AUTOSAR provides a modular and flexible software integration platform

- necessary step towards modularization and platform independence
AUTOSAR and timing

- the AUTOSAR software architecture is to a large part based on a client-server mechanism
  - introduces hidden timing dependencies (see talk by K. Richter)
  - well known problem from research
  - result of platform properties
  - simpler send-receive mechanism does not help either if response times of communication are not known
  - no solution in AUTOSAR (so far)
Consequences?

• timing dependencies are mapping dependent
  – challenges platform independence
  – challenges portability
  – challenges real-time behavior
    • hidden jitter
    • hidden delays
    • lost messages, ...

• the dependencies are fundamental and will not disappear with time
  – AUTOSAR software implementation cannot solve architectural shortcomings
  – FlexRay helps but is not sufficient
    • gated networks, local ECU software architecture, optimization challenges
What can we do?

- **solution 1: Be conservative**
  - put everything under a global time triggered strategy
  - performance issues, cost issues, integration issues
What else can we do?

• **solution 2:** Use formal models and strategies to control timing
  – use advanced, predictable and adaptable scheduling and arbitration concepts
    • network management for controlled jitter and delays
    • adapt software implementation
  – avoid integration legacies
    • use platform independent parameters rather than “once-and-for-ever-fixed” time slot and priority assignments
  – analyze and adapt the system carefully
    • include global analysis
    • requires appropriate models and tools
  – establish timing and QoS contracts between suppliers and OEMs to control overall timing behavior and service
Formal techniques - Revolution or evolution?

- most basic data are available
  - communication volume, buffering and driver strategies, software execution and response times
  - *if they are not available – how about real-time assumptions today?*

- AUTOSAR introduction can pave the way
  - software architecture must be complemented by a system timing view
  - automotive platform planning is much more systematic if supported by a global timing view
  - timing contracts between AUTOSAR software suppliers, ECU suppliers and OEM would make design much more transparent (*liability in case of real-time violations today?*)

→ an engineering evolution
but a cultural change in design process management
So AUTOSAR is in good shape?

Not really ..

• there will be much software developed now that does not adhere to or is qualified according to timing standards
  – how will global timing be determined in a more complex network of suppliers?
  – is this the timing legacy software of tomorrow?

• AUTOSAR urgently needs a timing standard NOW

• and finally some food for panel controversy

The revolutionary step would be a systematic consideration of realistic hardware timing and execution platform control strategies in software engineering