

Eingebettete Systeme

Prinzipien und Komponenten Eingebetteter Computer Systeme PKES

Wintersemester 2006/2007

Prof. Dr. Jörg Kaiser
Embedded Systems and Operating Systems (EOS)



Allgemeine Information

Dozent:

Prof. Dr. Jörg Kaiser
Institut für Verteilte Systeme (IVS)
Arbeitsgruppe Eingebettete Systeme und Betriebssysteme
Geb. 29 Zimmer 323
kaiser@ivs.cs.uni-magdeburg.de

Sekretariat:

Petra Duckstein
29 Zimmer 405
duckstein@ivs.cs.uni-magdeburg.de
67 18345

Übungsgruppenleiter:

Michael Schulze, Sebastian Zug
Institut für Verteilte Systeme (IVS)
Arbeitsgruppe Eingebettete Systeme und Betriebssysteme (EOS)
{zug, mschulze}@ivs.cs.uni-magdeburg.de



Inhalt

- ➔ Einführung: Was ist ein eingebettetes System?
Was macht den Unterschied aus?
Was muss man können?
- ➔ Sensoren, Aktoren und ihre
Unterstützung durch Funktionseinheiten
in Micro-Controllern
Analoge Schnittstellen
Zeitgeber und Zähler
Unterbrechungsbearbeitung
- ➔ Zuverlässigkeit und Fehlertoleranz
- ➔ Zeitgerechte Ausführung
- ➔ Betriebssysteme für eingebettete Systeme



web Ressourcen

http://ivs.cs.uni-magdeburg.de/eos/lehre/WS0607/vl_pkes/



Schein- und Prüfungsleistungen

- **Unbenoteter Schein:**
 - Teilnahme an den Übungen**
 - Lösung der Übungsaufgaben (alle Übungsblätter mit > 60%)**
 - Präsentation der Übungsaufgaben in den Übungen**
- **Prüfung**
 - Zulassung: Kriterien unbenoteter Schein erfüllt**
 - Durchführung: Klausur, bei weniger als 15 zu Prüfenden mündliche Prüfung**
- **Anmeldung erforderlich**
 - Details in den Übungen erfragen**



Empfehlung für Vorlesung und die Übungen:

Erzähle mir und ich vergesse.
Zeige mir und ich erinnere.
Laß mich tun und ich verstehe.

Konfuzius, 551 – 479 v.C.



was ist ..? ein erster Definitionsversuch

Eingebettetes System

Ein Artefakt mit einer informationsverarbeitenden Komponente, welche die Funktionen des Artefakt über eine gewohnte oder angepasste Benutzerschnittstelle unterstützt und erweitert.

oder

Ein Computer, der als (unsichtbarer) Bestandteil eines Geräts gekauft wird.

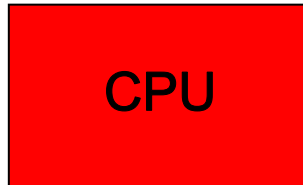
ein eingebettetes System:

- erfüllt eine spezielle Aufgabe.
- hat spezielle, problemangepasste Hardware und Software.
- hat spezielle, problemangepasste Benutzerschnittstelle.



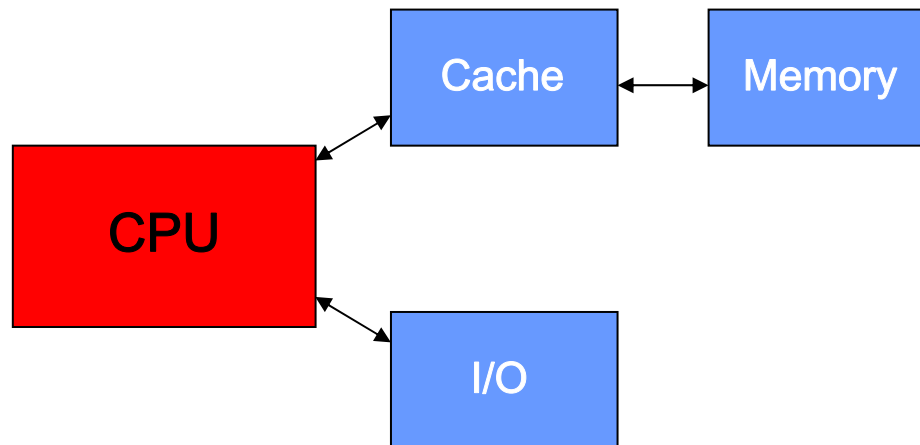
Systemsicht eines Computerarchitekten:

Leistungs-Eigenschaften gemessen in: **Performance**



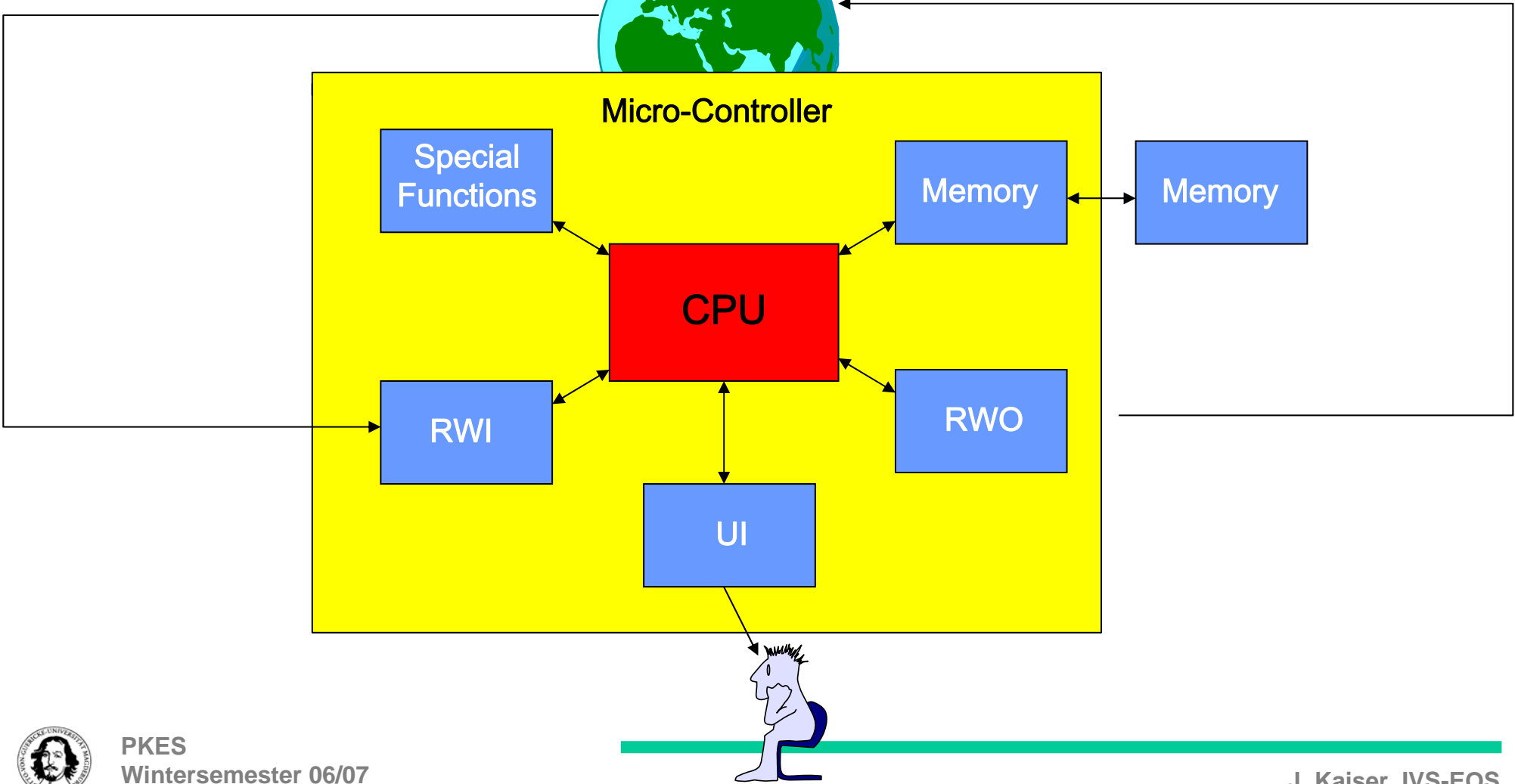
Systemsicht eines weitdenkenden Computerarchitekten:

Leistungs-Eigenschaften gemessen in: **Performance und Kosten**



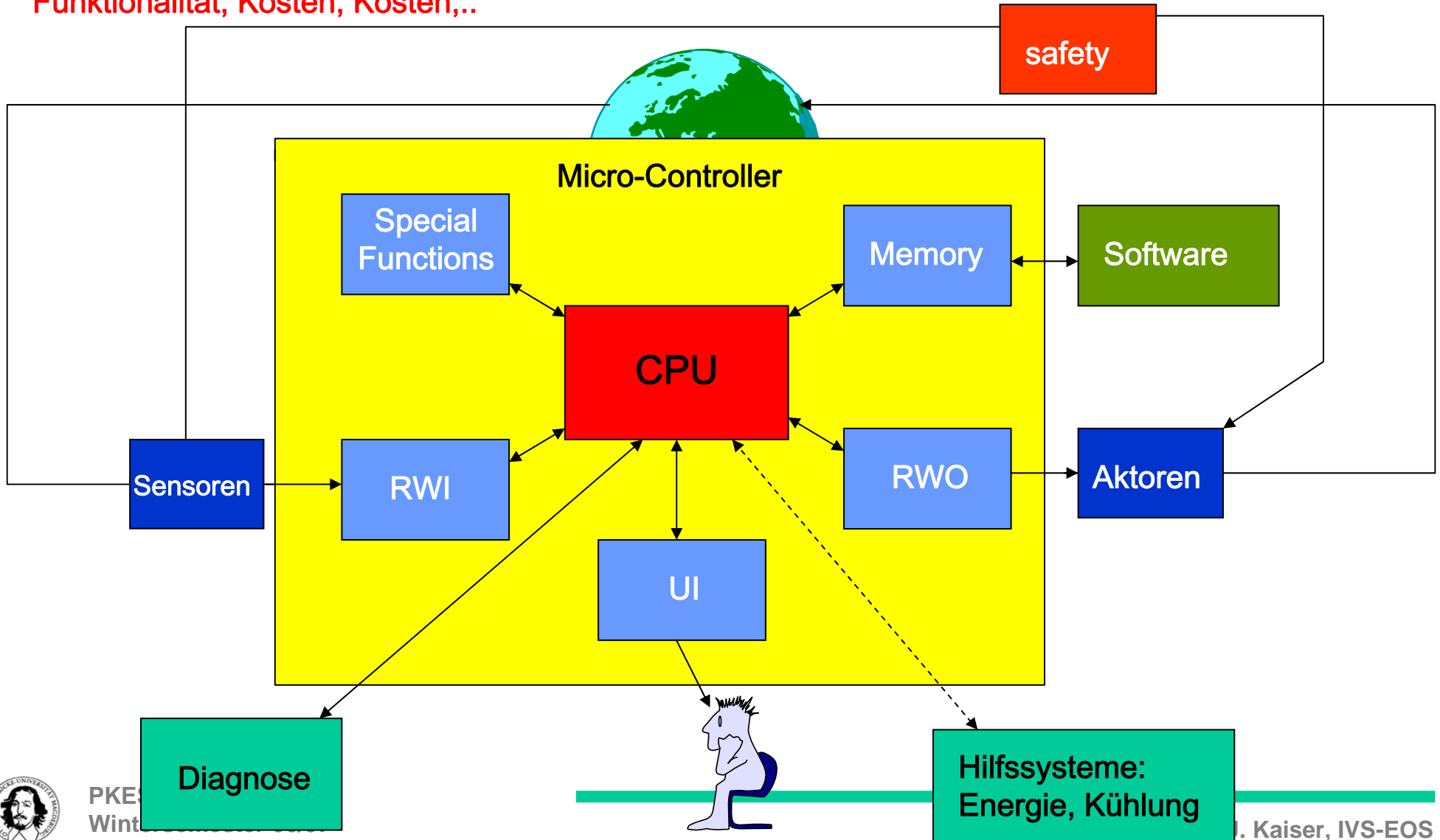
Systemsicht eines Architekten für eingebettete Systeme:

Leistungs-Eigenschaften gemessen in:
Kosten, I/O-Möglichkeiten, Speichergröße,
Performance

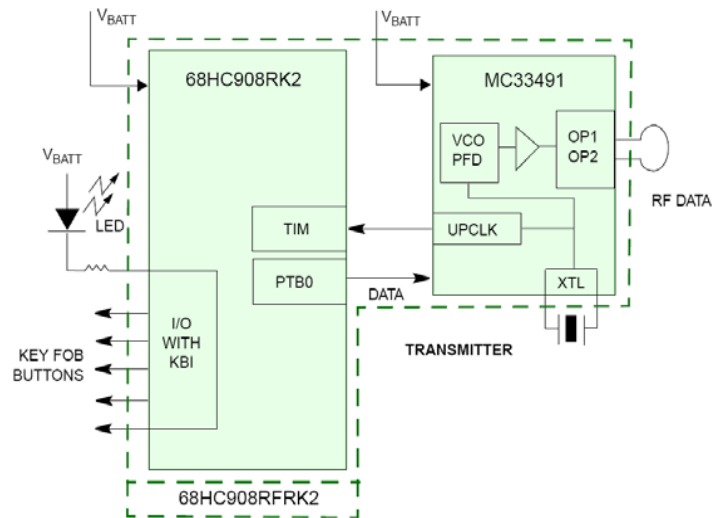


Systemansicht eines Architekten für Kontrollsysteme:

- Leistungseigenschaften gemessen in: **Kosten, Time-to-Market, Kosten, Funktionalität, Kosten, Kosten,...**



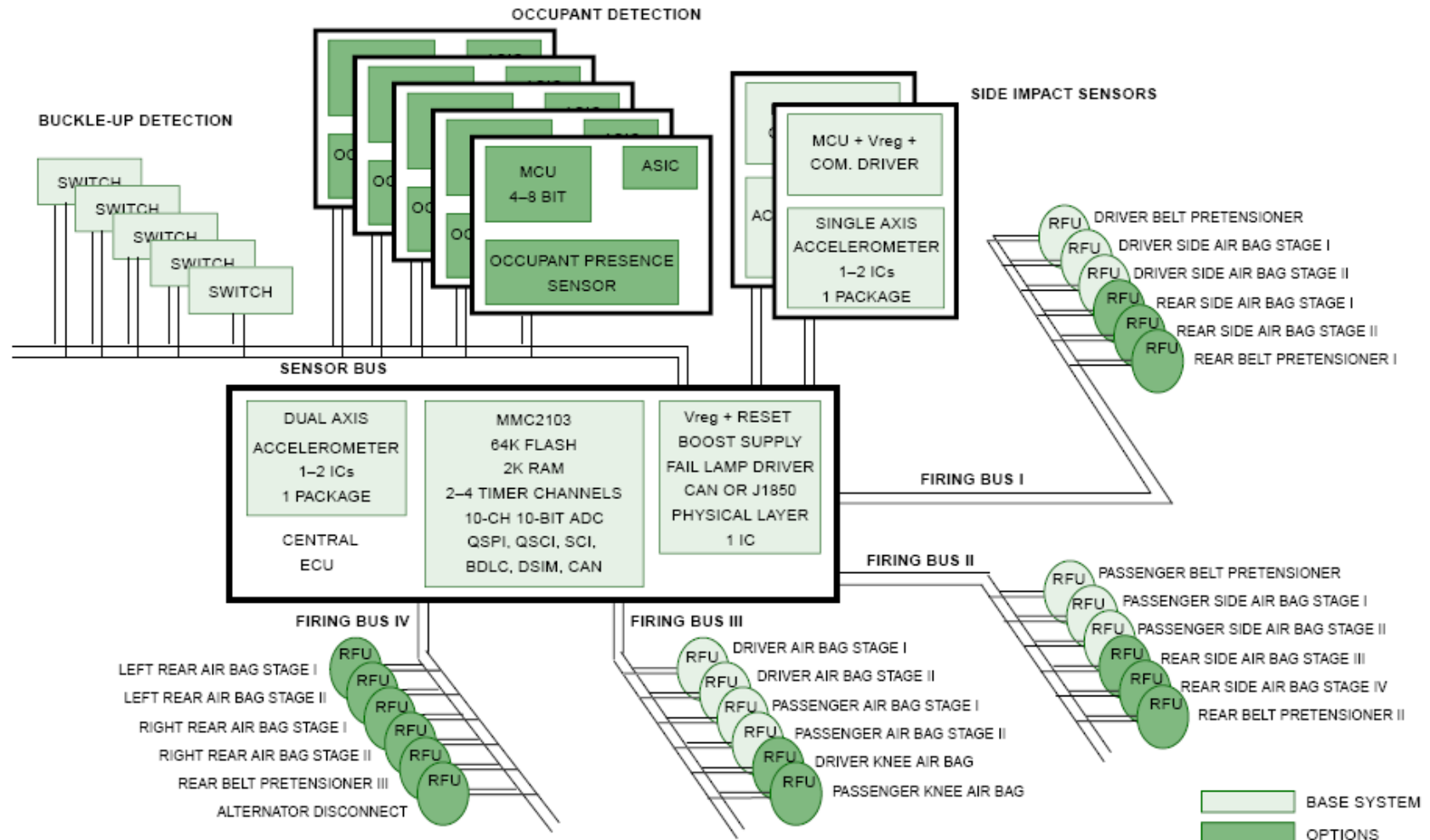
Beispiele eingebetteter Systeme: drahtloser Türöffner



Performance ~ 100KIPS,
Speicher <1K,
"handgestrickte" Software,
5 Jahre Batterielevensdauer einer Knopfzelle

Mehr Beispiele für Steuerungsaufgaben in:
 Aufzügen, Turbinen, Flugzeugen, Autos, industrielle Automatisierung:
 Sicherheitskritische Systeme, Echtzeitverarbeitung, verteilte Prozessoren, hohe Komplexität.

Beispiel: verteilte Airbag Auslösung

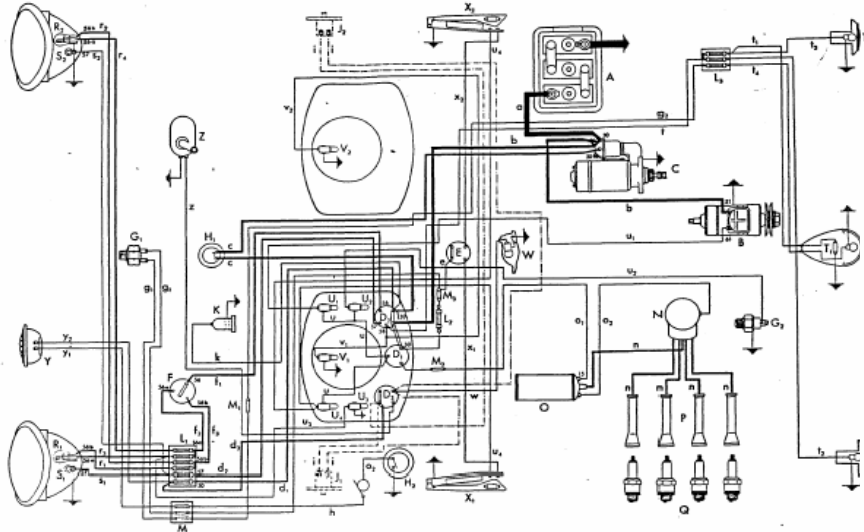


vom notwendigen Randbereich:

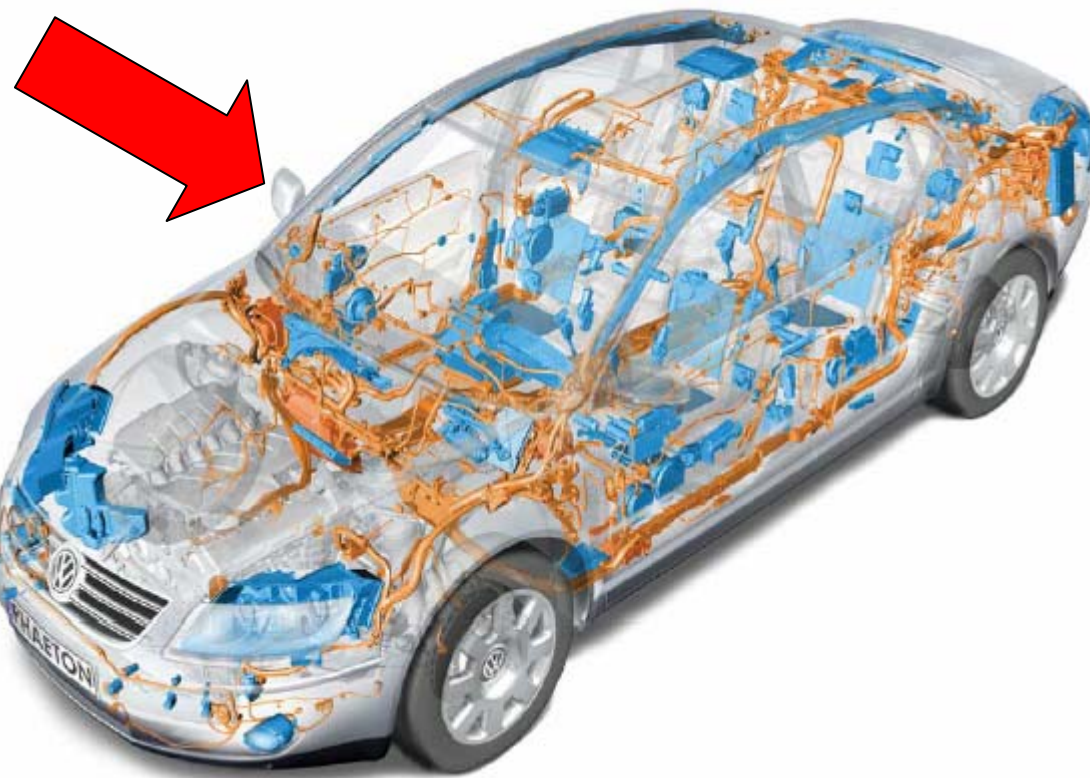


zur zentralen Problemstellung.

Elektrischer Schaltplan (Volkswagen)



| KABELSCHÜSSEL | | | | | | | | | | | |
|--|----------------------------------|---|---|--|---------------------------------|--|--|--|--|--|--|
| f ₁ schwarz-weiß-grün 1,0 mm ² | h braun 0,75 mm ² | f ₁ gelb-schwarz 1,5 mm ² | t grau 1,0 mm ² | u blau 0,5 mm ² | w grau-grün 0,5 | | | | | | |
| f ₂ weiß-schwarz 2,5 mm ² | i grau-grün 0,75 mm ² | f ₂ gelb 1,5 mm ² | t ₁ grau-rot 0,5 mm ² | v ₁ blau-grün 0,5 mm ² | x ₁ schwarz-weiß 1,0 | | | | | | |
| f ₃ weiß 3,5 mm ² | k rot 0,5 mm ² | f ₃ weiß-schwarz 1,5 mm ² | t ₂ grau-schwarz 0,5 mm ² | v ₂ blau-rot 0,5 mm ² | y ₁ schwarz-grün 1,0 | | | | | | |
| f ₄ gelb 2,5 mm ² | n schwarz 0,83 mm ² | f ₄ weiß 1,5 mm ² | t ₃ grau 0,5 mm ² | v ₃ blau-rot 0,5 mm ² | y ₂ braun 1,0 | | | | | | |
| h ₁ schwarz-rot 0,75 mm ² | o schwarz 0,75 mm ² | f ₄ gelb-schwarz 0,5 mm ² | t ₄ schwarz-rot 0,75 mm ² | v ₄ schwarz 0,5 mm ² | y ₃ schwarz-gelb 1,0 | | | | | | |



- 11.136 electrical parts
- 61 ECUs
- Optical bus for information and entertainment
- Sub networks based on proprietary serial bus
- 35 ECUs connected to 3 CAN-Busses
- 2500 signals in 250 CAN messages

Der Wert zukünftiger Automobile wird bis zu 40% durch eingeb. HW/SW Komponenten bestimmt.

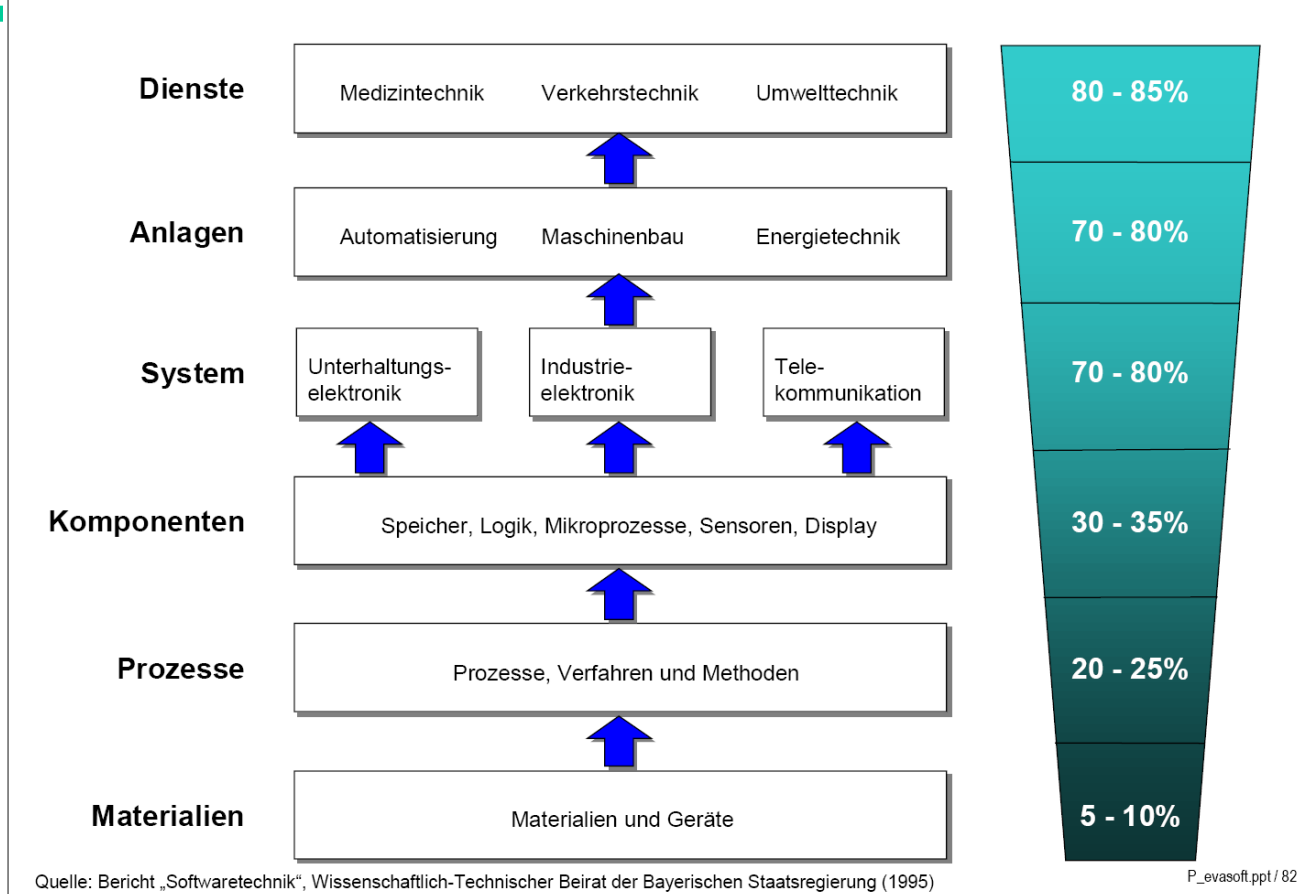


Anwendungen

- **Control of industrial production processes**
- **Plant management**
- **Power plant control (conventional, nuclear)**
- **Railway control systems**
- **Flight control systems**
- **Environmental acquisition and monitoring**
- **Space missions**
- **Military systems**
- **Telecommunication systems**
- **Robotics**
- **Team Robotics and autonomous vehicles**
- **Virtual and augmented reality**



Querschnittsbedeutung von Software



Anteil des Umsatzes, der auf die Entwicklung bzw. den Einsatz von Software zurückzuführen ist.

„Analyse und Evaluation der Softwareentwicklung in Deutschland“, Eine Studie für das Bundesministerium für Bildung und Forschung, Dez. 2001

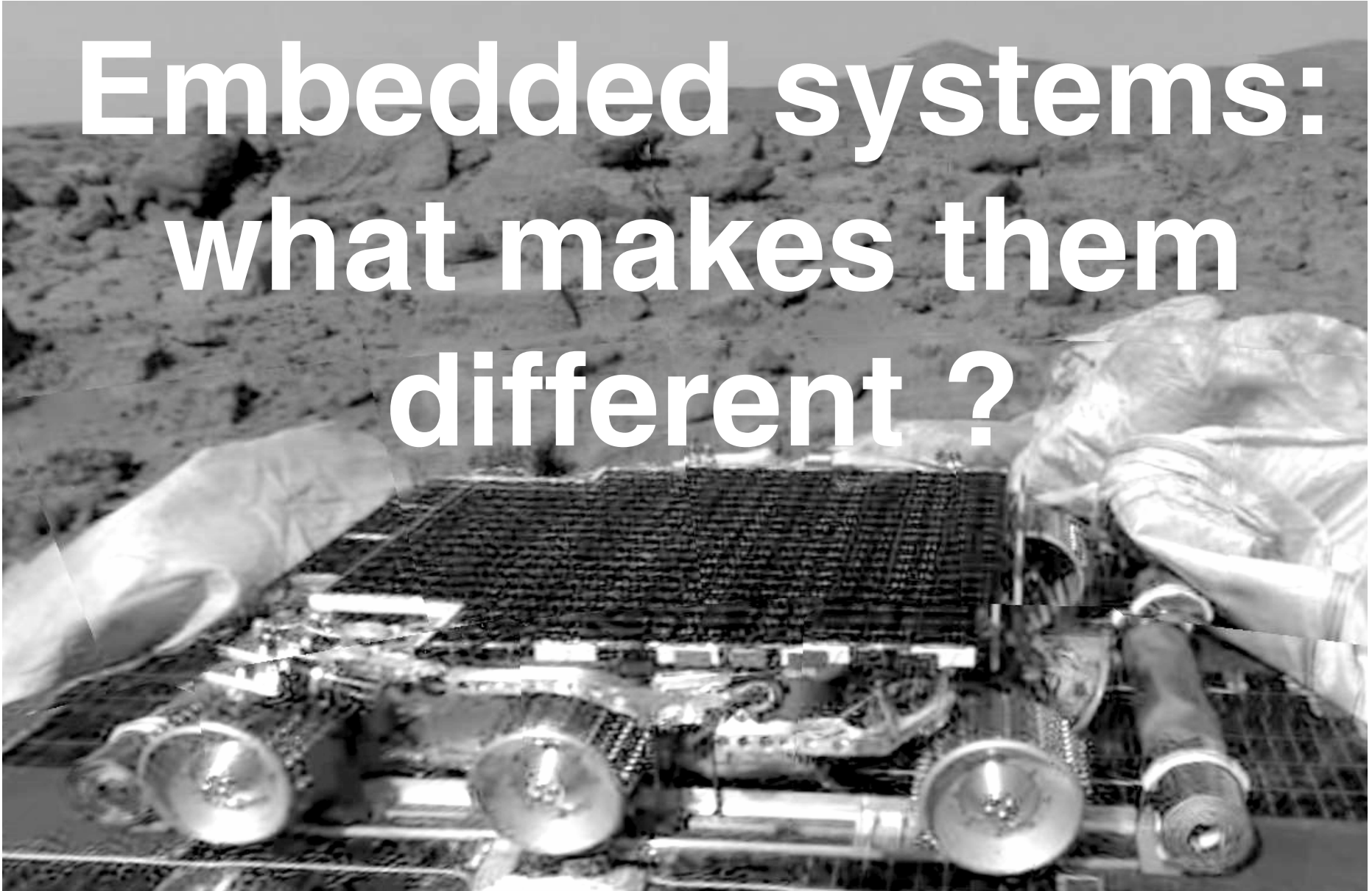


Stärken der deutschen Softwareindustrie

- **Sekundärbranchen repräsentieren in Deutschland die traditionellen Stärken der Industrie (Automobil, E-Technik, Telekommunikation)**
- **Deutsche Stärken**
 - **Bereitstellung von Individuallösungen/Varianten**
 - **Gute Beherrschung von Komplexität - ingenieurmäßiges Vorgehen**
- **Sekundärbranchen müssen diese Stärken auf ihre Software übertragen, die eine immer größere Rolle in ihren Produkten und Dienstleistungen spielt**
- **Dazu notwendig in Primärbranche und Sekundärbranchen**
 - **Beherrschung von Software-Variantenbildung**
 - **Garantie von Zuverlässigkeit/Sicherheit**
 - **Beherrschung der Auftraggeber-/Auftragnehmerschnittstelle**



Embedded systems: what makes them different ?



Typen eingebetteter Systeme

Allgemeine eingebettete Systeme:

Funktionen (und Probleme) ähnlich wie general purpose Rechner aber in einer speziellen "Verpackung".

Beispiele: Videospiele, Set-top-Boxen, PDAs, Navigationssysteme, Geldautomaten,...

Kontrollsysteme:

Kontrolle physischer Prozesse, Rückkopplungsschleifen unter Echtzeitbedingungen

Beispiele: Motoren, chemische Prozesse, Kraftwerke, Fly-by-Wire-Systeme, ABS, EPS,....

Signalverarbeitungssysteme:

Verarbeitung von Datenströmen, Störungsbeseitigung, Filterung, Signalerkennung

Beispiele: Radar, Sonar, Funktechnik, Video(de)kompression, Verschlüsselung....

Kommunikation und Netze:

Übertragung, Verbindung, Routing

Beispiele: Telefonverbindungsanlagen, Basisstationen, Router, Internetinfrastruktur....

Autonomie-Eigenschaften

Identifizierbares System

Unabhängigkeit

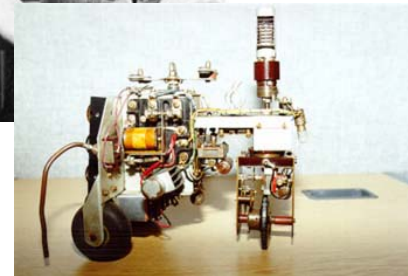
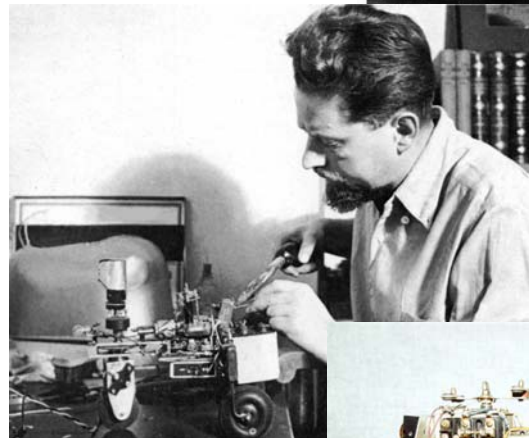
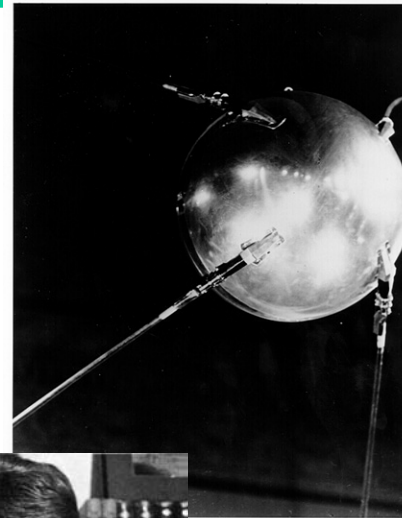
Erkennen von Aspekten der Umwelt

Eigenkontrolle

Selbstbewertung

Selbstständiges Agieren

Selbslokalisierung



Welche Funktionen werden in eingebetteten Systemen vorzugsweise benötigt?

Kontrollfunktionen:

PID Kontrolle, Fuzzy Logic, ...

Anwendungsbezogene Schnittstellenrealisierung:

Knöpfe, Anzeigen, LEDs, Beeper,

Fehlerbehandlung:

Erkennung, Rekonfiguration und Recovery

Signalverarbeitung:

Digitale Filter,

Eigenschaften:

Reaktiv: Berechnungen und Aktionen werden abhängig von externen Ereignissen durchgeführt.

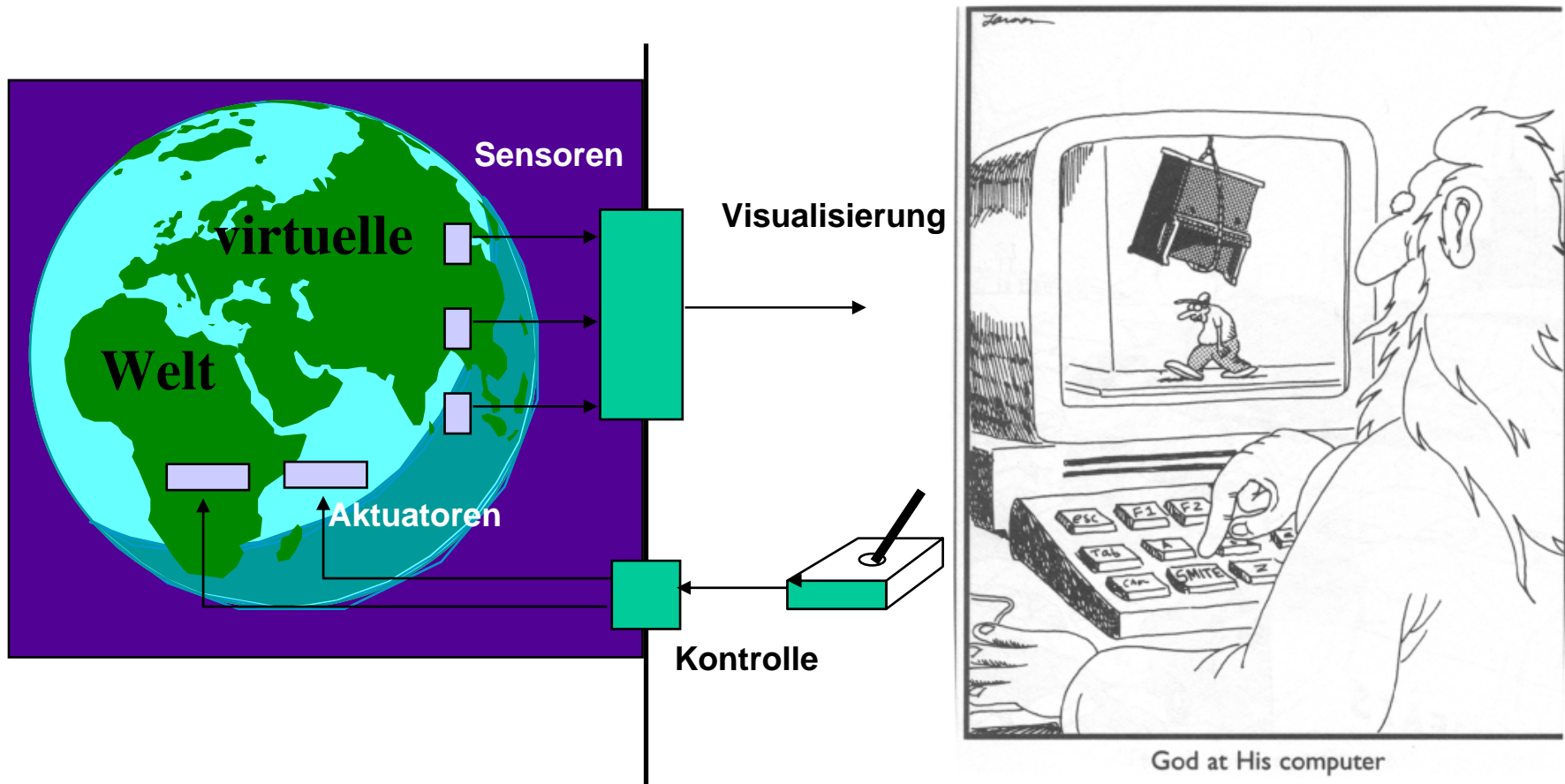
Proaktiv: Aktionen werden ohne Benutzerintervention vorbereitet und durchgeführt.

Vorhersagbarkeit: Korrektheit von Berechnungen und Aktionen müssen zuverlässig und zeitgerecht durchgeführt werden.

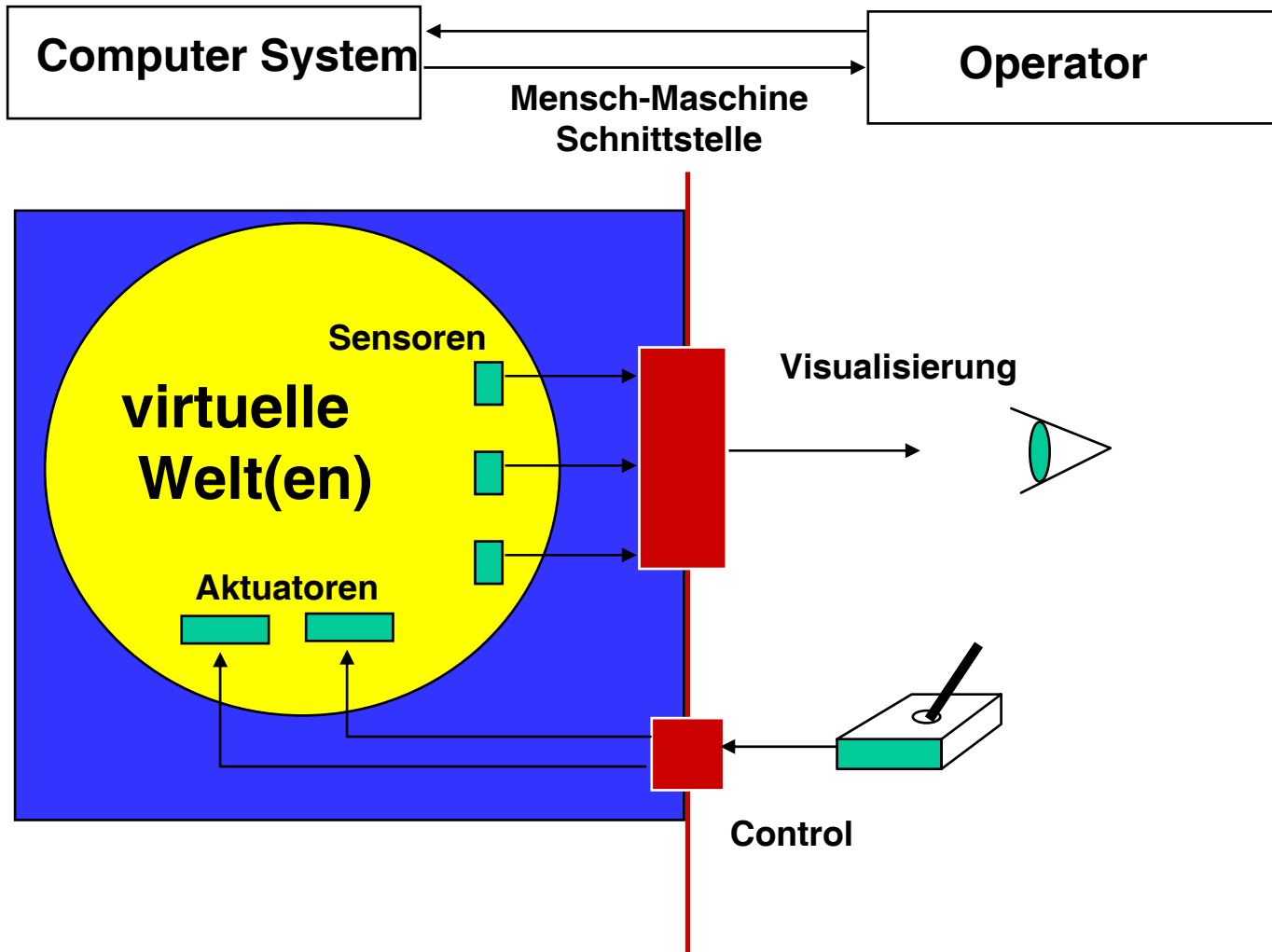


Simulated Worlds

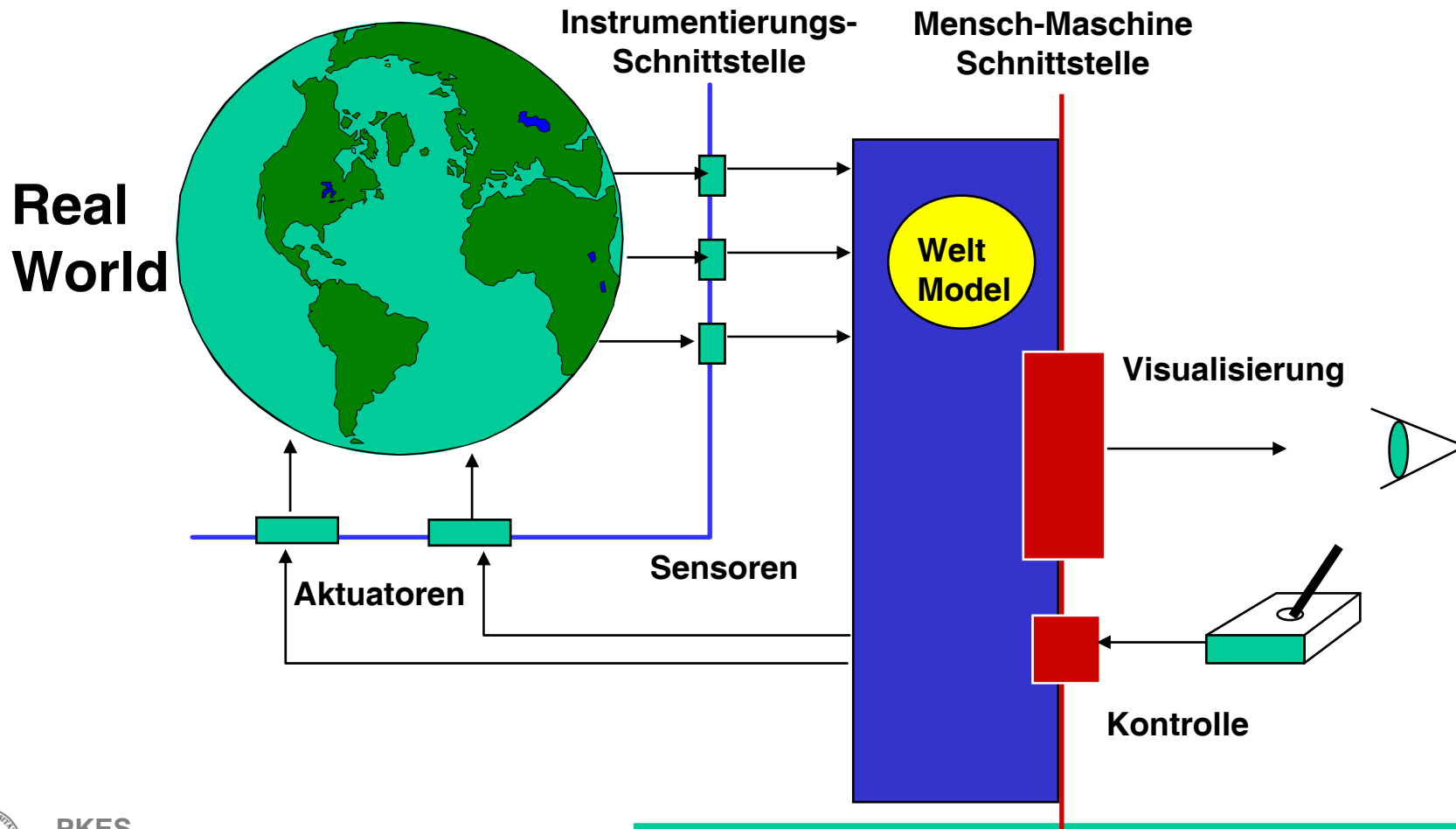
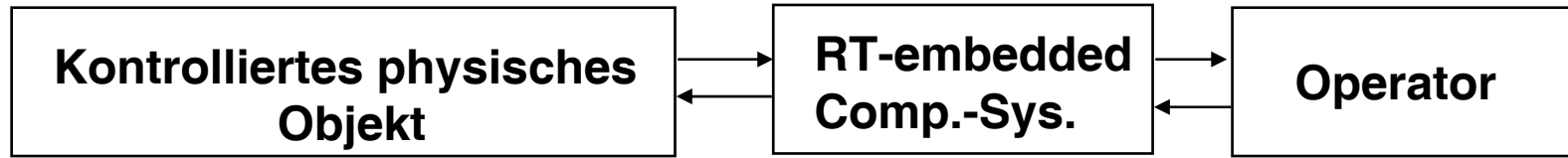
The Masters of the (Time) Universe



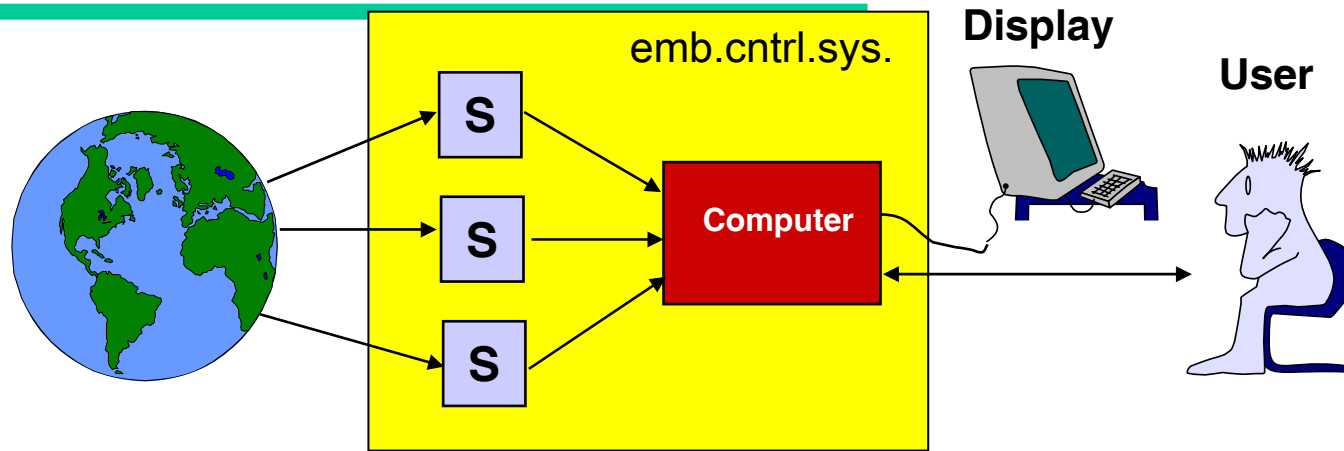
Ein Computer System



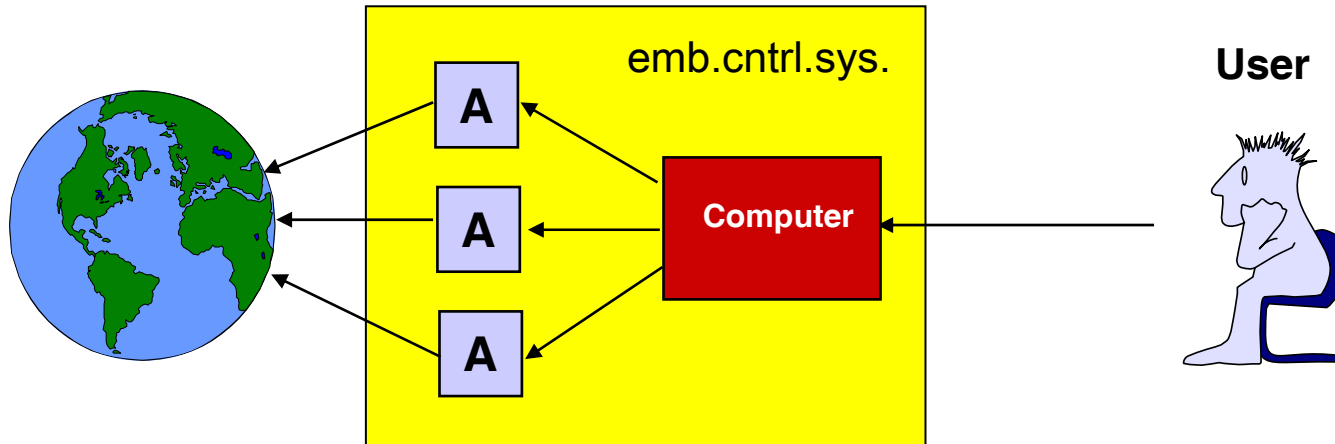
Ein Computer System zur Kontrolle physischer Prozessen



Monitoring System



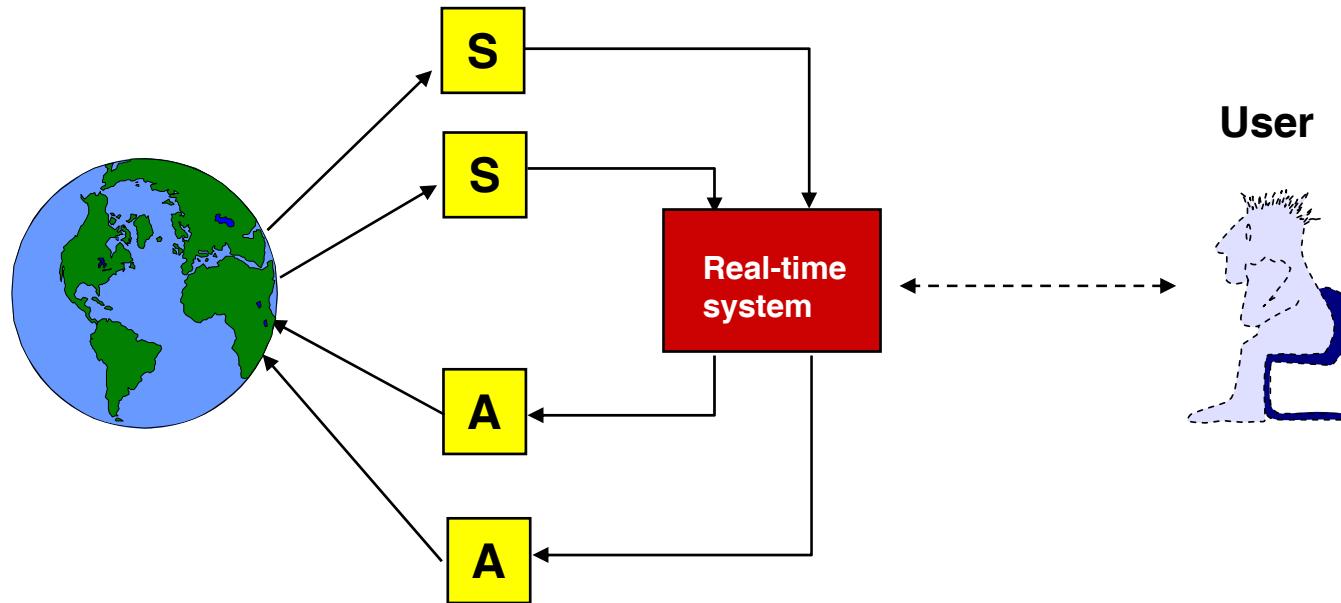
b.) "Open Loop" System



a + b = Human-in-the-loop



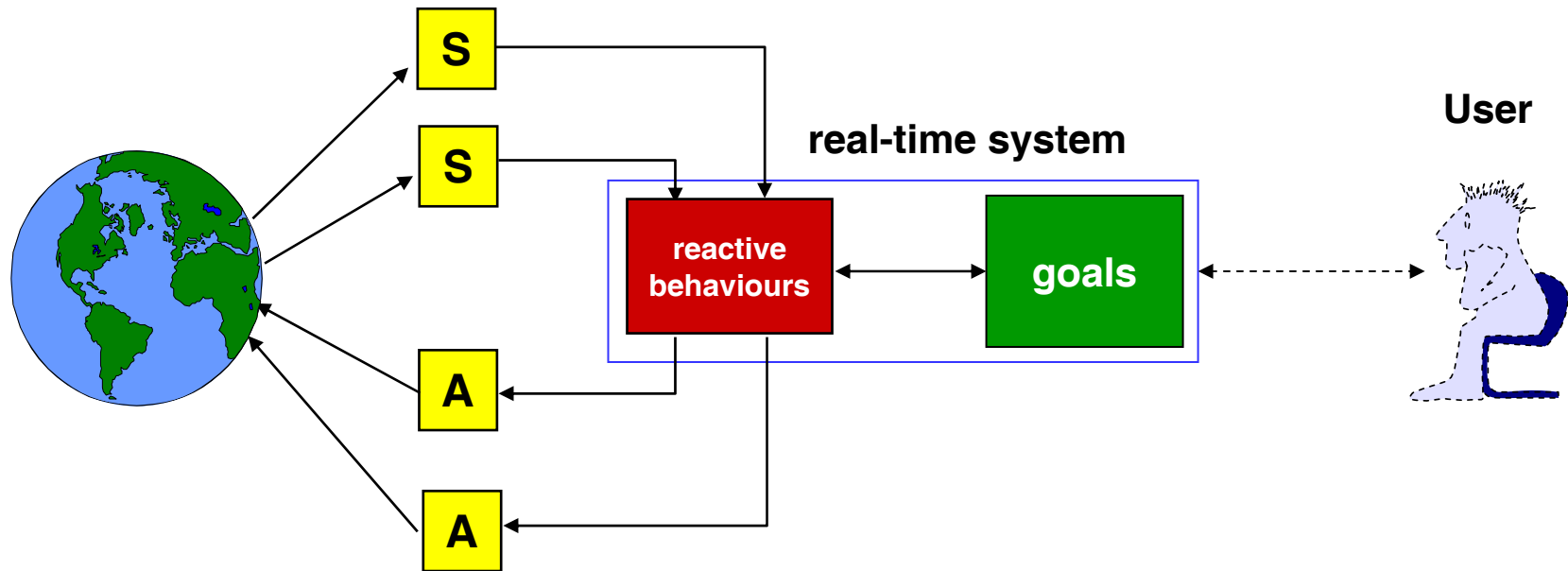
“Feedback Control” System



Aktuation ändert die Umgebung:
Die Umgebung ist ein Kommunikationskanal!

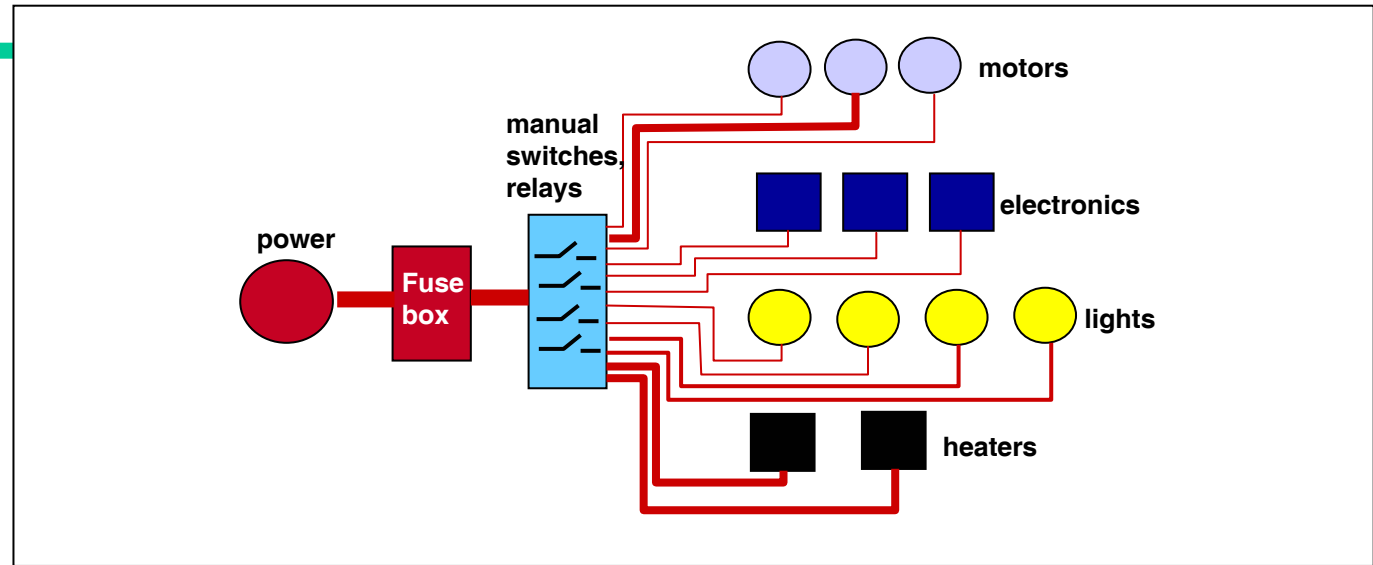


Autonome Kontrolle



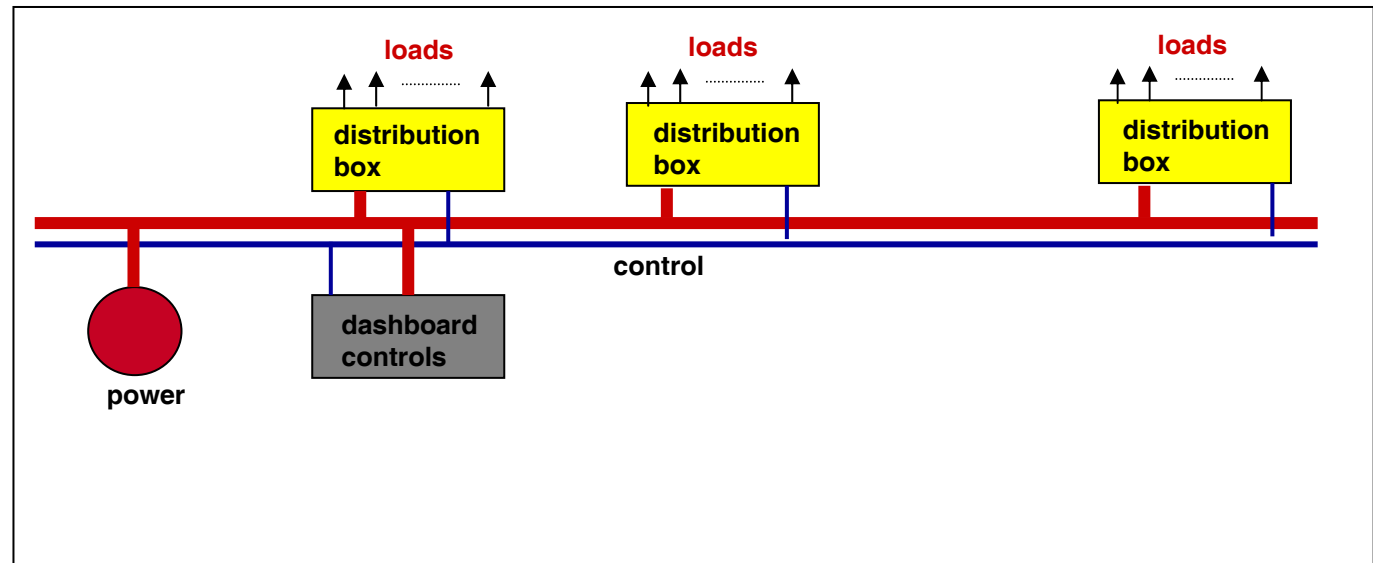
Szenario: Car Control

gestern



heute

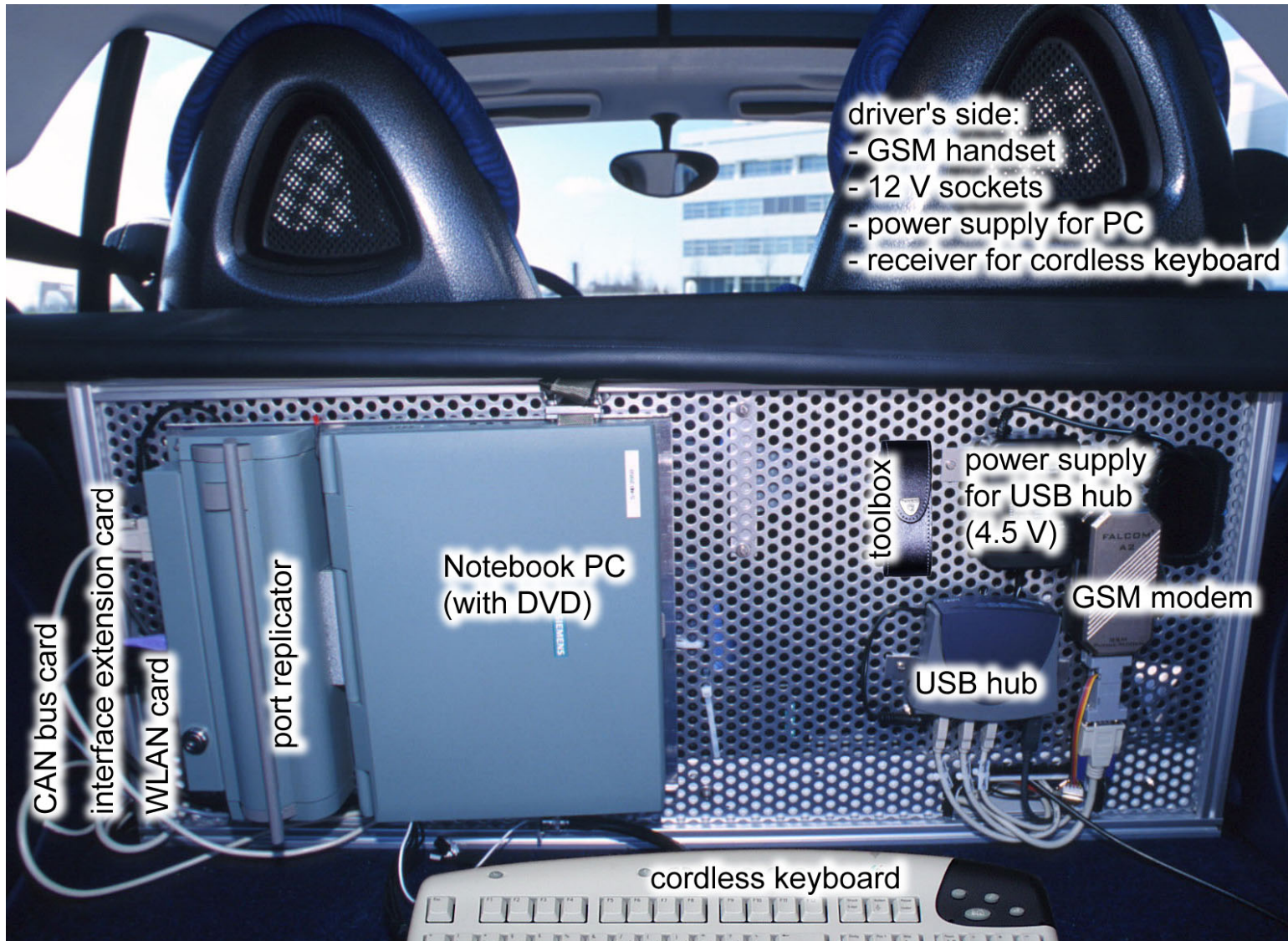
Die Ursache für jedes zweite liegengeliebene Auto ist ein Defekt der Elektronik!
Die Welt, Sept. 2002



Eine multimodale Schnittstelle



Konventionelle Rechnerinfrastruktur



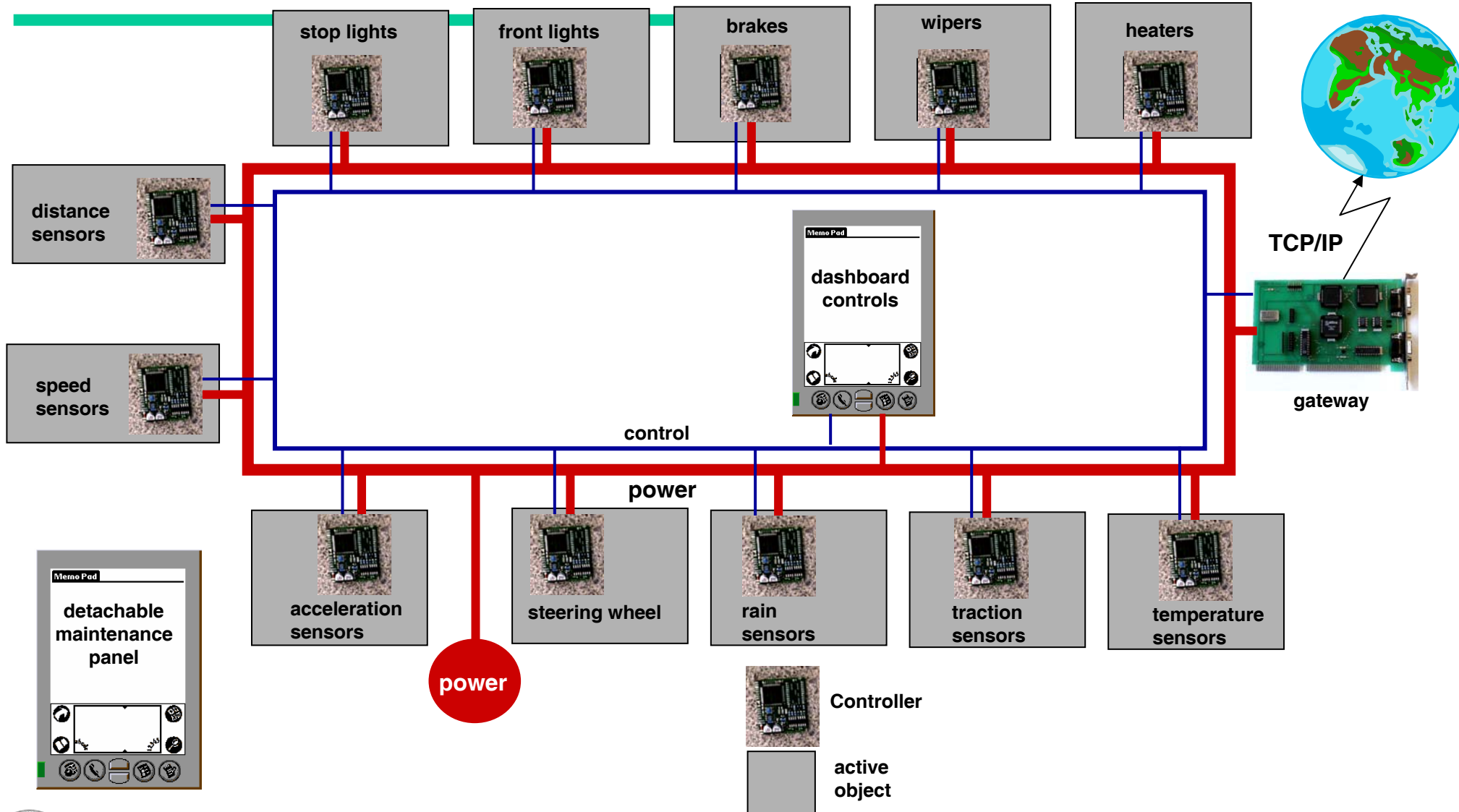
Services in einem Auto?

- Beschränkte Interaktion mit dem Fahrer.
- Breite Nutzer-Community ohne spezielle IT-Kenntnisse
- Services müssen ohne Fahrerintervention funktionieren.
- Nutzerprofile und on-board Ausstattung müssen bei der Wahl der Dienste berücksichtigt werden!
- Context Information wird bei der Erbringung der Dienste benötigt !

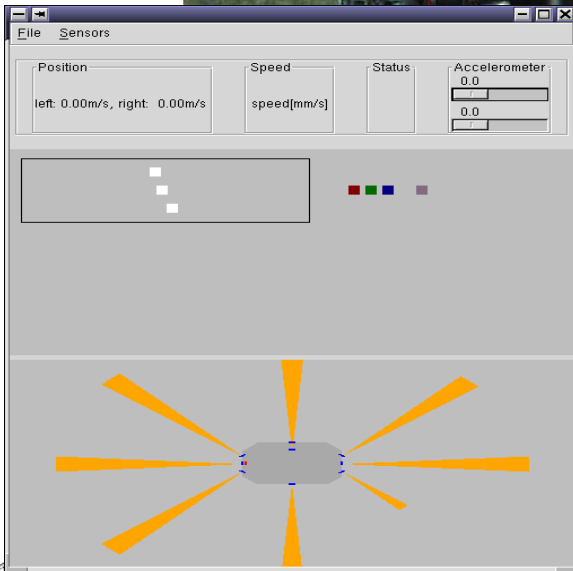
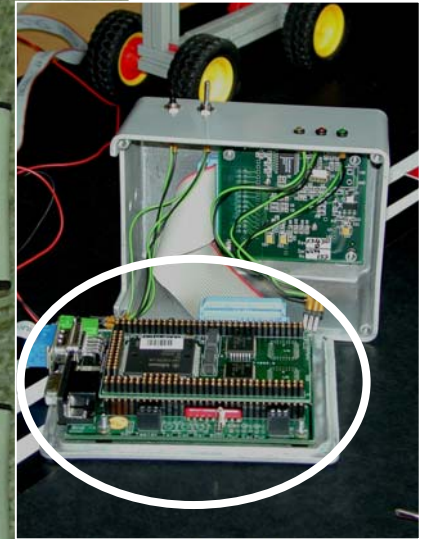
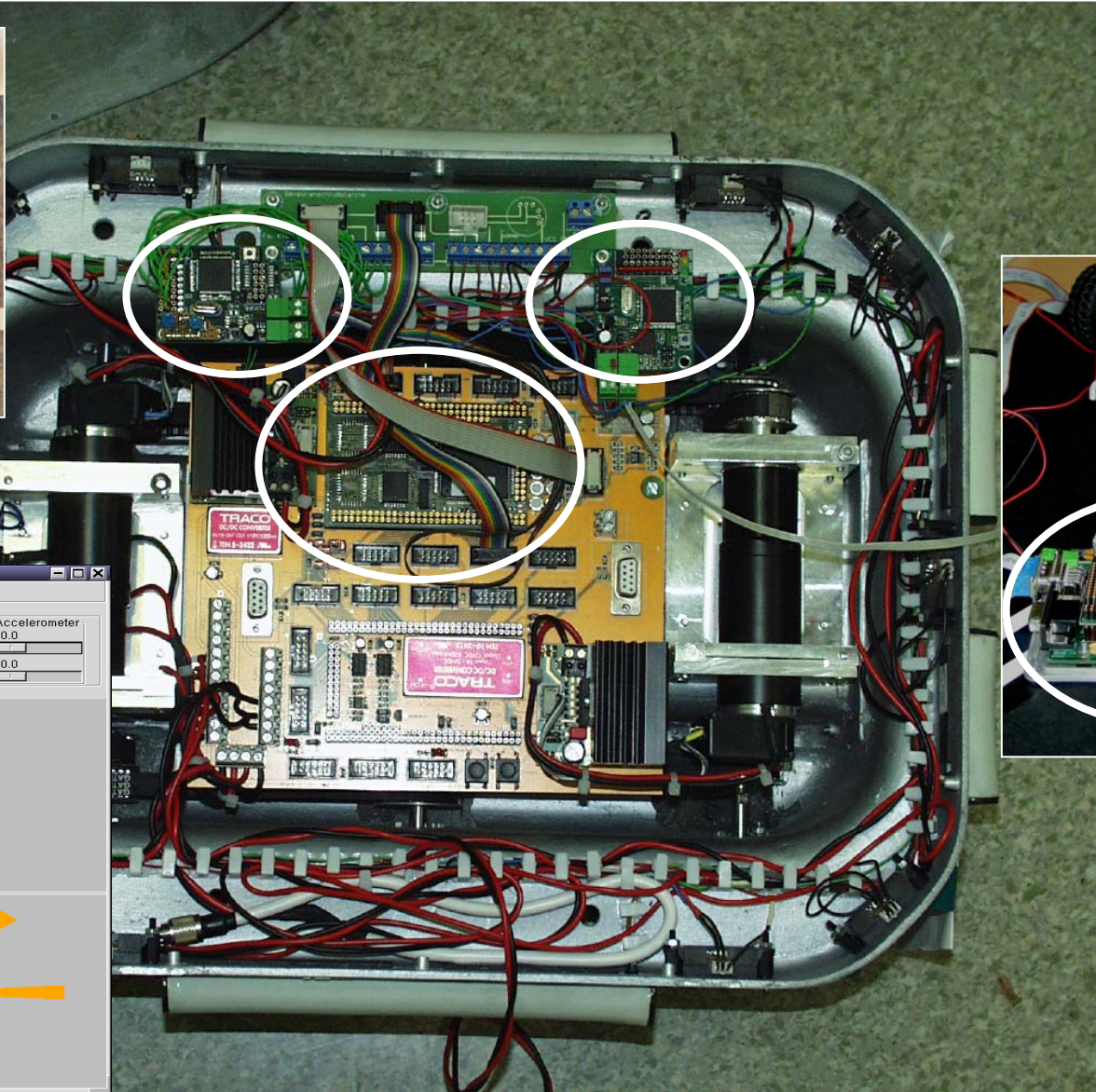
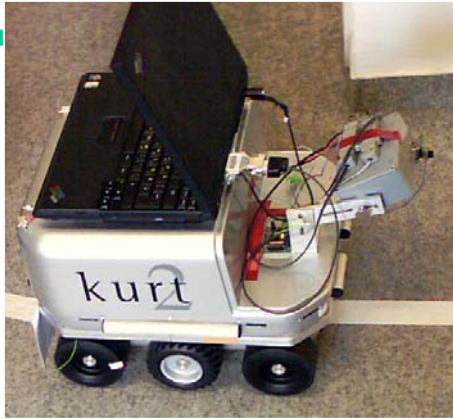


EINE UNFALLURSACHE, DIE AN HÄUFIGKEIT ZUNIMMT:
UNAUFMERKSAMKEIT BEIM FAXEN

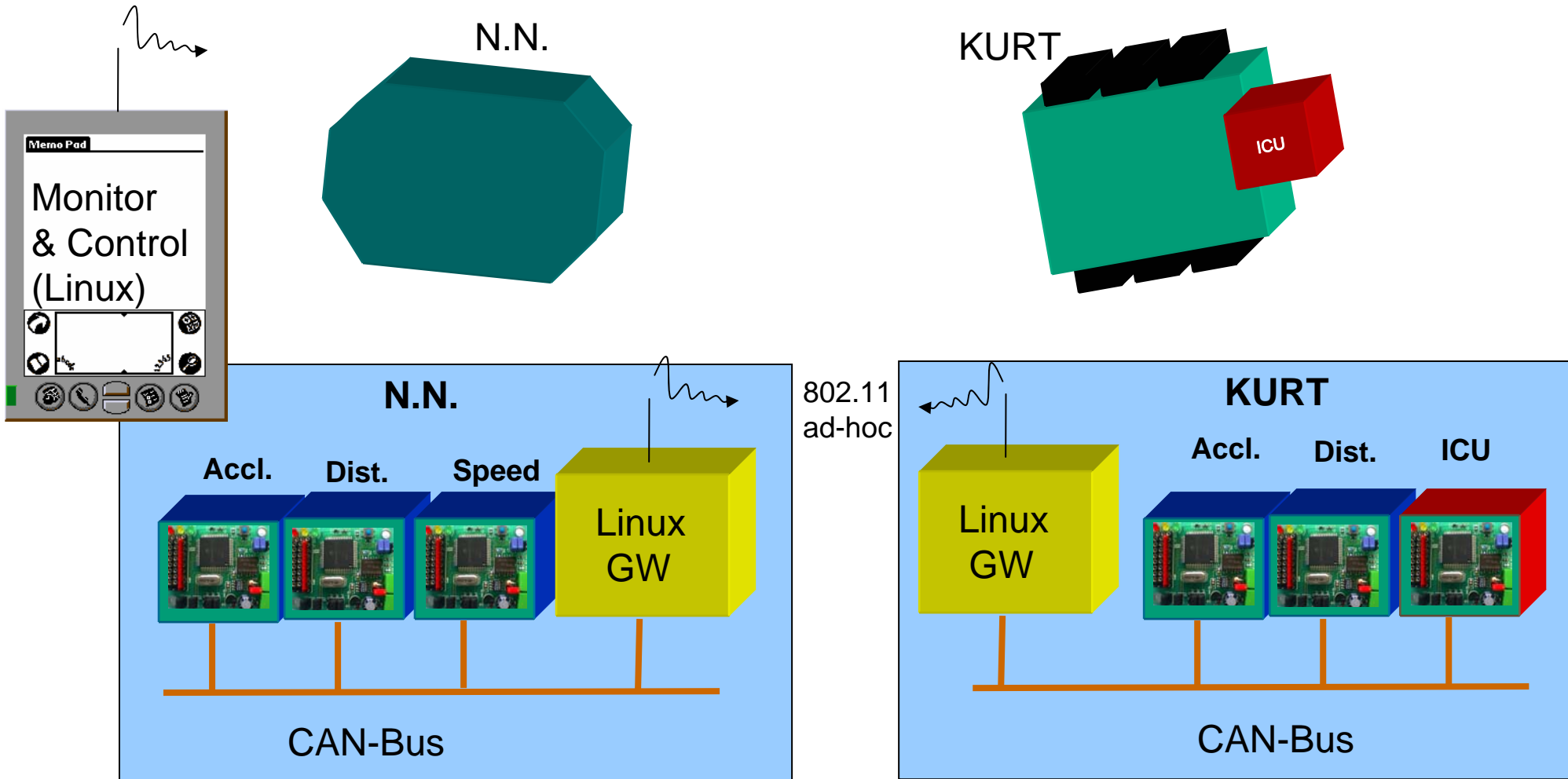
Zukunft: verteilte kooperative Kontrolle



Verteilte Robotersteuerung

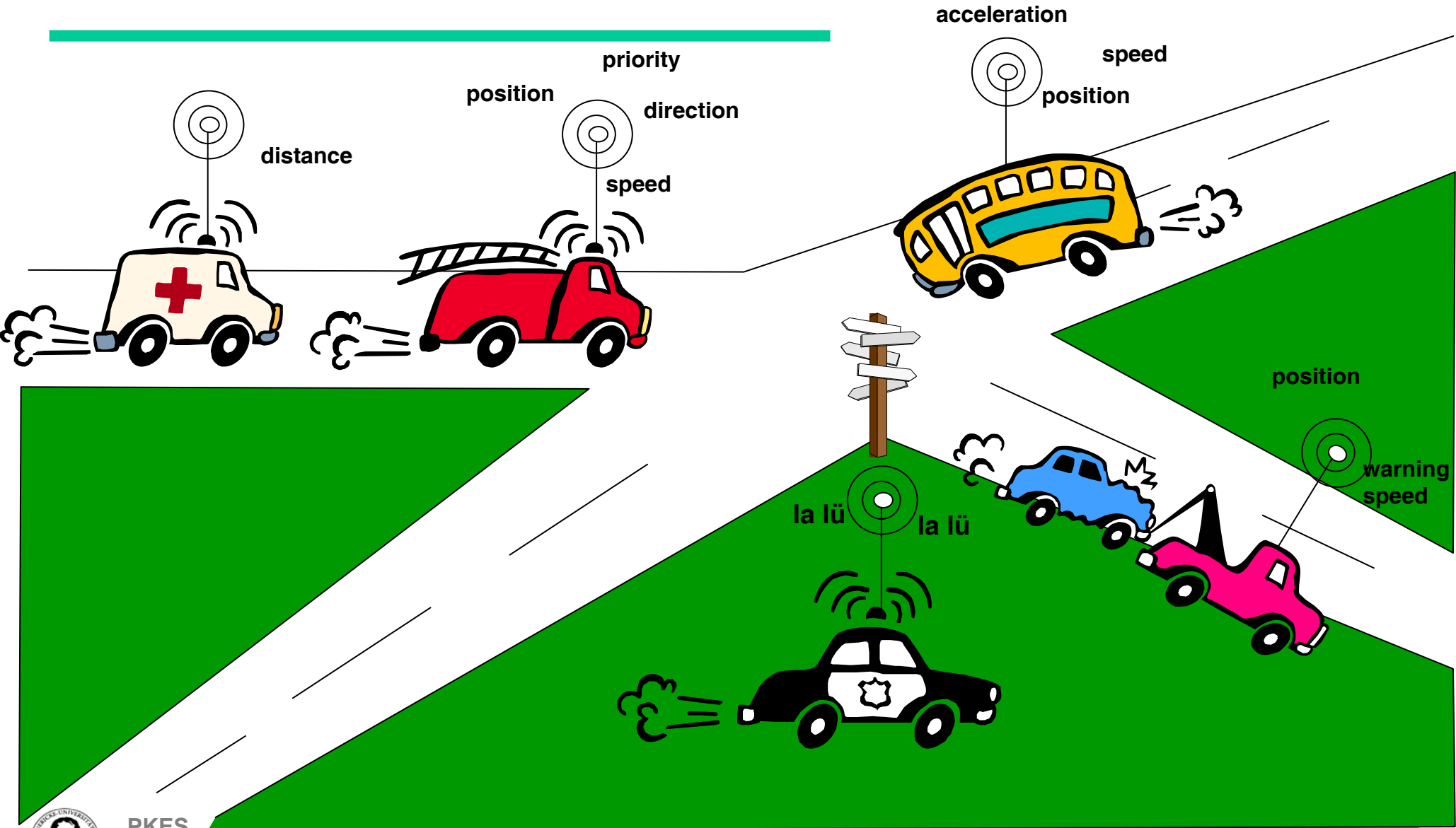


Demo Szenario





Kooperierende Fahrzeuge



Murphy's Gesetze über Steuersysteme

Murphy's general law:

If something can go wrong, it will go wrong.

Murphy's constant:

Damage to an object is proportional to its value.

Johnson's first law:

sooner or later, the worst possible combination of circumstances will happen.

Corollary:

A system must always be designed to resist the worst possible combination of circumstances



- **Es gibt keine technische Lösung für ein 100 % sicheres System !**

Vorhersagbarkeit des Verhaltens ist immer auf implizite und explizite Annahmen gegründet über:

- **die Umgebung**
- **die Systemkomponenten**

Es gibt gute wissenschaftliche Ansätze und Ingenieursprinzipien um solche Systeme zu bauen und ihre Funktion vorherzusagen.



Literatur:

Embedded Systems und Micro-Controller:

Wayne Wolf:
Computers as Components
Principles of Embedded Computing System Design
Academic Press, 2001

Uwe Brinkschulte, Theo Ungerer:
Mikrocontroller und Mikroprozessoren
Springer Lehrbuch, 2002

Architektur von RT-Emb.-Systemen und Übersicht:

Paulo Verissimo, Luis Rodrigues:
„Distributed Systems for System Architects“
Kluwer Academic Publishers, 2001

Hermann Kopetz:
„Real-Time Systems, Design Principles for Distributed
Embedded Applications“
Kluwer Academic Publishers, 1997

RT-Scheduling

Giorgio C. Buttazzo:
Hard Real-Time Computing Systems
„Predictable Scheduling Algorithms and
Applications“
Kluwer Academic Publishers, Boston, Dordrecht,
London, 2000

Dieter Zöbel, Wolfgang Albrecht
Echtzeitsysteme - Grundlagen und Anwendungen
Informatik-Lehrbuch-Reihe
(Hrsg. B. Mahr, A. Schill, G. Vossen)
International Thomson Publishing Comp., 1995

weitere Literatur in der Vorlesung.

