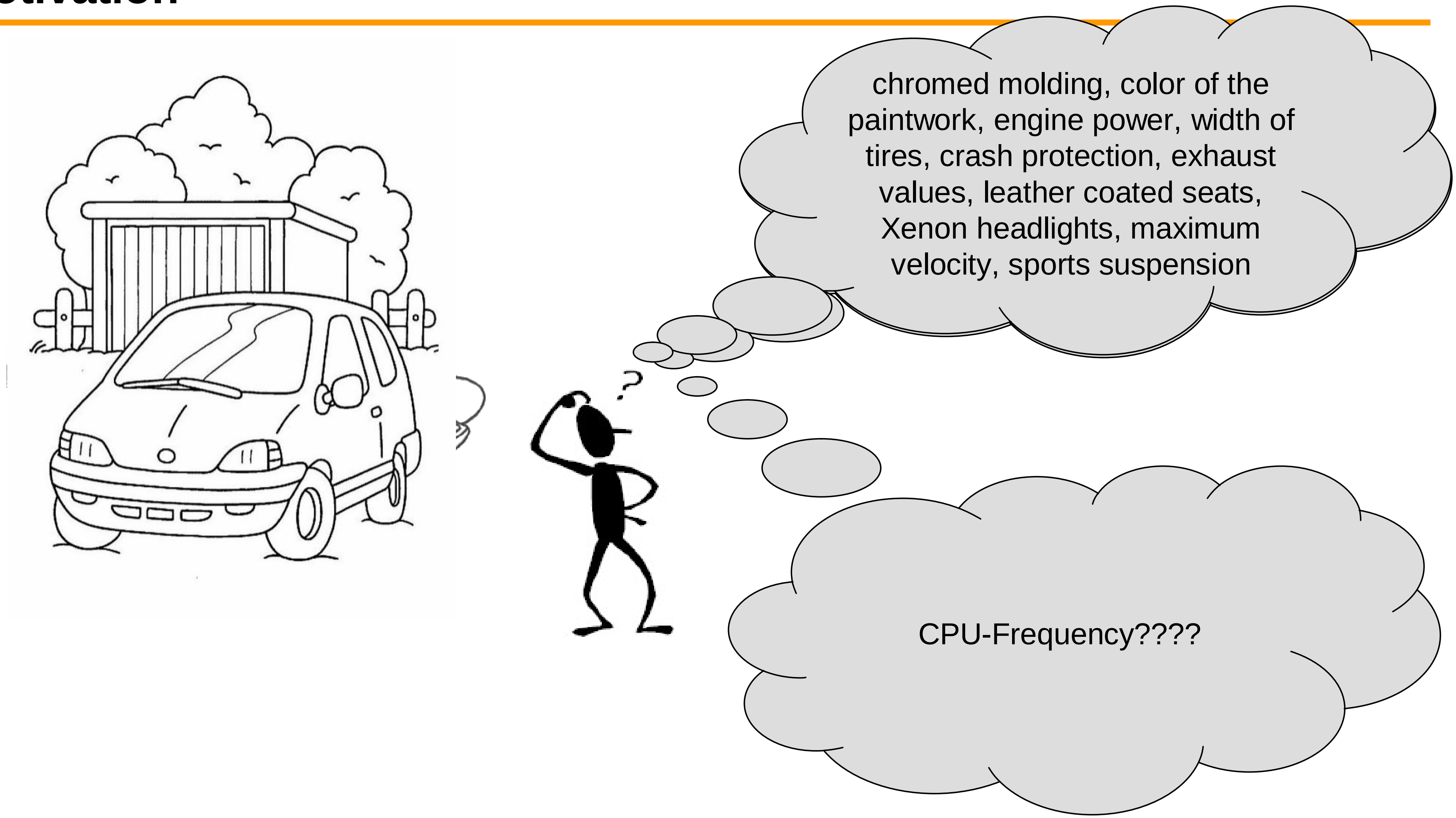
Quirks and Challenges in the Design and Verification of Reliable, Efficient, High-Load Real-Time Software Systems

Ulrich Margull	1 mal 1 Software GmbH
Michael Niemetz Gerhard Wirrer	Continental Automotive GmbH, Regensburg

Motivation


chromed molding, color of the paintwork, engine power, width of tires, crash protection, exhaust values, leather coated seats, Xenon headlights, maximum velocity, sports suspension
CPU-Frequency????

Motivation

- An engine control system (ECS) has to perform many tasks with a wide spread of deadlines ranging from less than 1 µs to several seconds
- In its core functionality, many of the deadlines ...
  - ... are quite "hard", since they are related to the rotating engine, e.g. points in time for injection and ignition
  - ... and at the same time very fast, e.g. in the µs time range for injector control
- High reliability
- High safety requirements
- Due to the market requirements and high volumes, a highly efficient resource consumption and design-to-cost principles are mandatory for system development

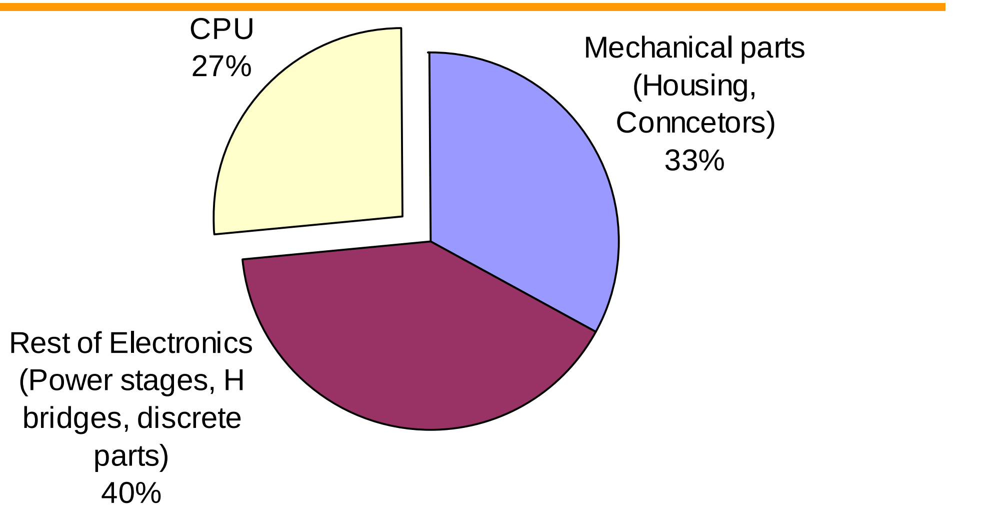
Continental

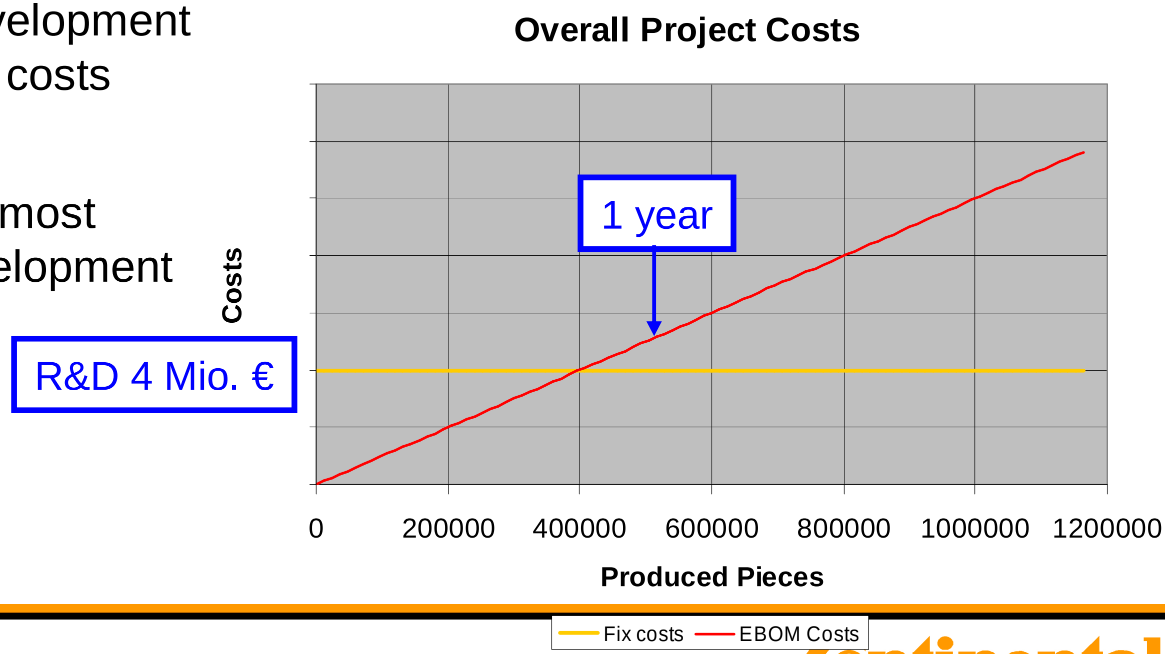
Motivation:
Why does Anybody want a 95% CPU Load?

- CPU is most expensive hardware element in the EBOM (~40%)
  - BOM = bill of materials
  - EBOM = BOM of electronic parts + circuit board
- High volumes (up to the millions per year)
- Due to high volumes, software development costs are much smaller than BOM costs

=> savings in BOM will be realized almost independently of the software development costs !

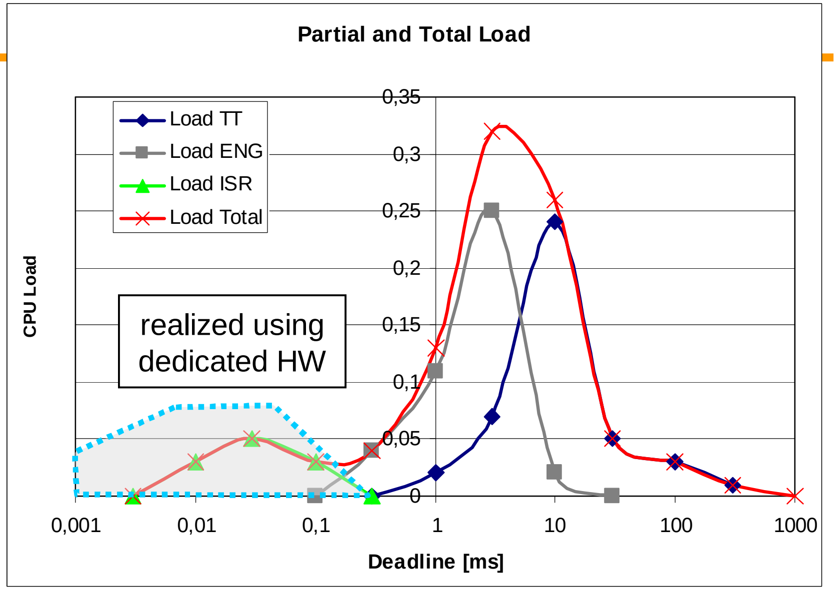
- Example:
  - 500k Pieces / year
  - 10 € EBOM

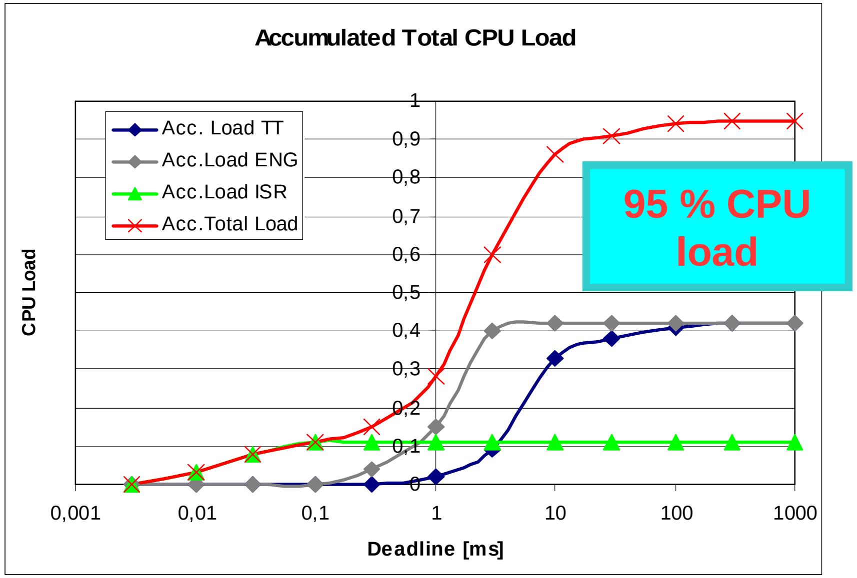




Characterization of System Dynamics

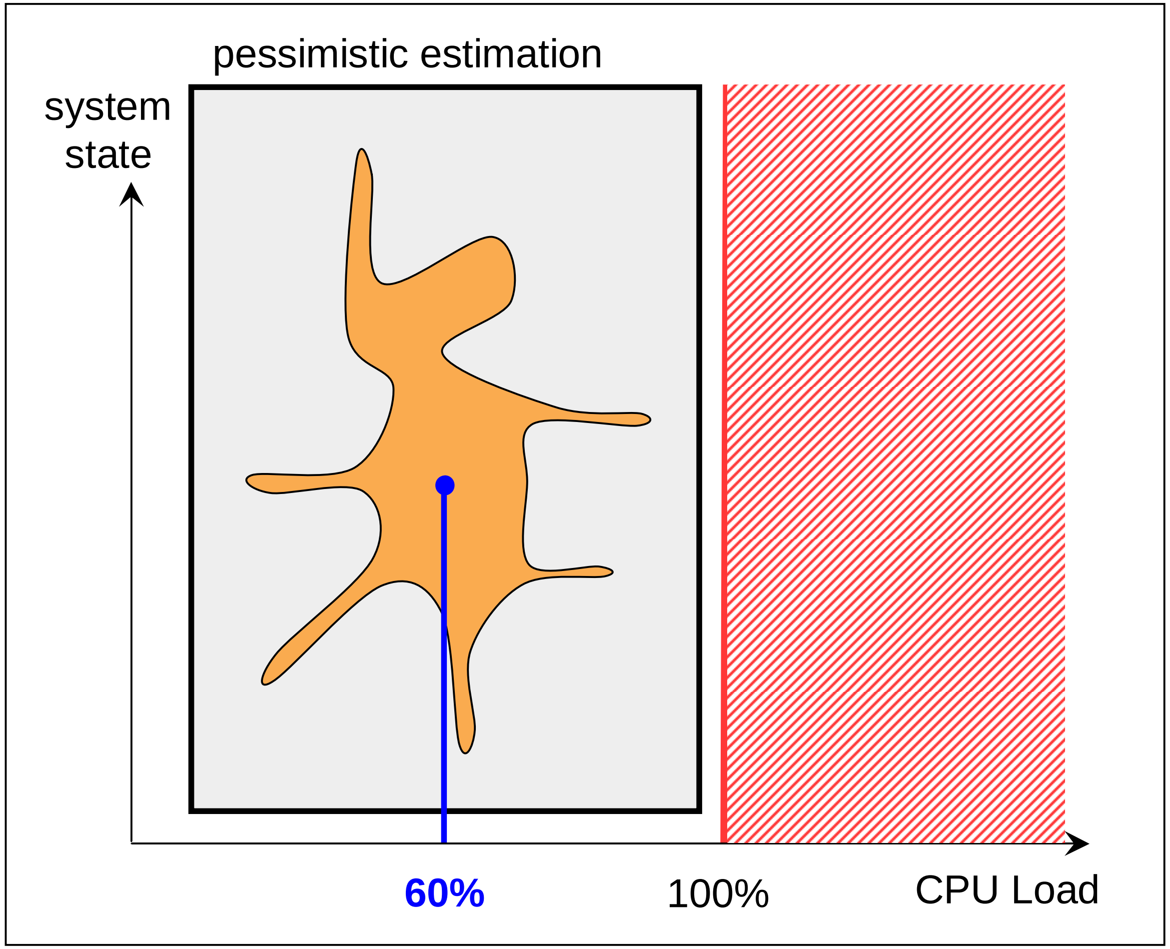
- An ECS has to perform many tasks with a wide spread of deadlines ranging from less than 1us to several seconds
- Software on the CPU can cover a range from some microseconds up to long-time calculations
- Hierarchical composition according to deadlines
  - each time range should be robust against some overload conditions
  - try hard to make each deadline a “soft deadline” (degradation of service instead of failure)





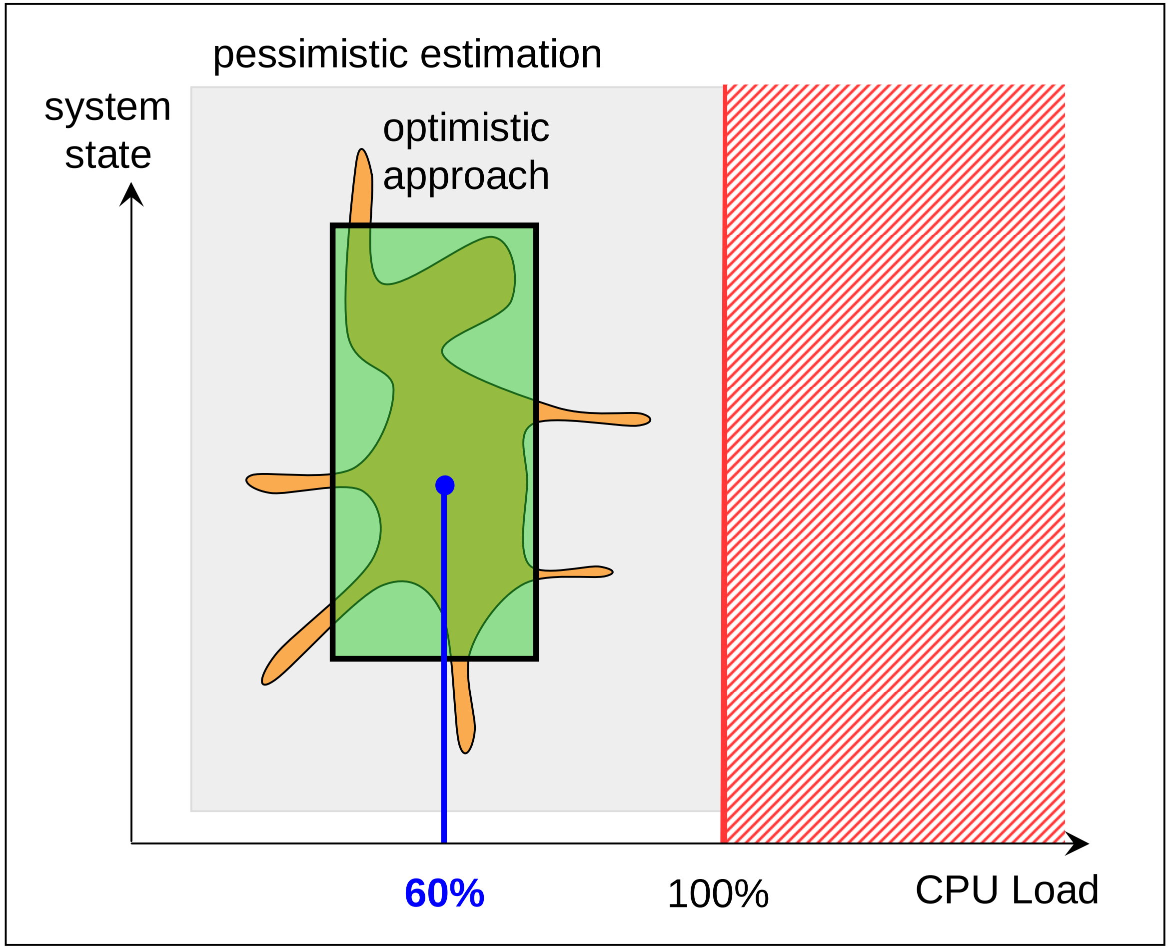
Development Challenges

- Classic analysis uses pessimistic approaches (e.g. WCET)
- Even with very exact estimates only a medium average CPU load (e.g. 60%) can be achieved



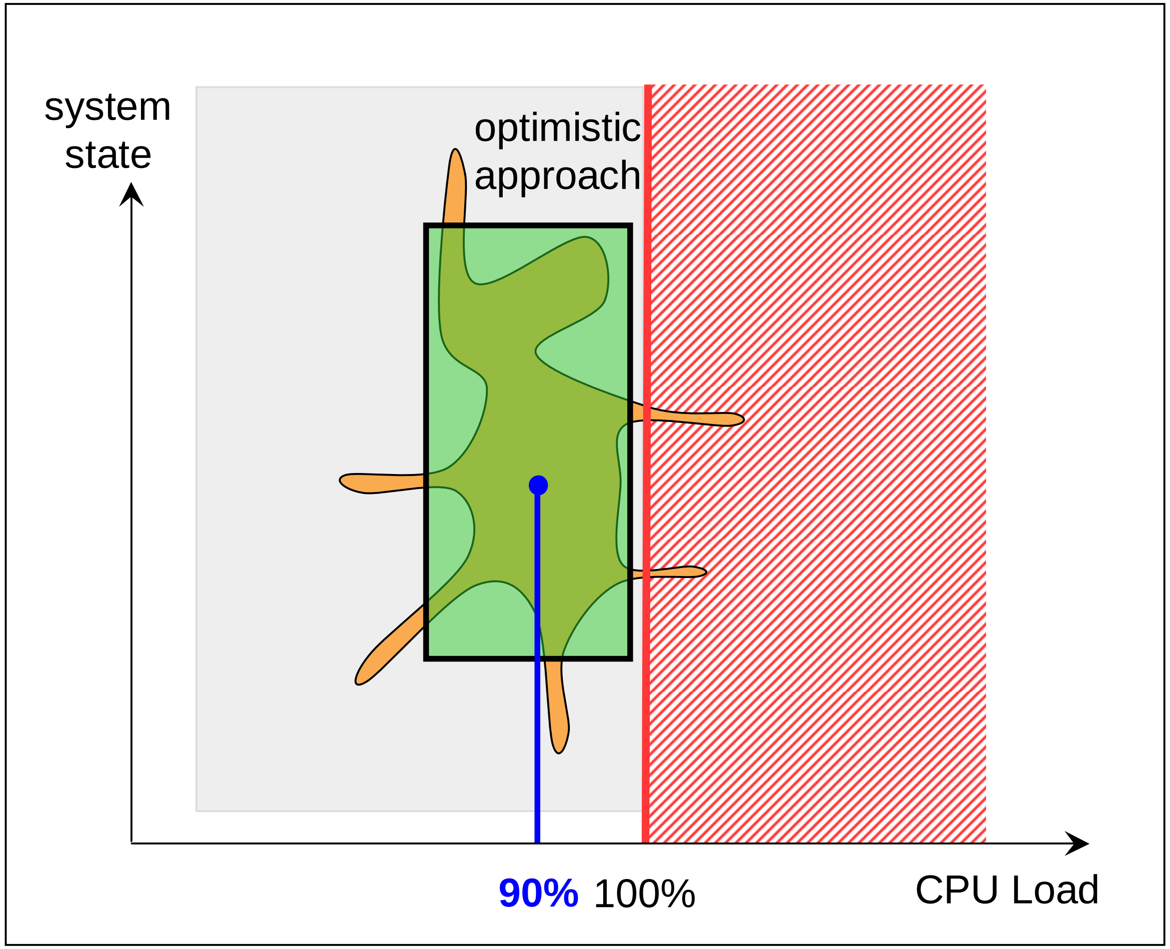
Development Challenges

- System must be robust way, i.e. temporary overload
- Using an optimistic approach, most of the system state is covered, but not all
- This allows a much more efficient system design



Development Challenges

- Using an optimistic approach, most of the system state is covered, but not all
- This allows a much more efficient system design
- and higher average CPU loads
- Two-fold advantage:
  - ... smaller CPU (for the functionality)
  - ... or more functionality (with the same CPU)



Example:
Validation of the Timing Behavior of an ECU using Simulation

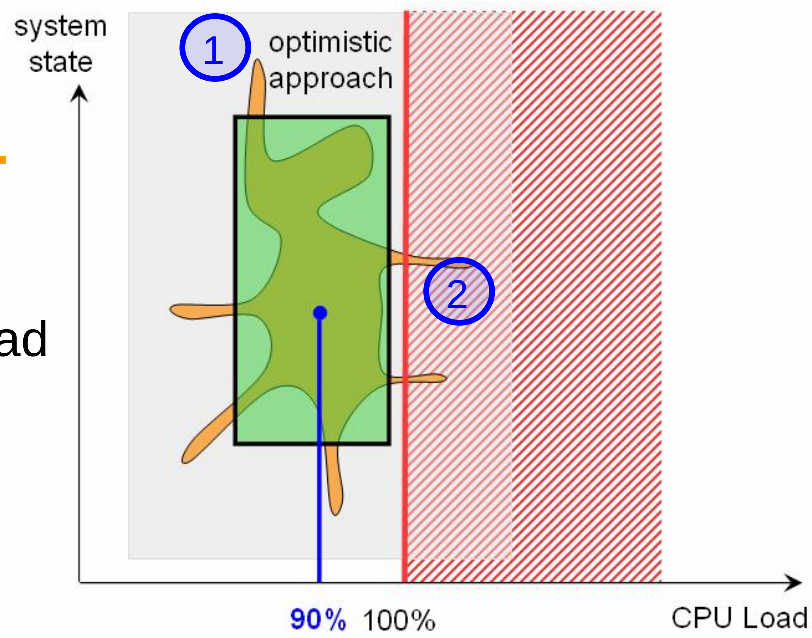
- Build a model of the system (choose the necessary abstraction level)
  - Based on OSEK OS
  - Tasks replaced stubs using the measured runtimes (***average*** value)
- Analyze the behavior using simulation techniques

Optimistic approach

  - Use suitable metrics to quantify the simulation results
    F. König, et.al., *Application Specific Performance Indicators for Quantitative Evaluation of the Timing Behavior for Embedded Real-Time Systems*, Date 2009
- Benefits
  - High flexibility: existing software can be modeled with the necessary complexity (e.g. mix of preemptive / cooperative behavior, mix of different scheduling approaches, correlation between software behavior, sporadic behavior of calculations)
  - Increased reliability due to stress tests
  - Better understanding of internal dynamics: simulation gives a "white-box" view of the system

R. Münzenberger, et.al., *Entwurf echtzeitfähiger Steuergerätesoftware in FlexRay-Netzwerken*, KFZ Entwicklerforum 2007

Development Challenges



Important questions

- How "robust" is the system against temporary overload conditions ?
- How large are the areas that are not covered by the “optimistic approach” ? (1)
- What is the the impact of those areas ? (2)
- How can the areas be minimized or completely removed ?
- How can the “optimistic approach” be improved ?
- What are good design principles when using the optimistic approach ?
- What are good validation principles when using the optimistic approach ?

Summary & Required New Concepts

- Highly efficient hard real-time systems are possible in the Automotive domain with
  - ... CPU loads up to 95%
  - ... high reliability as well as strong safety requirements

However, research is needed to support the development of

reliable **and** cost-efficient real time systems.

Involved fields:

- Design principles
  - Classification of systems (cyclic, acyclic, ...)
  - Algorithms (functional algorithms, scheduling strategies)
  - Hierarchic deadline realization
  - Hardware / software co-design
- Verification/Validation concepts

Thank you for your attention