Dependability in Automotive Systems
-- Verified Committee Meeting --

28 May 2010

Martijn van den Heuvel and Reinder J. Bril
Technische Universiteit Eindhoven
Mathematics and Computer Science
System Architecture and Networking
{m.m.h.p.v.d.heuvel, r.j.bril}@tue.nl
http://www.win.tue.nl/~mheuvel, ~rbril/
Who are we?

• Systems Architecture and Networking:
  – Parallel and distributed systems
  – Resource constrained embedded systems

• Quality aspects:
  – Performance
  – Predictability
  – Dependability
  – Programmability
  – Security

• More info: http://www.win.tue.nl/san/
Contents

• Overview Workshop CARS at EDCC2010

• Relevance of reservations for VERIFIED

• Achieved Milestones

• Roadmap
CARS@EDCC2010

• Integration of Matlab and ASD:
  – *Presented paper:*
    • *Verification-based development of In-Vehicle Safety Critical Software: A Case Study,*
      Workshop Critical Automotive Applications: Robustness Safety, April 2010

• Keynote talk from Christoph Jung
  – ISO 26262: Challenge or Chance in Automotive Industry

• Key topics:
  – Safety and Development process (traceability)
  – Design and Verification (formal models / code generation)
  – Scheduling (Time-triggered / Mode changes)
  – Robust Architectures (Temporal and Spatial Isolation)
• Key topics:
  – Fault injection
  – (Fault) Modeling techniques
  – Run-time mechanisms for fault-tolerance
  – Distributed Protocols
  – Diagnoses and monitoring techniques
  – Fault-tolerant hardware design
  – Approaches and Methodologies
Our Challenges towards Robustness

• *Isolation*: applications shall not “interfere”
  – *Temporal* isolation: processor and bus;
  – *Spatial* isolation: memory.

• *Development and analysis versus integration*
  – Independent analysis of application on “virtual” platforms;
  – Application specific scheduling algorithms;
  – Applications may also share *logical* resources;
  – Composition of applications and virtual platforms.

• *Evolution* rather than revolution
  – Adhering to de-facto industry standards (e.g. FPS);
  – Extension of existing RTOS.
Relevance for VERIFIED

• Control loops:
  – Fluctuations in delay may cause serious problems
  – Bounding delay and jitter is therefore essential,
  – hence, real-time support is required, such as
    • An RTOS;
    • Corresponding analysis.

• Reservation-based resource management
  – provides support for robustness.
  – bring together:
    • real-time, and
    • component-based development
Achieved Milestones (1/2)
(period Sep 2009 – May 2010)

• Extension of COTS RTOS
  – \(\mu\)C/OS-II, currently available:
    • simulation environment
    • 2-level FPPS scheduling
    • Global EDF scheduling
    • Two-level Synchronization based on SRP
      – i.e. local level: SRP;
      – global level SIRAP, HSRP, BROE
    • More: http://www.win.tue.nl/~mholende/ucos/
  – Accepted paper (co-authored):
    • Tracing, Visualizing and Measuring the Behavior of Real-Time Systems, Workshop on Analysis Tools and Methodologies for Embedded and Real-time Systems (WATERS), July 2010

• Virtual Timer management
  – Accepted paper:
    • Virtual Timers in Hierarchical Real-time Systems, (WiP) session of the 30th IEEE Real-time Systems Symposium, December 2009
Achieved Milestones (2/2)
(period Sep 2009 – May 2010)

• Inter-application Synchronization:
  – Accepted papers:
    • Extending an HSF-enabled Open Source Real-Time Operating System with Resource Sharing,
      Workshop on Operating Systems Platforms for Embedded Real-Time Applications (OSPERT), July 2010
    • Protocol Transparent Resource Sharing in Hierarchically Scheduled Real-Time Systems,
      IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), September 2010

• Integration of Matlab and ASD:
  – Accepted paper:
    • Verification-based development of In-Vehicle Safety Critical Software: A Case Study,
      Workshop Critical Automotive Applications: Robustness Safety, April 2010
Main Roadmap
(period May 2010 – May 2011)

• Hierarchical Scheduling:
  – Integration of Time-triggered approaches
  – Design and analysis of application interfaces
    • Capture application’s timing requirements
    • Capture application’s resource requirements
  – Integration of (legacy) COTS applications
  – Multi-resource management:
    • E.g. include Flexray scheduling
Roadmap - Continued

- Multi-resource scheduling
  - E.g. multi-processors, memory, bus, network

- Modes of operation
  - Inter- and intra-application
  - Redistribution of resources

- Criticality versus optimal resource usage

- Multi-level Hierarchical Scheduling