmen
- KEIN Bootloading

Ubiquitous Computing WS 03/04 Michael Beigl, TeCo

3-33

Von der Anwendung zum Programm



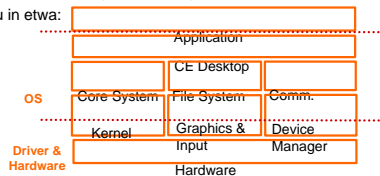
Ubiquitous Computing WS 03/04 Michael Beigl, TeCo

3-34

Software Betriebssystem Windows CE / Handheld Windows

Eigenschaften

- Für persönliche Multimediageräte mit „Windows“ Logik
- pre-emptive Multithreading, Semaphore, prorisierte Interrupts...
- „etwas“ Realtime: Latencies Task-Switch, Memory Switch...
- Minimal 500 kB ROM, 35 kB RAM, Normal: 2 MB RAM
- Aufbau in etwa:



Ubiquitous Computing WS 03/04 Michael Beigl, TeCo

3-35

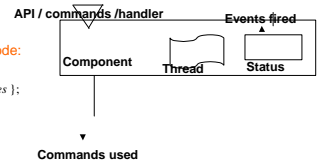
Software Betriebssystem Tiny OS

Tiny OS (Berkeley) für SmartDust / MOTS

- cooperatives Multitasking, kein Speicherschutz
- Ereignisbasierter Wechsel, FIFO-Scheduler + priorisierte Ereignisse (Netzwerk)
- Vor allem für ad-hoc Sensornetzwerke
- Komponente:
- Bsp:

Deklarationsdatei zus. zu C-Code:

```
TOS_MODULE name;
ACCEPTS{ command_signatures };
HANDLES{event_signatures};
USES{ command_signatures};
SIGNALS{event_signatures};
```



Ubiquitous Computing WS 03/04 Michael Beigl, TeCo

3-36

Übung

Beispiele fotografieren

- Wo tatsächliche und empfunden Eigenschaften differieren (Affordances)
- Wo Similarity, Closure, Proximity, Simplicity nützlich sind