Beyond AUTOSAR
Robustness to Change

Alberto Ferrari
Deputy Director
PARADES GEIE - Rome - Italy

Alberto Sangiovanni Vincentelli
The Edgar L. and Harold H. Buttner Chair of EECS
University of California at Berkeley
Scientific Director
PARADES GEIE - Rome - Italy

with contributions from General Motors
Key Issues

- Commonize as much as possible electronic platforms
- Optimization and integration
- Robustness to change
  - Include fail-safe, fail-soft issues
- Need for a virtual integration environment that allows the architect to take advantage of the architectural degrees of freedom and efficiently analyze the impact of the changes.
Strategy for Commonization

• Potential areas of commonization
  – Process
    – Development and deployment
  – Architectures
    – Functional architecture
    – Subsystem architectures
  – Hardware architecture
  – Software architecture
  – Components
    – ECU components
    – Software components
    – Sensor/Actuator components
AUTOSAR: decoupling functionality from platform

- Mapping performed at design time:
  - Require non-functional information
  - Optimize solutions for known application/architectural space
- Control architecture not addressed
Rich component models: decorate components with non functional views

• Time views:
  – WCET, state based ET
  – WCCT

• Safety views:
  – Fault masking & detection
  – Fail silent, fail operational behavior

• Power views:
Beyond AUTOSAR: Robustness to change

• Capability to adapt to changing conditions
  – Changing conditions known at design time,
    – Solved at design-time
      – E.g. product variants
    – solved at run-time
  – Changing conditions unpredictable at design time:
    – minimize sensitivity at design time
      – Extensibility, scalability (incremental mapping)
    – solved at run-time
      – Run-time adaptability
Robustness to change: designing for dynamic integration

System Requirements  System Decomposition  Component Implementation

Static Compositions

Component Integration

Dynamic Compositions

System Integration Prod-A
System Integration Prod-B
System Integration Prod-C
System Integration Prod-D

Maintenance Prod-A/2
Maintenance Prod-B/2
Communication robustness

- Focus on optimally utilize redundancies in schedules for extensibility and scalability
  - Idle time and slacks are traditionally incorporated in hard real time embedded systems schedules to increase system robustness

- We should utilize these redundancies to:
  - Tolerate incremental design changes
  - Accommodate new tasks to be added in future product updates
### Extensibility
- Tolerate changes of Task WCET
- Tolerate changes of Data WCTT
- Maintain Bus Schedule
- Maintain non-involved ECU schedules
- Maintain involved ECU schedules without reconfiguration
- Message left & Right slack
  - Max Sum of all slacks
  - Min Variance of all slacks

### Scalability
- Accommodate NEW tasks by statically scheduling them on a legacy system
- Provide blocks of computation time for future computation intensive tasks
- Provide porosity in schedules to allow for future tasks with tight deadlines
- ECU idle time distribution
- Bus idle time distribution
  - Evenly distribute all idle time

---

**Extensible and Scalable Time Triggered Scheduling for Automotive Applications,** Wei Zheng, JiKe Chong, Claudio Pinello, Sri Kanajan, Alberto Sangiovanni-Vicentelli - ACSD-05
Adapting to change

• Reconfiguration of software (hardware) and communication mapping
  – At Initialization: components agree on the software task and communication mapping
  – Maintenance and component reuse

• Run-time adaptability:
  – Components agree on new mode of computation and communication at run-time
  – Robustness to faults
Thanks