

Ubiquitous Computing

(Ubiquitäre Informationstechnologien)

Vorlesung im WS 01/02



Prof. Lars Wolf

Michael Beigl

Universität Karlsruhe

Institut für Telematik

Telecooperation Office

www.teco.uni-karlsruhe.de

Aufbau der Vorlesung

1 Grundlagen

2 Geräte

Grundlagen

Internet

Wearable

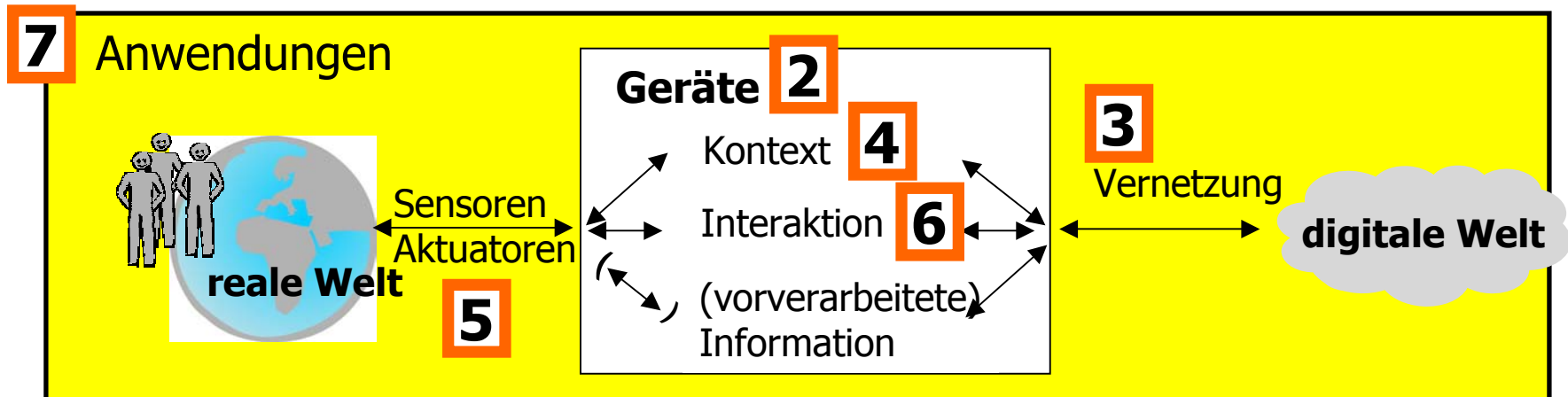
3 Vernetzung

4 Kontext

5 Sensoren/Aktuatoren

6 Interaktion

7 Anwendungen



Geräte: Grundlagen

- **Definition**
- **Entwicklungstrends**
- **Architektur**
- **Software**
- **Hardware**
- **Energieversorgung**

Fragestellung

Hauptmerkmal der Post-PC-Generation: Diversifikation

- Geräte für unterschiedliche Anwendungen (“**Inform. Appliances**”)
- Diversifikation bzgl. Rechenleistung, Kommunikation, Form-Faktor, Betriebssystem, User Interface, Schnittstelle zur realen Welt,...
- Computer so verschieden wie Alltagsgegenstände

Architektur und Klassifikation von Geräten notwendig

- Wo liegen die Unterschiede zu herkömmlichen Systemen
- Was sind Merkmale zur Unterscheidung verschiedener Geräte

Definition

Abgrenzung gegen Universalrechner

- **Leistungs- / Funktionsumfang**

- mehr Leistung/Funktionalität als in nicht-digitalen Objekten
- weniger Leistung/Funktionalität als in Universalrechnern

- **Interaktion**

- eingebettete Digitale Artefakte: keine direkte Benutzerinteraktion
- interaktive Digitale Artefakte: einfachere Benutzerschnittstellen als bei Universalrechnern

- **Kontext**

- viele Digitalen Artefakte zeichnen sich durch Bezugnahme auf (und Interaktion mit) physischer Umgebung aus (im Gegensatz zu Standardrechnern, die von ihrer Umg. abstrahieren)

Definition

Ein „**Digitales Artefakt**“ ist ein **physisches Objekt** mit eingebettetem **Prozessor, Speicher, Sensorik und Netzwerkverbindung**.

- Viele „Digitalen Artefakte“ haben auch eine Benutzerschnittstelle und/oder die Fähigkeit, mit ihrer physischen Umgebung zu interagieren.

Minimalanforderungen

- „Digital“: bezieht sich auf die 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: Grundlagen

- **Definition**
- **Entwicklungstrends**
- **Architektur**
- **Software**
- **Hardware**
- **Energieversorgung**

Entwicklungstrends

Woher kommt der Bedarf ?

Interaktiv

- **Informationszugang:** Zugangsgeräte für verschiedene Zielgruppen und Situationen; „Internet-Appliances“
- **Konvergenz:** Integration von Geräten aus Telekommunikation und Unterhaltungselektronik; „Multimedia-Geräte“
- **Mobilität:** für Mobilität (ständige Begleitung) optimierte Geräte
- **Spezialisierung:** Norman's „Information Appliances“

Eingebettet

- **Flexibilität:** „Gleichstellung“ von Peripherie und Komponenten
- **Automatisierung:** Erweiterung von Geräten/Maschinen um „Smart Control“ für Fernwartung und -steuerung
- **„Informatisierung“:** Integration nicht-digitaler Geräte/Objekte

Informationszugang

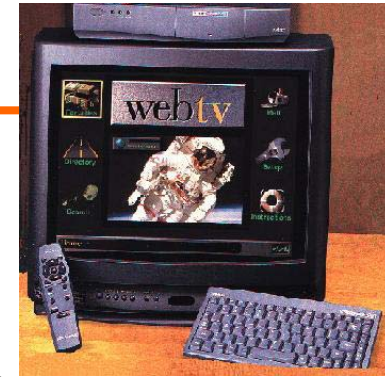
Zugang für alle

- Informationsgesellschaft, Teledemokratie: Zugang in jedem Haushalt
- v.a. auch Zugang ohne Computer-Kenntnisse
- preisgünstiger Zugang
- Integration in bekannte Geräte (TV, Telefon)

Zugang nicht nur am Schreibtisch

- Mobiler Zugang
- Bequemer Zugang („auf dem Sofa“)

WebTV



Screenphone



WAP Handy



Surfpads

Konvergenz

Geräte aus anderen Bereichen finden Anschluß

- Geräte aus Unterhaltungselektronik und Telekommunikation integrieren Computeranwendungen (insbes. Informationszugang)
- Spielkonsolen, Fernseher, VCR, Telefone, ...



Sega Dreamcast
Spielkonsole mit
Browser, Email

Konvergenz in neuen Geräte

- Bündelung konvergierender Anwendungen in neuen Geräteklassen
- Screenphones, Internet-Handys, Organizer-Handys, Interactive TV



Mobilität

„Information Anytime, Anywhere“

- Organizer, Personal Digital Assistants (PDA), Handheld PC, Wearable Computer
- einfache Synchronisation, zunehmend auch Mobilkommunikation

Mobilkommunikation

- GSM-Telefone: ubiquitäre Plattform
- WAP, i-mode: Internet-Integration
- UMTS: Infrastruktur für smart mobile devices



npl



TecO



Psion

Flexibilität

Aus Komponenten werden Geräte

- Software-Anwendungen „out of the box“, z.B. Web-Server
- Hardware: statt Einschubkarte Plug&Play-Gerät, z.B. Speicher
- Peripherie: Gleichstellung im Netz, direkt ansprechbar

Flexibilität durch Unabhängigkeit von Hosts

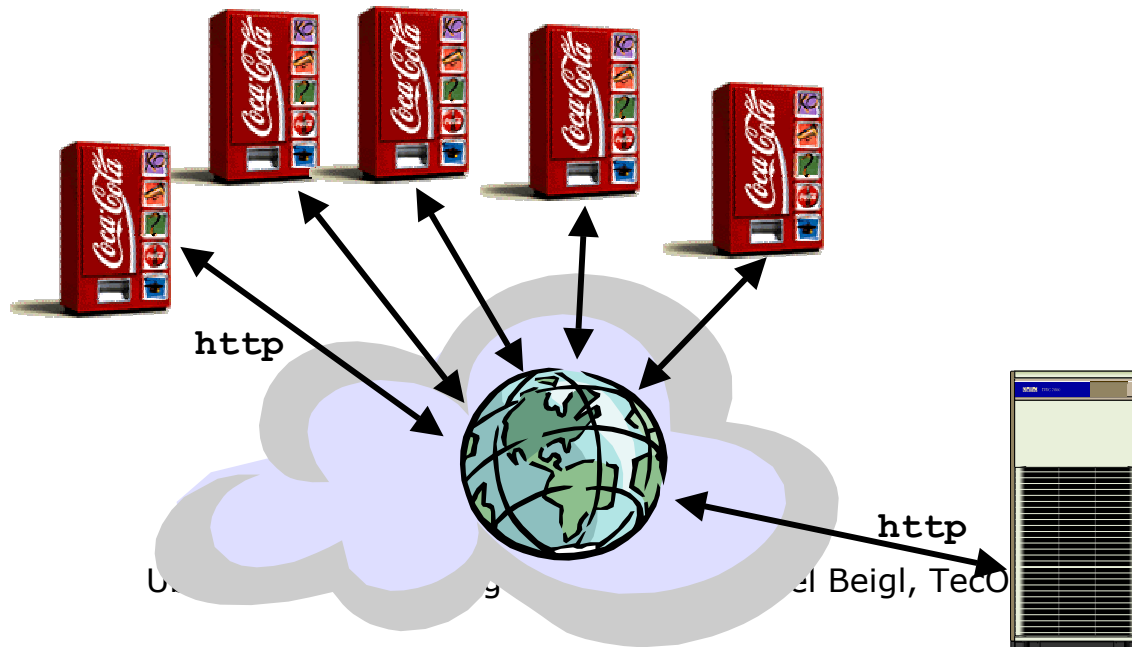
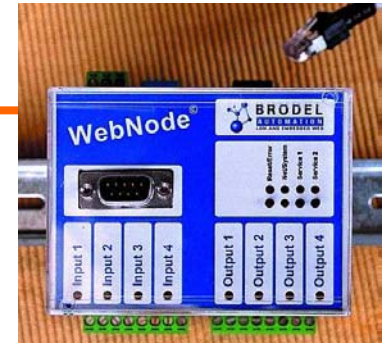
- flexible Anbindung an andere Geräte, Plug&Play



Automatisierung

Smart Control

- Anbindung von eingebetteten Systemen an die Informationswelt
- Fernbedienung, Fernwartung
- Haushaltsgeräte, Facility Management, Automatenindustrie



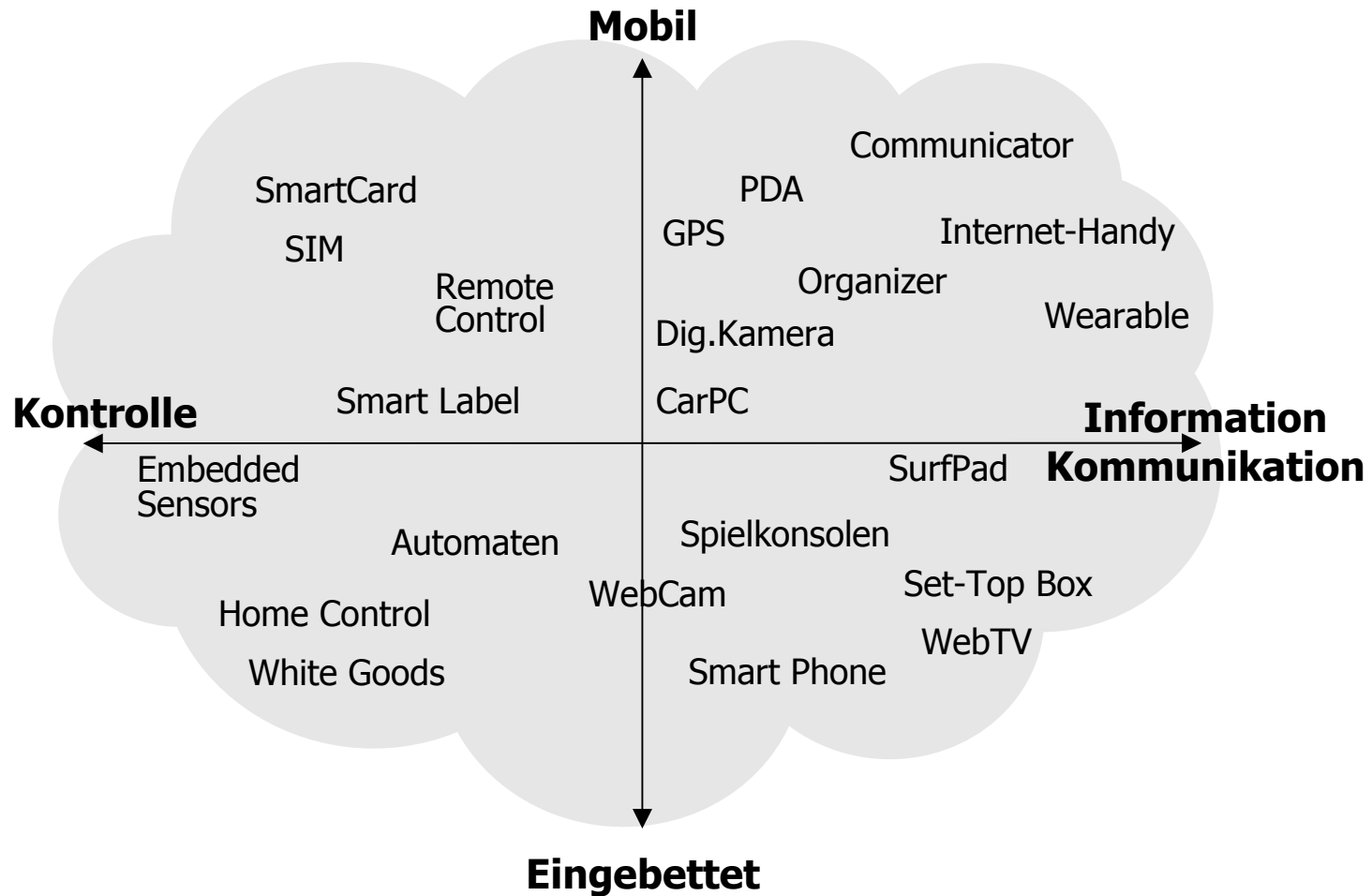
Informatisierung

Integration nicht-digitaler Geräte/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



Klassifikation

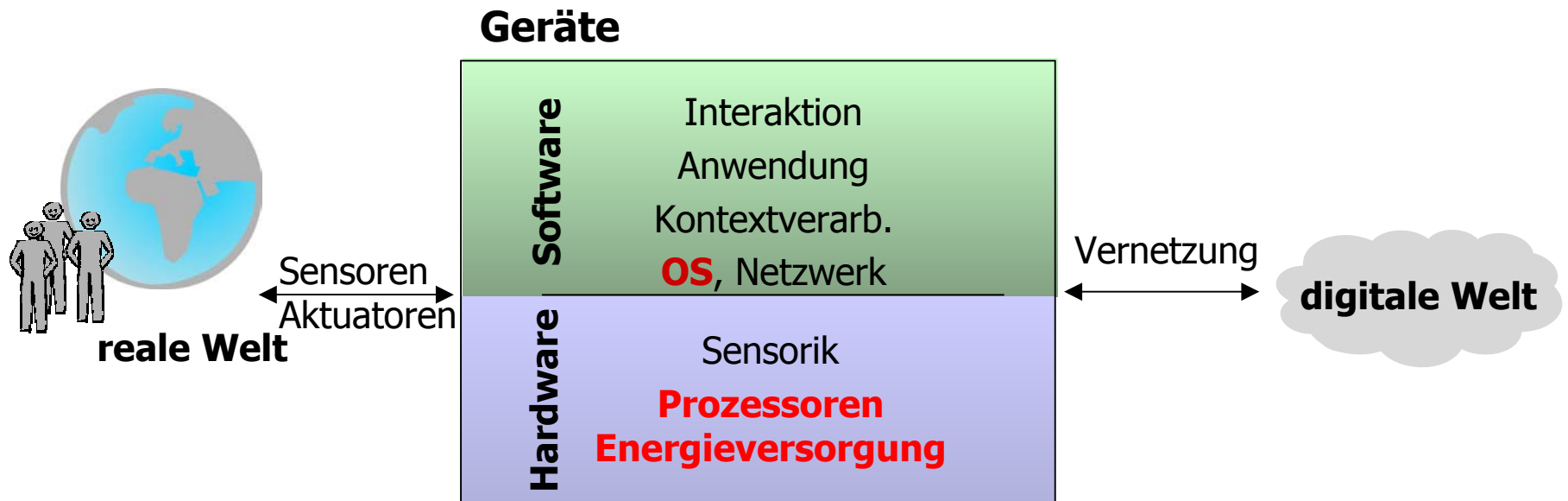


Geräte: Grundlagen

- **Definition**
- **Entwicklungstrends**
- **Architektur**
- **Software**
- **Hardware**
- **Energieversorgung**

Modell zur Charakterisierung

- Unterschiedliche Systeme mit unterschiedlichen Ausprägungen: Bereiche können als eigene Schichten ausgeprägt oder zusammengefaßt sein
- Qualitative größte Gruppe: Zusammengefaßte Software- bzw. Hardware-schichten
- → direkte Reflektion von Entwurfsentscheidungen auf Umwelt / Benutzer, Technologie und HCI wichtig!



Merkmale Charakterisierung Digitale Artefakte

Äußere Merkmale:

Interaktion

- Bandbreite: keine Interaktion ... Text/Grafik ... Multimedia/multimodal
- Häufigkeit: nie ... gelegentlich ... regelmäßig ... permanent
- Interaktionsdauer: kurze oder lange „Sitzungen“
- Wahrnehmung: peripher ... zentral

Kommunikation

- offline vs. online, lokal vs. global
- Häufigkeit, Verbindungsdauer, Bandbreite

Kontext

- eingebettete Systeme: Interaktion mit Host
- weitergehende Wahrnehmung der Umwelt (Lokation, Situation)

Geräte: Grundlagen

- **Definition**
- **Entwicklungstrends**
- **Architektur**
- **Software**
- **Hardware**
- **Energieversorgung**

Software

Betriebssysteme

- im Universalrechnerbereich sehr umfangreich
- Hauptaufgaben: Betriebsmittelvergabe zwischen vielen Prozessen, Hardware-Abstraktion

Betrieb von Digitalen Artefakten

- wesentlich differenzierter
- Unterschiede in zu verwaltenden Betriebsmittel, z.B. i.allg. kein externer Speicher
- oft Realzeit-Anforderungen (Steuerung, Multimedia), dabei begrenzte Rechenpower (d.h. best effort nicht akzeptabel)
- i.allg. nur wenige Anwendungen/Prozesse

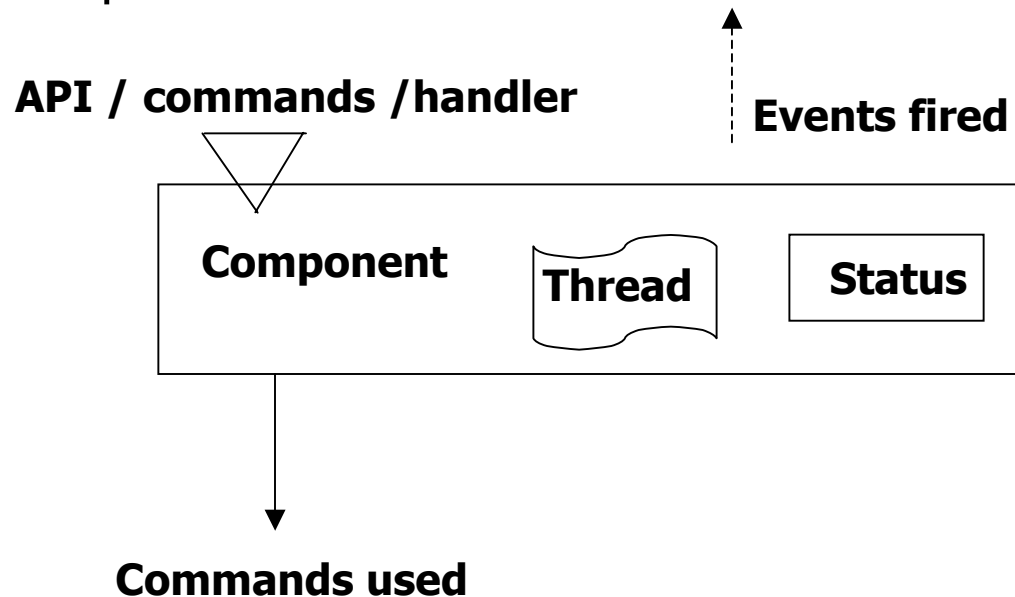
Software: Betriebssysteme

- **Microkernel / API**
- **Unterstützt Kommunikation, Zugriff auf Sensorik, z.T. Scheduling**
- **Sehr Klein**
 - Keine Prozesse, manchmal Threads
 - Wenige Variablen
 - Interrupt-gesteuert
 - Kooperatives Scheduling
 - Problem: Hauptspeicher!
- **Betriebssystem- Beispiele**
 - RTOS (Real Time OS, z.B. CMX für PIC)
 - TinyOS (Berkeley)
 - L4
 - BasicStamp

Software Betriebssystem Tiny OS

- **Tiny OS (Berkeley):**

- 2 Taskwechsel alle 50 μ s
- Ereignisbasierter Wechsel, FIFO-Scheduler + priorisierte Ereignisse (Netzwerk)
- Komponente:



Software

Java

Java Programmierung für Digitale Artefakte

- Einfache, sichere Sprache
- Kompakter Bytecode zur Ausführung auch auf kleinen Plattformen
- Portabilität: Ausführung von Bytecode auf virtuellem Prozessor (JVM, Java Virtual Machine)
- Probleme: begrenzte Plattformunabhängigkeit der JVM; Annahmen über Speicher, Dateisystem, Peripherieanbindung die auf Digitalen Artefakte oft nicht zutreffen

Java 2 Familie

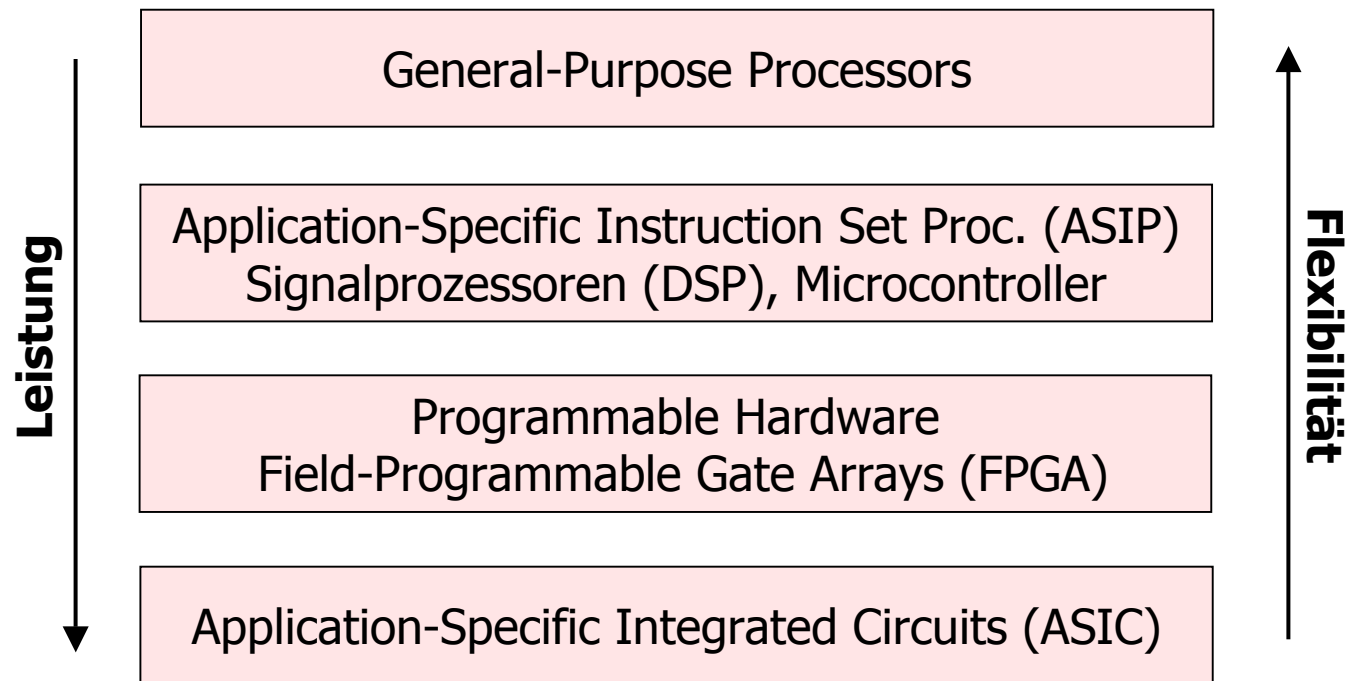
- Unterschiedliche Editionen für Spektrum von Plattformen
- MicroEdition (J2ME) für Information Appliances, minimaler Funktionsumfang, dynamisch nachladbare Erweiterungen
- Verschiedene Profile in J2ME: Car, TV, PDA, Phone

Geräte: Grundlagen

- **Definition**
- **Entwicklungstrends**
- **Architektur**
- **Software**
- **Hardware**
- **Energieversorgung**

Hardware

Implementierungsarten



Hardware Implementierungsarten

General-Purpose Prozessor

- hohe Performance durch fest verdrahtete hochoptimierte Funktionseinh.
- Flexibilität durch Ausführung externer Programme
- große Leistungsaufnahme

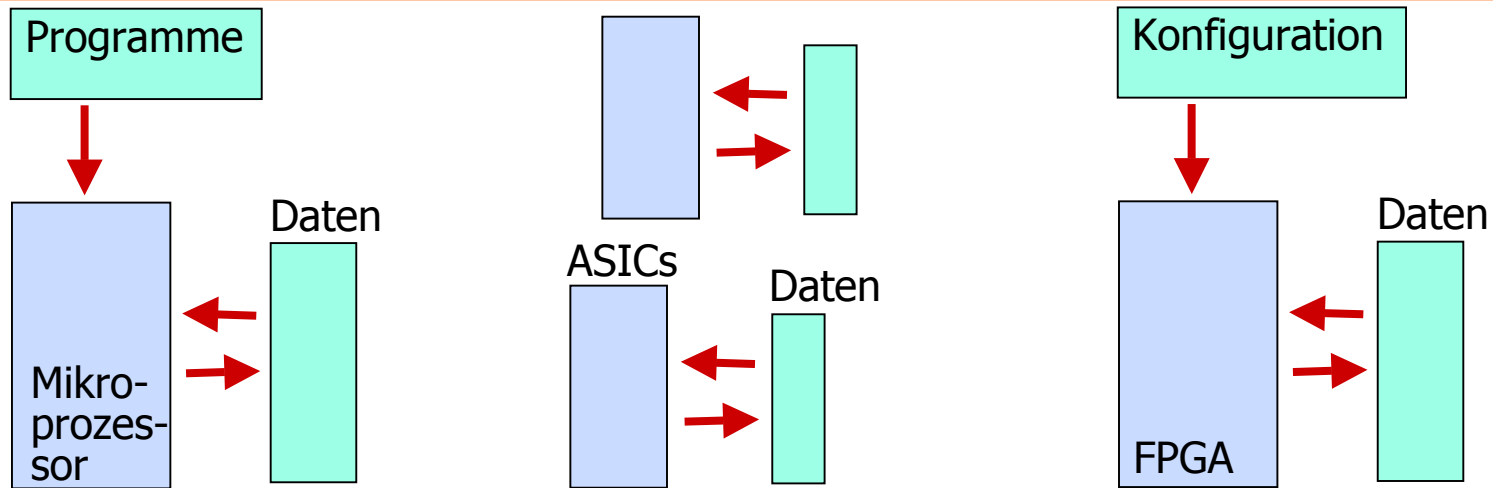
ASIP

- Anwendungsspezifische Spezialisierung
 - des Instruktionssatzes (z.B. Operatorverkettung)
 - der Funktionseinheiten (z.B. Pixel-Operationen)
 - der Speicherarchitektur (z.B. Parallelzugriff)
- höhere Performance, niedrigere Kosten (kl. Chipfläche), geringere Codegröße, geringere Leistungsaufnahme

Microcontroller

- spezialisiert für steuerungsdominante Anwendungen
 - Bit- und Logikoperationen statt Arithmetik
 - Register in RAM: Kontextswitch durch Pointeroperation
 - Integration von Peripherie: A/D, D/A, Timer, ...

Hardware Implementierungsarten



General-Purpose

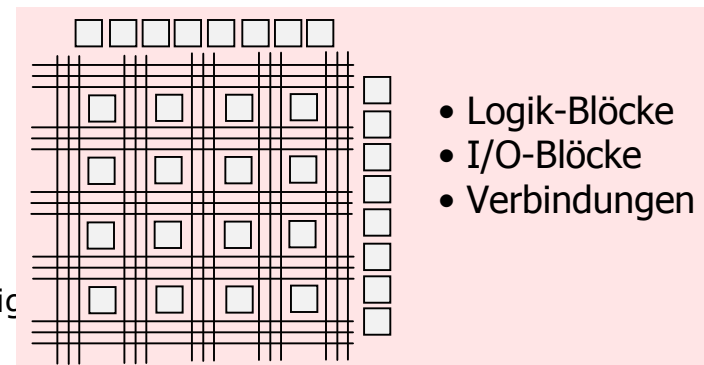
Vielseitigkeit durch Laden extern gehaltener Programme

ASIC

Anwendung bereits durch Verdrahtung festgelegt

FPGA

programmierbare interne Konfiguration sonst ASIC-Verhalten



Superkompatible Hardware

Optimierungsziel / Kaufkriterien

- bisher: Verhältnis von Performance zu Kosten, durch Minimierung der Chipfläche
- an Bedeutung gewinnende Kriterien: niedrige Verlustleistung (cool chips are hot, s.u.) und Kompatibilität

Superkompatible Prozessoren

- Kostenoptimierung durch hohe Stückzahl
- Superkompatibilität
 - direkte Unterstützung einer weitverbreiteten Sprache
 - höchste Performance für diese Sprache
 - akzeptable Performance für andere Sprachen
 - geringe Chipfläche, somit niedrige Kosten
- für anspruchsvolle, vernetzte eingebettete Systeme

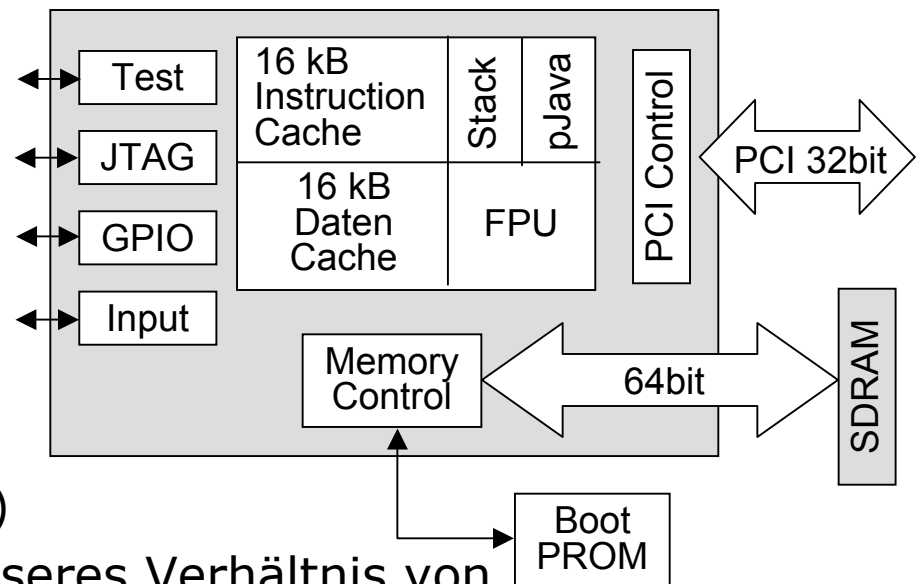
Superkompatible Hardware

picoJava Architektur: „JVM in Hardware“

- Java-Bytecode wird unmittelbar als Maschinencode verarbeitet
- komplexe Instruktionen erzeugen Trap und werden in Software emuliert
- kann aus anderen Sprachen generierten Binärcode ausführen

MicroJava 701 Prozessor

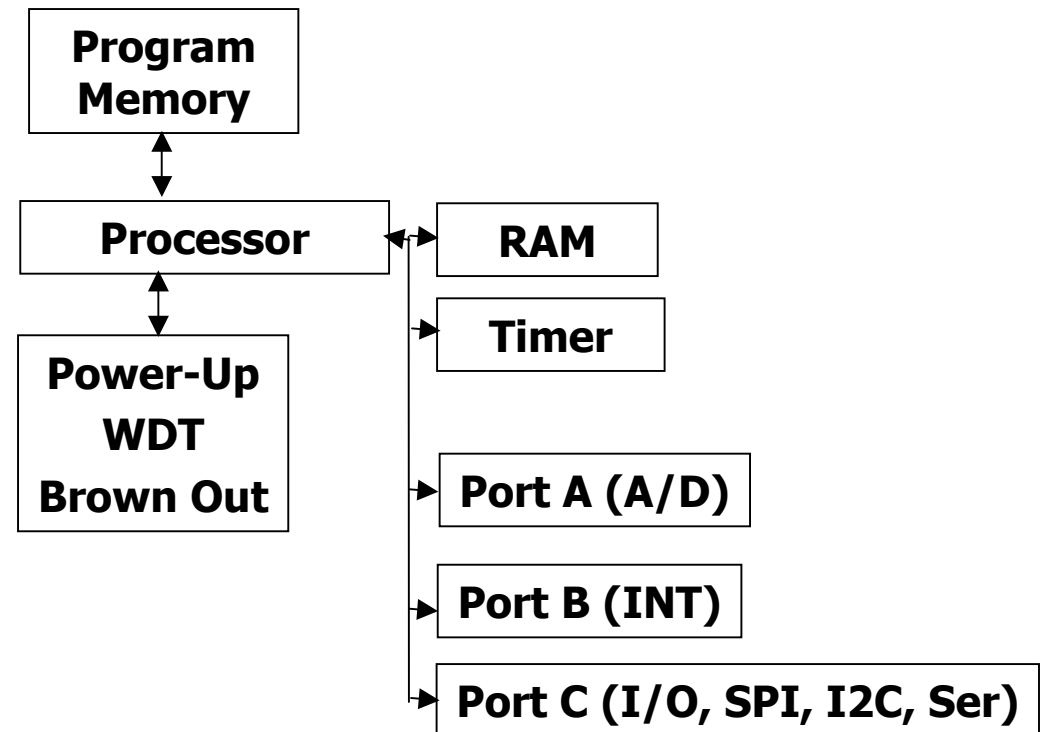
- 32-bit-RISC-Prozessor, 200 MHz
- Vergleich mit Intel-PII-300
 - nur 2.8 Mio. Transistoren, kleinere Chipfläche (50mm²)
 - für Ausführung von Java besseres Verhältnis von Performance zu Kosten



Angepaßte Hardware

z.B. PIC 16F876

- Low Cost, Low Power
- Einfache interne Rechenstruktur, komplexe I/O (viel I/O → viel Kontext)
- All in one chip: (z.B. A/D, Power-Up)
- (fast) keine ext. Bauteile notwendig
- Programmierung: C, Assembler



Häufig verwendete Hardware

Typen

- Arizona Microchip PIC 16xxx, 17xxx, 18xxx (RISC) Reihe
- Atmel 8051 sowie RISC Architektur (und weitere 8051er Derivate)
- Hitachi H8/3294 (RCX)
- 68000 – Derivate (insb. Information Appliances)
- 80x86 Derivate (Steuerung, PC 104, embedded Web-Server)
- ARM, StrongARM: Info-Appliances

Leistungsparameter

- RAM: von <16 byte bis mehrere MB
- ROM: ab 1 k byte
- Geschwindigkeit: 32 kHz bis >200 MHz (dual Mode)
- On Board I/O, oft A/D, PWM, D/A, Treiberstufen
- Schnittstellen: Seriell, I2C, SPI, CAN

Hardware

Vergleich Prozessoren

Typ	Pro-gramm Speicher (Byte)	Daten Speicher (Byte)	max Takt MHz	I/O	Ver-brauch Max/typ	Größe (mm)
16F84A	1792	68	10	13	2mA/20 uA	1,1x1
16F877 (16F876)	14336	368	20	33 (22)	2mA/30 uA	2,3x2,3 (1,1x2,3)
16F828	3584	224	20	16	2mA/30 uA	1,1x1
Beck-IPC	512k	512k	20	8	200mA/100mA	40x50(M odul)
MN10L200 (OnHandpc)	2 MB	128k	10	-	-.	-
NC68EZ328 (3Com PalmPilot)	2-8 MB	2-8 MB	16	-	-	-
Intel-kompatible (>i486)	Typ.: minimum 32 MB	Typ.: minimum 32 MB	> 66	-	>1A	70x50 (Modul)

Geräte: Grundlagen

- **Definition**
- **Entwicklungstrends**
- **Architektur**
- **Software**
- **Hardware**
- **Energieversorgung**

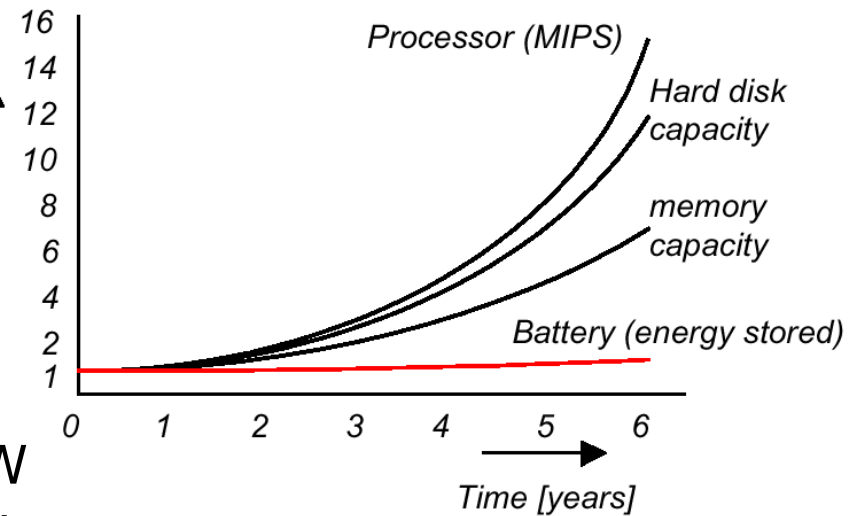
Energieversorgung

Batterietechnologie

- langsamer Fortschritt (nur 20% mehr Kapazität in 10 Jahren)
- Hoffnungsträger: Methanol-Brennstoffzellen

Energieverbrauch

- Desktop-Rechner $\sim 10^2 W$
- Laptop $\sim 10 W$
- Single-Board Comp. $\sim 1 W$
- Low-power Microcontr. $\sim 10^{-3} W$



3 Möglichkeiten der Energieversorgung

- Energiesparen
- Unbemerkte Energiezufuhr
- Benutzergesteuerte Energiezufuhr

Energieversorgung

Energiesparen

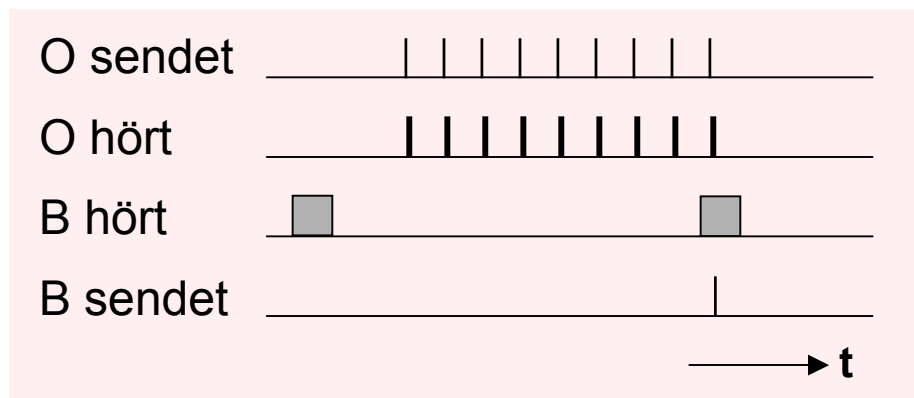
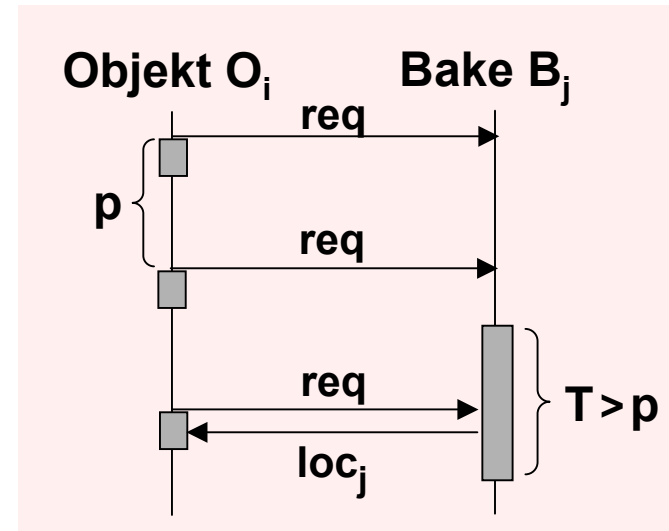
Energiesparen

- **Technologie-Ebene (CMOS):** Kapazität minimieren, minimale Chipfläche, möglichst wenig externe Verbindungen
- **Hardwarearchitektur:** hohe Verlustleistung in Bussen, Treibern, Multiplexern → einsparen durch Lokalisierung von Operationen
 - anwendungsspezifische Co-Prozessoren (am Prozessor vorbei, weniger Verkehr auf dem Bus)
 - Cachehierarchien (häufige Operationen in der Nähe des Prozessors gespeichert)
- **Kommunikation:** Minimierung der Zeit, die die Netzwerkschnittstelle aktiv ist: Energie-bewußte Zugangsprotokolle
 - z.B. Traffic Scheduling in der Basisstation und periodische Bekanntgabe der Schedules: wer nicht aufgelistet ist, kann sich eine Periode lang abschalten

Energieversorgung Energiesparen

Beispiel Infrarot-Lokationssystem

- Dienstnehmer sendet Anfragesignal
- Bake antwortet mit Ortsinformation
- Energiesparendes Protokoll: Minimierung von Sende- und Empfangszeiten
- autonome Energieversorgung der Baken (Solarzellen und Kondensatoren)

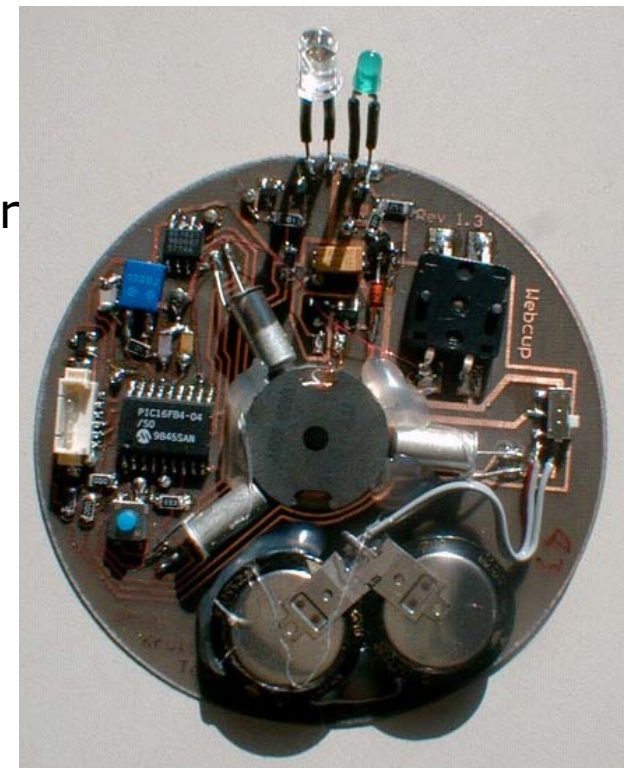


Energieversorgung

Unbemerkte Energiezufuhr

Beispiel Mediacup

- Prozessor
 - reduzierte Taktrate 1 MHz
 - Schlafmodus so oft wie möglich
- Bewegungsmessung
 - Kugelschalter statt Beschleunigungssensor
 - ermöglicht Interrupts statt Polling
 - keine Bewegung → keine Messung
→ kein Stromfluß
- Berührungsloses Aufladen
 - Via Induktion
 - Kondensatoren, 2F
 - kein Batteriewechsel

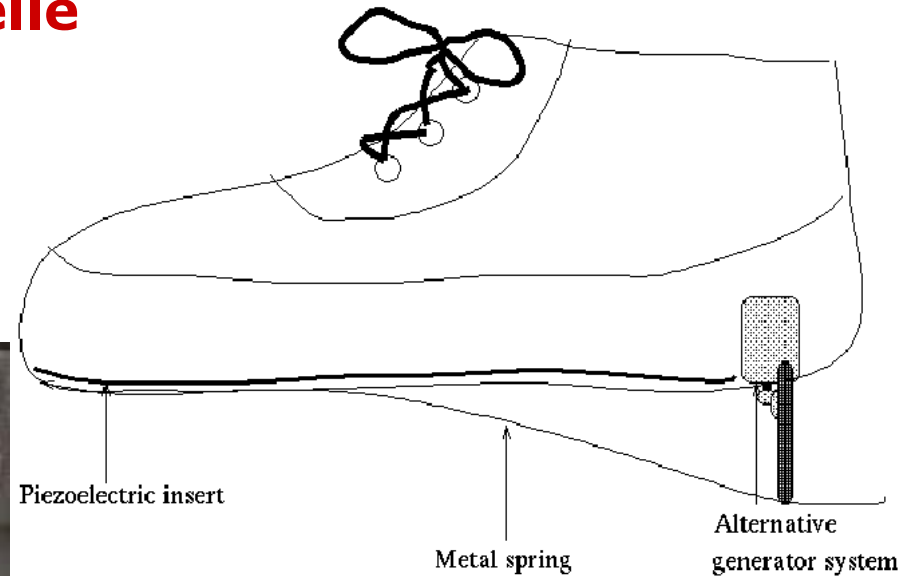


Energieversorgung

Unbemerkte Energiezufuhr

Der Mensch als Energiequelle

- „Kraftwerk im Schuh“
- Joe Paradiso



Energieproduktion ist ein Kontext:



Ubiquitous Computing WS 01/02 Michael Beigl, TecO