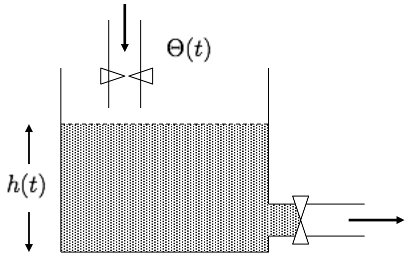
Real-Time Systems: Examples / Case Studies

- Simple Control System
- Sampling Periods
- Quality of the Control vs. Processing Cost
- Protection of Resources in Integrated Systems
- Multimedia / Real-Time Communication
- Anomalies in Asynchronous Systems
  - Example: Advanced Fighter Technology Integration (AFTI) F16
- Priority Inversion

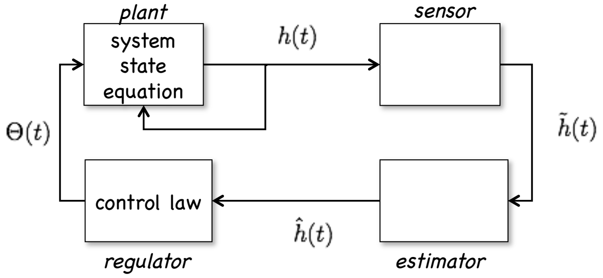
- Real-Time Systems
- Hard and soft deadlines; operational definition

Application Areas: Control Systems

- Example: Water Tank



- In other words



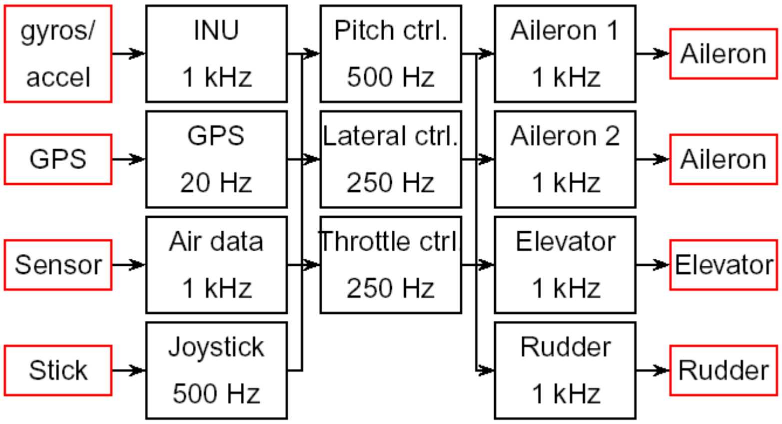
Control Systems (cont)

- Control Loop:

```
DO FOREVER
  wait_for_delay
  h      := fluid_height
  theta  := valve_position
  r      := table_lookup(h, theta)
  IF       r = left   THEN turn_left
  ELSE IF  r = right  THEN turn_right
  ELSE     do_nothing
ENDDO
```

Example: Avionics System

Hard real-time system with multirate behavior

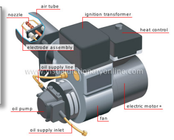


Quality of Control vs. Processing Cost

Example: Open-Loop Temperature Control

[Simplified from : Setol, Lehoczky, Sha, and Shin, “On Task Schedulability in Real-Time Control Systems”, Proceeding of the 1996 IEEE Real-Time Systems Symposium]

System: Temperature of a unit is controlled by a burner.



Dynamic equation:

$$\dot{x} = -ax + bu$$

x – difference between unit and ambient temperature, $x(0) = 0$
u – control input (rate of heat)

Control Problem: change temperature of unit to x_d within time t_f; consume minimum amount of fuel. Allow for a tolerance δ.

$$|x(t_f) - x_d| \leq \delta$$

Performance Index $J(u)$ of control system: measure of total cost of control and accuracy generated in time period $[0, t_f]$ by control u. Generally:

$$J(u) = S(s(t_f), t_f) + \int_0^{t_f} L(x(t), u(t), t)dt$$

Optimal control $u^*(t)$ with performance index J^*.

Open-Loop Temperature Control (cont)

Our case: minimize fuel.

$$min_u J = \frac{1}{2}p(x(t_f) - x_d)^2 + \frac{1}{2}\int_0^{t_f} u^2(t)dt$$

Resulting **optimal control:**

$$u^*(t) = \frac{x_d pabe^{at}}{ae^{at_f} + pb^2 sinh(at_f)}$$

Final State:

$$x^*(t_f) = \frac{x_d pb^2 sinh(at_f)}{ae^{at_f} + pb^2 sinh(at_f)}$$

Open-Loop Temperature Control (cont)

Discretize control input *u*:

Sampling period *P*.

$$\dot{x}^*(t) = -ax^*(t) + bu^*(kP) \qquad kP \le t \le (k+1)P$$

Performance index for discrete optimal control:

$$J_D^*(P) \simeq S(x^*(t_f), t_f) + \sum_{k=0}^{n-1} \int_{kP}^{(k+1)P} L(x^*(t), u^*(kP), t)dt$$

In our case:

$$J_D^*(P) \simeq \frac{1}{2} p x_d \left(\frac{1 - e^{-aP}}{1 + e^{-aP}} \right)$$

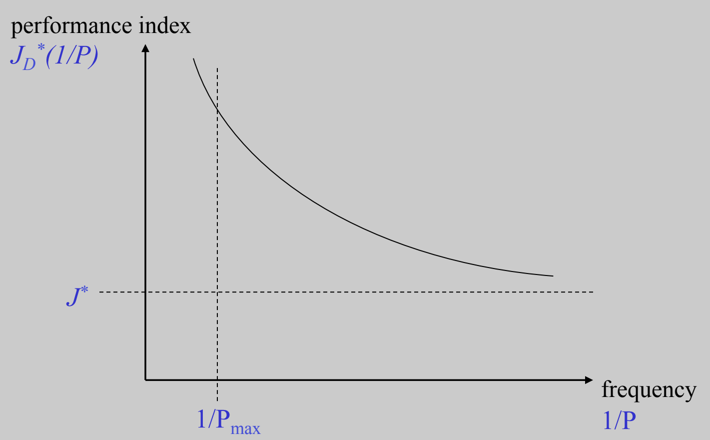
Constraints:

$$|x(t_f) - x_d| \le \delta$$

$$x_d \left(\frac{1 - e^{-aP}}{1 + e^{-aP}} \right) \le \delta \quad \Rightarrow \quad P \le \frac{1}{a} ln \frac{x_d + \delta}{x_d - \delta}$$

Open-Loop Temperature Control (cont)

- Effect of sampling period on performance index.



Quality of Control *vs.* Processing Cost (cont)

Task frequencies must be determined to optimize the performance indices without overloading the available processing capabilities.

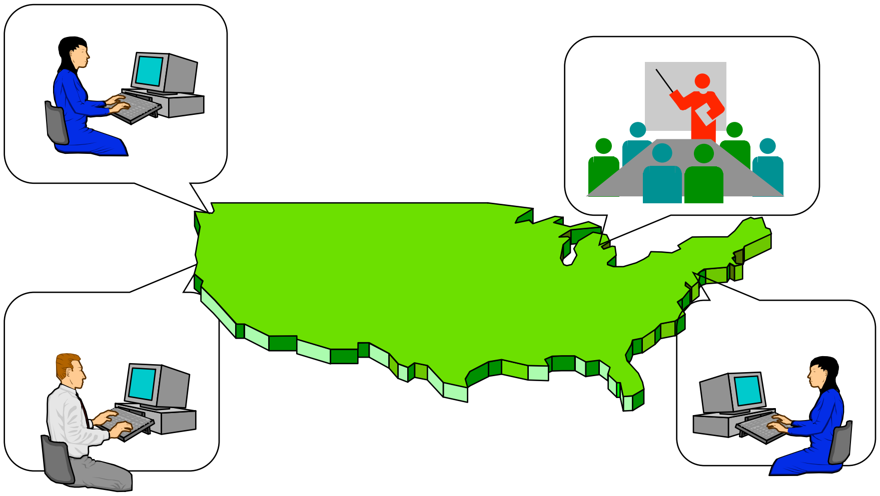
Notation: $\Delta J^*(P) := J_D^*(P) - J^*$

Optimization problem:

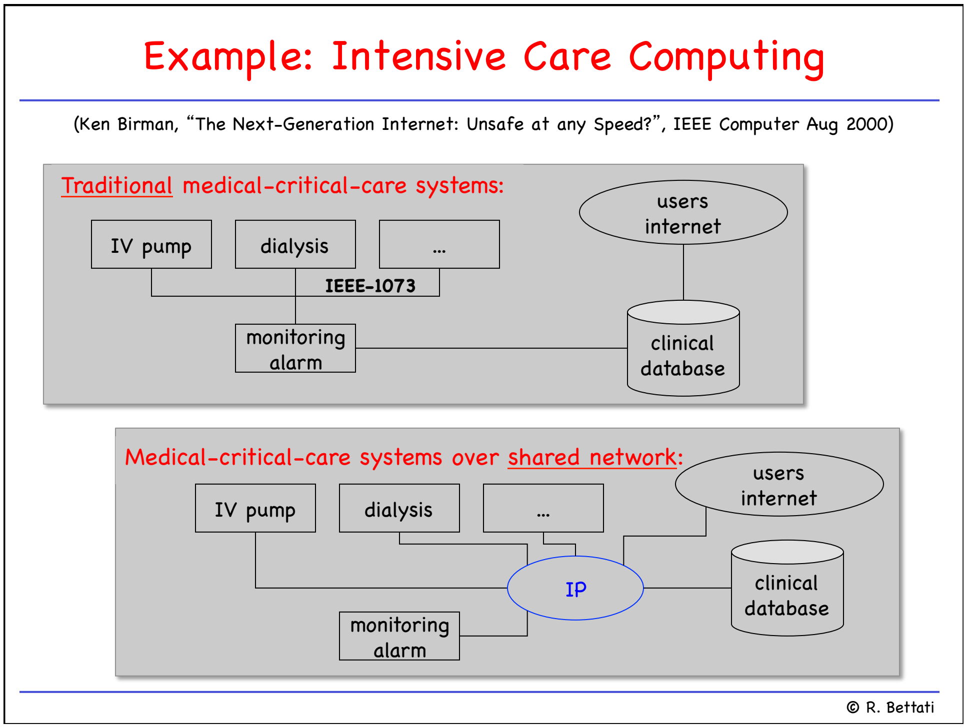
Given a set of tasks, $T_1, \ldots, T_n$, with given $\Delta J^*_i(\bullet)$ and execution times C_i, find a set of periods P_i, such that

1. $P_i <= P_i^{max}$ // Maintain stability
2. Minimize (maximize) $\sum_{i=1}^{n} \Delta J_i^*(P_i)$ // Optimize total // performance index
3. Resource Constraint: $\sum_{i=1}^{n} \frac{1}{P_i} C_i \leq U$ // CPU capacity

Example: Multimedia



Example: Teleseminars



Example: Intensive Care Computing

(Ken Birman, “The Next-Generation Internet: Unsafe at any Speed?”, IEEE Computer Aug 2000)

Traditional medical-critical-care systems:

IV pump

dialysis

...

IEEE-1073

monitoring alarm

users internet

clinical database

Medical-critical-care systems over shared network:

IV pump

dialysis

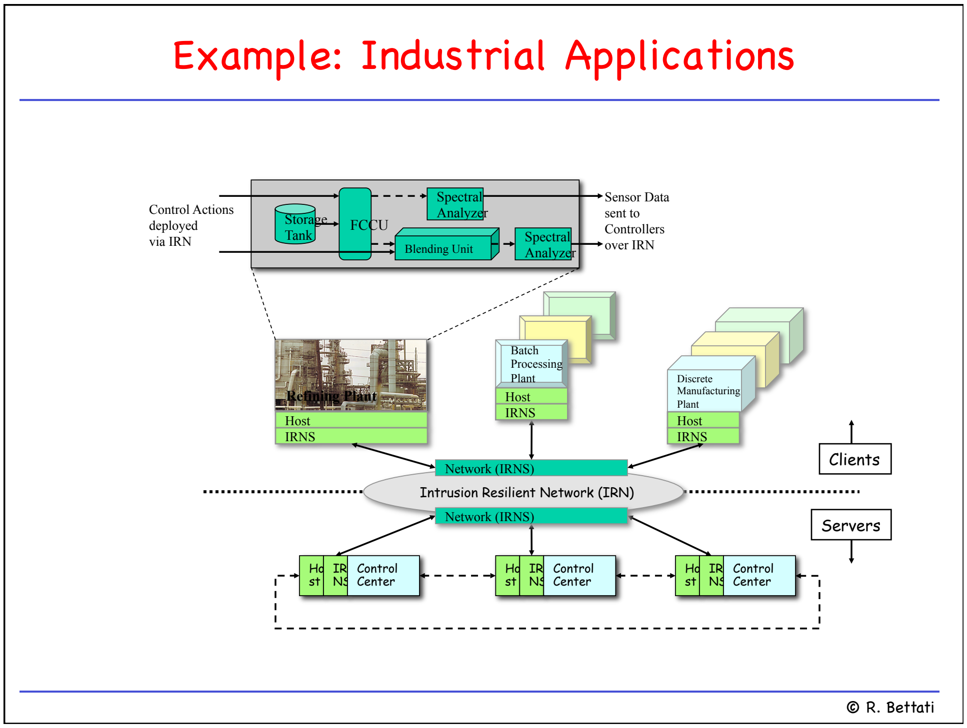
...

IP

monitoring alarm

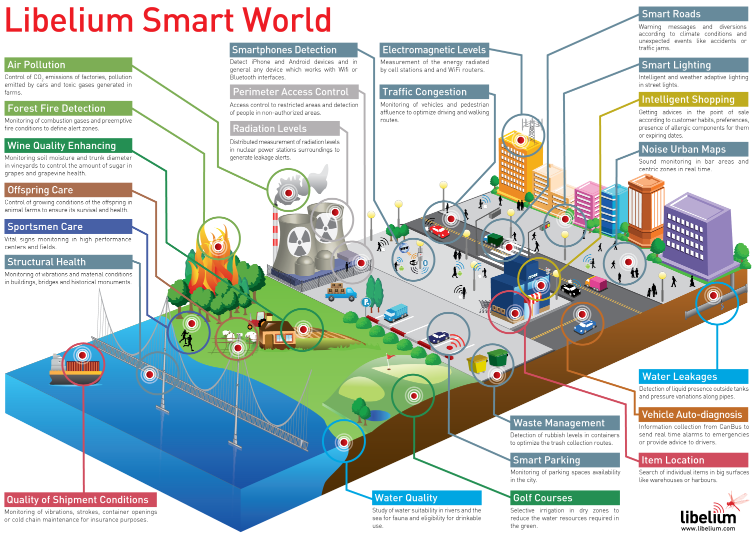
users internet

clinical database



Example: Industrial Applications

Example: Internet of Things



Libelium Smart World

Air Pollution
Control of CO_2 emissions of factories, pollution emitted by cars and toxic gases generated in farms.

Forest Fire Detection
Monitoring of combustion gases and preemptive fire conditions to define alert zones.

Wine Quality Enhancing
Monitoring soil moisture and trunk diameter in vineyards to control the amount of sugar in grapes and grapevine health.

Offspring Care
Control of growing conditions of the offspring in animal farms to ensure its survival and health.

Sportsmen Care
Vital signs monitoring in high performance centers and fields.

Structural Health
Monitoring of vibrations and material conditions in buildings, bridges and historical monuments.

Smartphones Detection
Detect iPhone and Android devices and in general any device which works with Wifi or Bluetooth interfaces.

Perimeter Access Control
Access control to restricted areas and detection of people in non-authorized areas.

Radiation Levels
Distributed measurement of radiation levels in nuclear power stations surroundings to generate leakage alerts.

Electromagnetic Levels
Measurement of the energy radiated by cell stations and and WiFi routers.

Traffic Congestion
Monitoring of vehicles and pedestrian affluence to optimize driving and walking routes.

Smart Roads
Warning messages and diversions according to climate conditions and unexpected events like accidents or traffic jams.

Smart Lighting
Intelligent and weather adaptive lighting in street lights.

Intelligent Shopping
Getting advices in the point of sale according to customer habits, preferences, presence of allergic components for them or expiring dates.

Noise Urban Maps
Sound monitoring in bar areas and centric zones in real time.

Water Leakages
Detection of liquid presence outside tanks and pressure variations along pipes.

Vehicle Auto-diagnosis
Information collection from CanBus to send real time alarms to emergencies or provide advice to drivers.

Item Location
Search of individual items in big surfaces like warehouses or harbours.

Waste Management
Detection of rubbish levels in containers to optimize the trash collection routes.

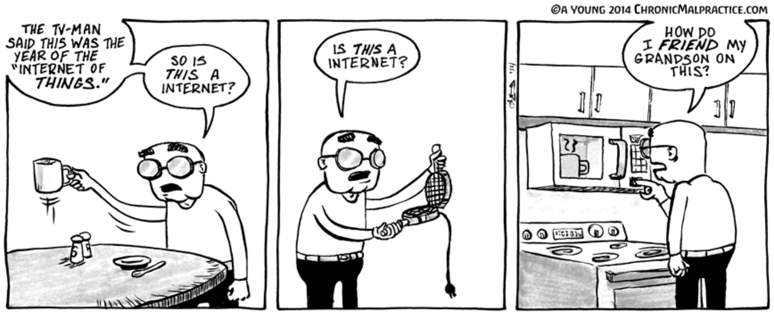
Smart Parking
Monitoring of parking spaces availability in the city.

Golf Courses
Selective irrigation in dry zones to reduce the water resources required in the green.

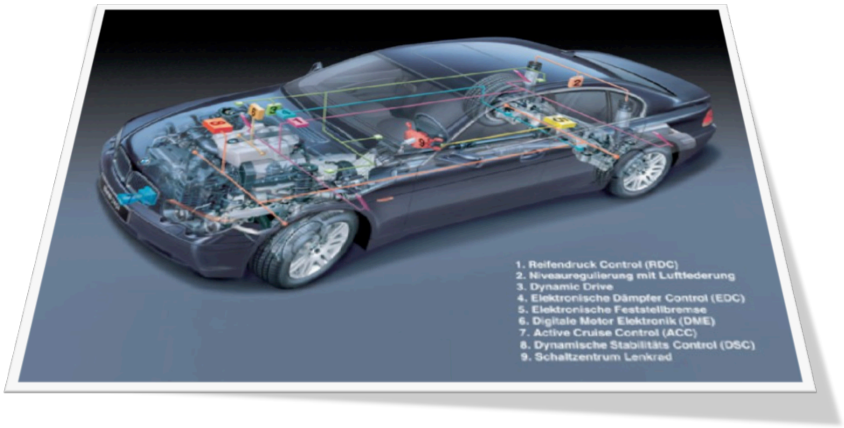
Water Quality
Study of water suitability in rivers and the sea for fauna and eligibility for drinkable use.

Quality of Shipment Conditions
Monitoring of vibrations, strokes, container openings or cold chain maintenance for insurance purposes.

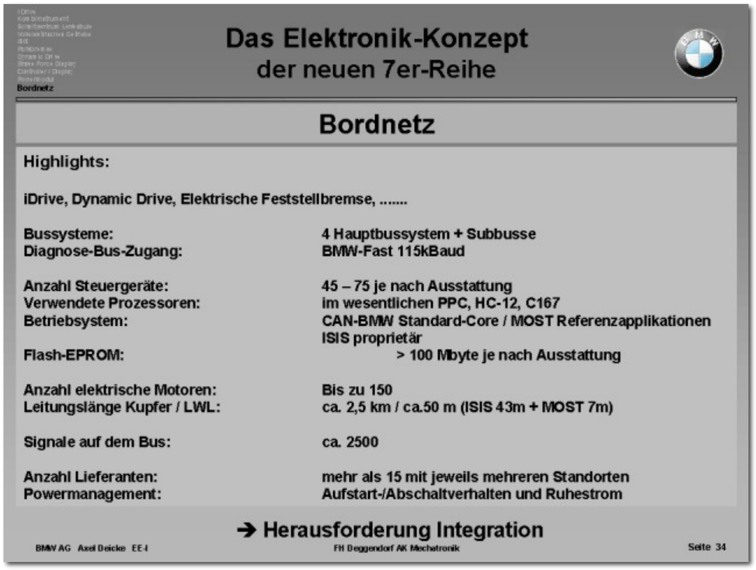
Internet of Things (II)



Example: Cars as Systems-of-Systems



Cars as System of Systems (II)



Bordnetz

Das Elektronik-Konzept
der neuen 7er-Reihe

Bordnetz

Highlights:

iDrive, Dynamic Drive, Elektrische Feststellbremse, ...

Bussysteme:	**4 Hauptbussystem + Subbusse**
Diagnose-Bus-Zugang:	**BMW-Fast 115kBaud**
Anzahl Steuergeräte:	**45 – 75 je nach Ausstattung**
Verwendete Prozessoren:	**im wesentlichen PPC, HC-12, C167**
Betriebsystem:	**CAN-BMW Standard-Core / MOST Referenzapplikationen** **ISIS proprietär**
Flash-EPROM:	**> 100 Mbyte je nach Ausstattung**
Anzahl elektrische Motoren:	**Bis zu 150**
Leitungslänge Kupfer / LWL:	**ca. 2,5 km / ca.50 m (ISIS 43m + MOST 7m)**
Signale auf dem Bus:	**ca. 2500**
Anzahl Lieferanten:	**mehr als 15 mit jeweils mehreren Standorten**
Powermanagement:	**Aufstart-/Abschaltverhalten und Ruhestrom**

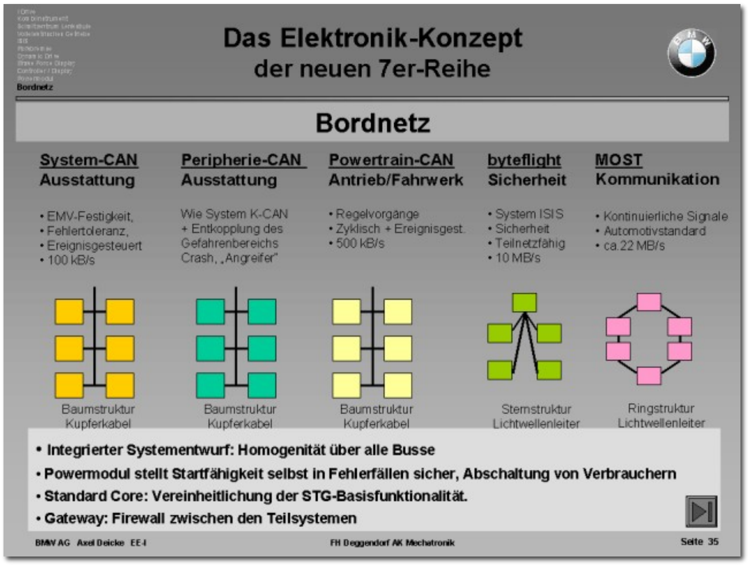
➔ Herausforderung Integration

BMW AG Axel Deicke EE-I

FH Deggendorf AK Mechatronik

Seite 34

Cars as SoS (III)



Bordnetz

Das Elektronik-Konzept
der neuen 7er-Reihe

Bordnetz

System-CAN
Ausstattung
- EMV-Festigkeit,
- Fehlertoleranz,
- Ereignisgesteuert
- 100 kB/s

Peripherie-CAN
Ausstattung
Wie System K-CAN
+ Entkopplung des
Gefahrenbereichs
Crash, „Angreifer"

Powertrain-CAN
Antrieb/Fahrwerk
- Regelvorgänge
- Zyklisch + Ereignisgest.
- 500 kB/s

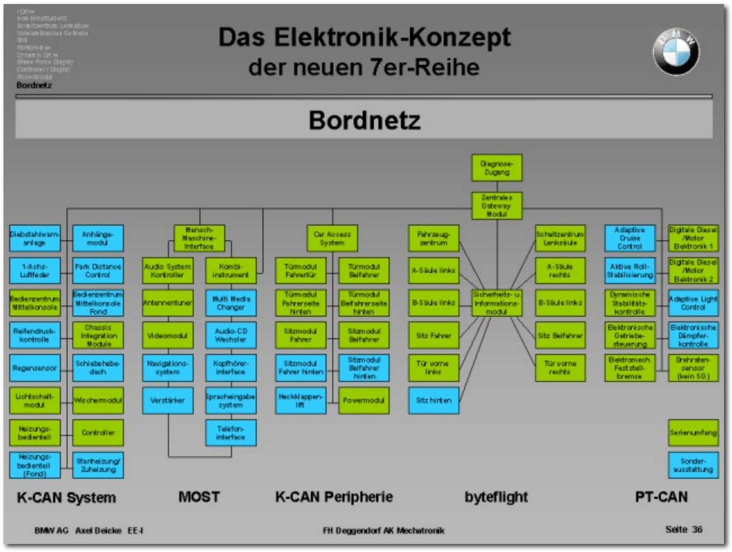
byteflight
Sicherheit
- System ISIS
- Sicherheit
- Teilnetzfähig
- 10 MB/s

MOST
Kommunikation
- Kontinuierliche Signale
- Automotivstandard
- ca. 22 MB/s

- **Integrierter Systementwurf: Homogenität über alle Busse**
- **Powermodul stellt Startfähigkeit selbst in Fehlerfällen sicher, Abschaltung von Verbrauchern**
- **Standard Core: Vereinheitlichung der STG-Basisfunktionalität.**
- **Gateway: Firewall zwischen den Teilsystemen**

BMW AG   Axel Deicke   EE-I   FH Deggendorf AK Mechatronik   Seite 35

Cars as SoS (IV)



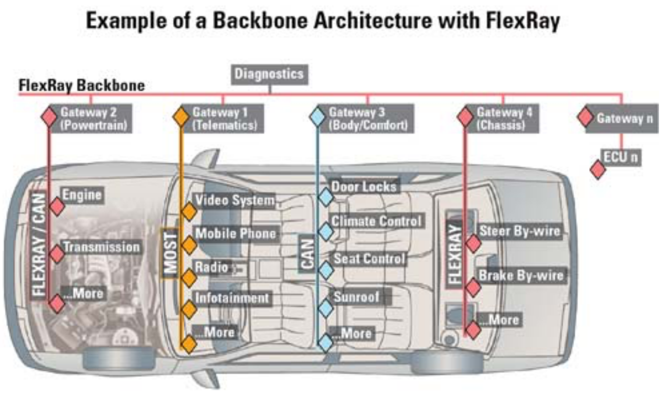
Bordnetz

Das Elektronik-Konzept
der neuen 7er-Reihe

Bordnetz

BMW AG   Axel Deicke   EE-I   FH Deggendorf AK Mechatronik   Seite 36

Cars as SoS

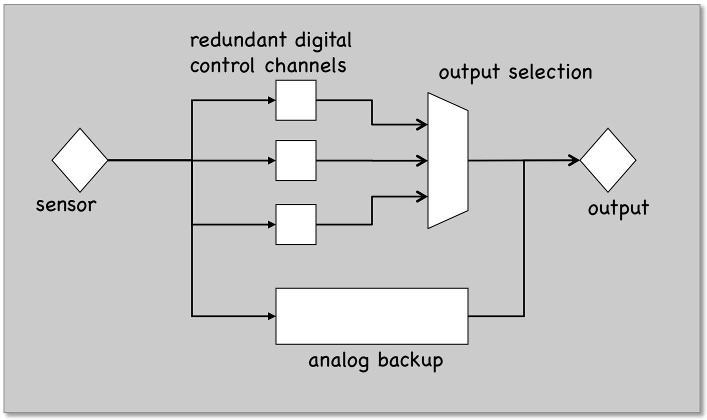


(www.autofieldguide.com)

Example: Asynchronous Design of Digital Flight Control Systems

(J. Rushby, SRI-CSL-93-07, Nov. 1993)

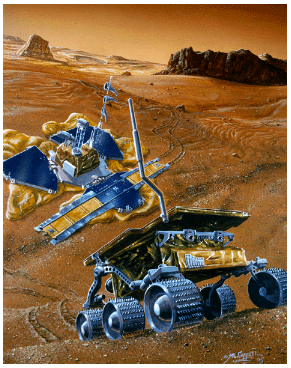
- Advanced Fighter Technology Integration (AFTI) F-16 DFCS:



Asynchronous Design of Digital Flight Control Systems

> ‘‘... The asynchronous design of the [AFTI-F16] DFCS introduced a random, unpredictable characteristic into the system. The system became untestable in that testing for each of the possible time relationships between the computers was impossible. This random time relationship was a major contributor to the flight test anomalies. Adversely affecting testability and having only postulated benefits, asynchronous operation of the DFCS demonstrated the need to avoid random, unpredictable, and uncompensated design characteristics.’’
>
> *D. Mackall, flight-test engineer AFTI-F16 flight tests*

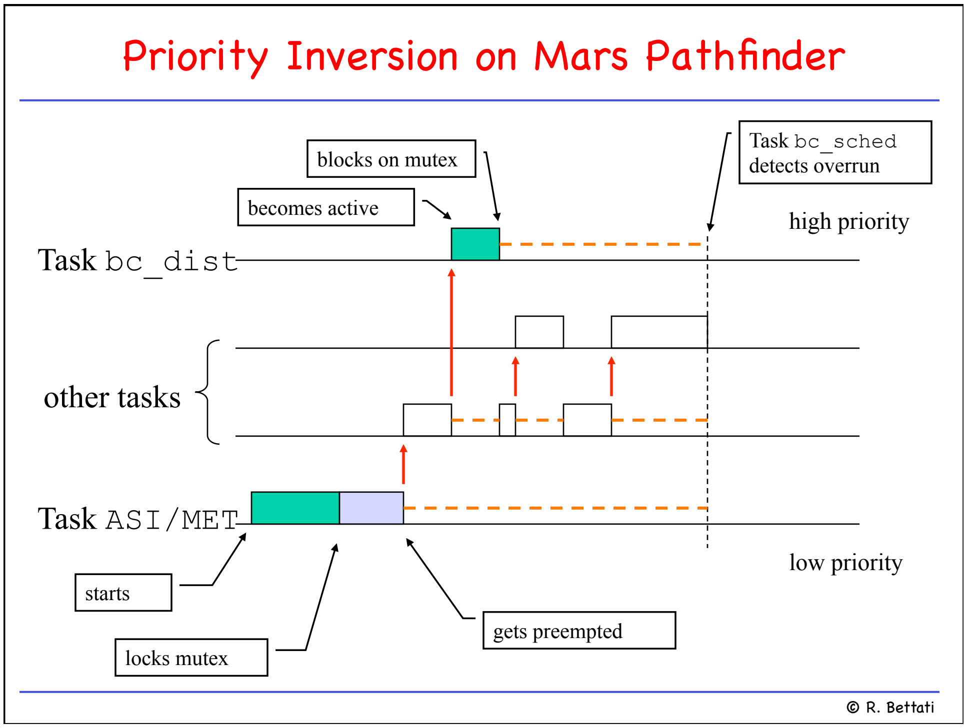
Example: Mars Pathfinder Incident



- Landing on July 4, 1997
- “experiences software glitches”
- Pathfinder experiences repeated RESETs after starting gathering of meteorological data.

- RESETs generated by watchdog process.
- Timing overruns caused by priority inversion.

- Resources:
  `http://research.microsoft.com/en-us/um/people/mbj/Mars_Pathfinder/Mars_Pathfinder.html`



Priority Inversion on Mars Pathfinder

blocks on mutex

becomes active

Task `bc_sched` detects overrun

high priority

Task `bc_dist`

other tasks

Task `ASI/MET`

low priority

starts

gets preempted

locks mutex

Real-Time *vs.* Non-Real-Time Systems

Q: What distinguishes RT systems from non-RT systems?

A: Timing Constraints!

Players in Real-Time Systems

Jobs and Processors:

- **Job**: Unit of work executed by the system
- **Processor**: Jobs require resource to execute (CPU, disk, network link)

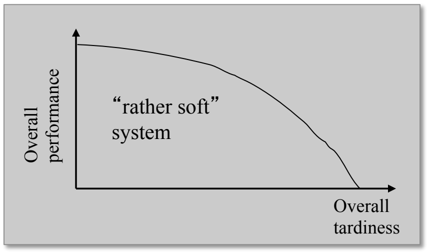
(We don't distinguish between types of processors!)

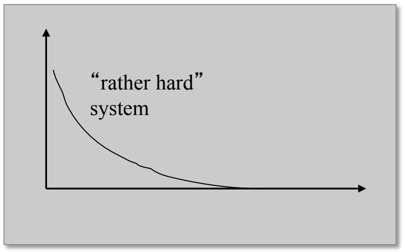
Timing constraints:

- **Release Time**: time when job becomes available for execution
- **Deadline**: time when execution must be completed
- **Relative Deadline**: maximum response time

Hard *vs.* Soft Deadlines

- **Hard Deadline:** Late result may be a fatal flaw, of little use, or cause disastrous consequences
- **Soft Deadline:** Timely completion desirable. Late results useful to some degree.
- Quantitative measure: Overall system performance as function of tardiness of jobs.





Operational Definition: A job has a **hard** deadline whenever the system designer must prove that the job never misses its deadline.

Hard Real-Time Systems

Definition: A real-time system is **hard-real-time** when a large portion of the deadlines is hard.

- Examples:
  - Embedded systems
  - Recovery procedures in high-availability systems
  - many others …

Hard Real-Time Systems

Q: Does real-time mean fast ?

Hard Real-Time Systems

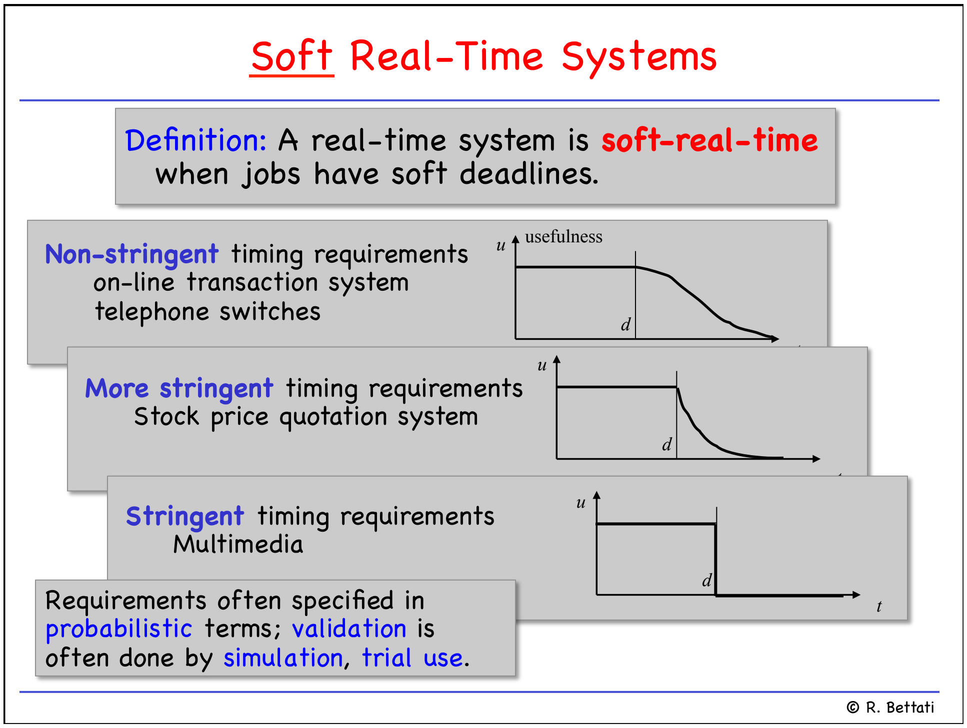
Q: Why not use commercial (general purpose) OSs?

A: Verification, Certification

Hard Real-Time Systems

Q: Why do we need to meet deadlines 100% of the time?

A:
- Validation of probabilistic timing requirements.
- Assessment of compound effect of missed deadlines with other factors.



Soft Real-Time Systems

Definition: A real-time system is **soft-real-time** when jobs have soft deadlines.

Non-stringent timing requirements
on-line transaction system
telephone switches

More stringent timing requirements
Stock price quotation system

Stringent timing requirements
Multimedia

Requirements often specified in probabilistic terms; validation is often done by simulation, trial use.