ARAMI S II – Project Overview

Prof. Dr.-Ing. Dr. h. c. Jürgen Becker, Karlsruhe Institute of Technology (KIT)
Increase of Electronic Systems (HW + SW) is required

- ... to integrate additional features
- ... to meet environmental challenges
- ... to enhance competitiveness
- ... to improve cost efficiency

Degree of automation will directly depend on embedded computing power!
Embedded Computational Performance

- Singlecore will not provide enough computing power in the future (scaling is over)
- Multicore is the best known solution that is able to provide sufficient performance

Quota of deliveries based on multi-core CPU at VW/AUDI (not yet in safety critical applications):

[Source: The Quest for More Processing Power: “Is the single core CPU doomed?”, Johan De Gelas, 2006]


[Source: Shared SW development in multi-core automotive context, L. Michel, et. al, 2016]
Key Enabling Technology Embedded Multicore

- Singlecore will not provide enough computing power in the future (scaling is over)
- Multicore is the best known solution that is able to provide sufficient performance

Future System Aspects and Characteristics

- Functionality is safety critical in many cases
- Systems are highly connected in networks
- Evolutionary Development (Legacy Code)
- User friendliness and smart ecosystems: Use of APPS
- Cyber Physical Systems with mixed criticality and dynamic reconfiguration of structures and services
But: Multicore comes with challenges ... 

Common resources shared between different execution units can lead to system dysfunction (loss of functions / malfunctions) caused by:

- Time interferences (determinism issues)
- Space interferences (segregation issues)
- Common Cause Failures (e.g. SEE, electronic fail)
- Race Condition (Time-of-check vs. Time-of-use)

Issues depend on multicore architecture:

- Mono-Bus / Multi-Bus / Crossbar / NoC / etc.
- Core local memory or only shared memory
- Lock-Step-Mode core / end2end ECC / etc.

Mitigations needed for safe and secure usage (per SW or HW):

- **Failure Detection**: Monitoring, Voting
- **Failure Isolation**: Partitioning, Time Slicing / Deadlines, Budgeting
- **Failure Correction**: Function Recovery, Redundancy, Architectural Patterns
Summary on Results of ARAMiS

- Improvement of Basic Software Architectures
- Improvements of platforms on system, hardware and software level.
- First Results on holistic tool support
- Work on Methodologies
- Prototypical implementations and evaluations in laboratory setups
- Demonstrators as feasibility studies and proof for deployment of multicore systems in real industrial environments

ARAMiS proved successfully the applicability of multicores in Safety-critical applications in principal...

... but uncovered further challenges in multicore development
ARAMiS proved the applicability of multicores in safety critical applications in principle.

ARAMiS II targets the efficient use of multicores in safety critical applications in practice by preparation of:

- Structured Multicore Development
- Multicore Methods and Tools
- Industrial Platforms for Multicore Systems
Challenges for Multicore-System Development

1. Separated steps in multicore development are not sufficient for a structured development of multicore-based systems
   - Process: How could a superior (generic) multicore development process look like?
   - Continuity: How can continuity in the process be achieved and which artefacts are needed?

2. Available methods and tools are not sufficient to master the complexity in the development of multicore-based systems
   - Partitioning: When and where to split and distribute functionality?
   - Allocation: Which could be the right platform for a certain application scenario?
   - Binding: Which deployment of (basis-) software components is the most optimal solution?
   - Scheduling: Which schedule of software can be run most efficiently?
   - Guarantees: How can platform aspects (e.g. WCET, Safety, Security, correctness) be ensured?
   - Design Space: How can a design space exploration be performed in such complex systems?

3. Well established platform standards and software architectures are not supporting the requirements of multicore-based systems (e.g. segregation, synchronization, communication)
Summarized Working Focus and Project Goals

**STRUCTURED MULTICORE DEVELOPMENT**
Provision of systematic and structured approaches for the development of multicore software and platforms

**INDUSTRIAL PLATFORM ARCHITECTURES**
Development and extension of established industrial platforms with respects to multicore requirements

**METHODS AND TOOLS**
Development of methods and tools supporting the structured multicore development
Subproject 2: Structured Multicore Development

- Definition of a generic and seamless development process for multicore systems
- Model based “Top-Down” development process, avoiding unnecessary iterative loops...
- ... but considering bottom-up and legacy aspects
- Implemented by methods and tools developed in ARAMiS II
Subproject 3: Multicore Methods and Tools

- Development of specific methods and tools to support the structured multicore development
- Extension of methods for all steps in the development process (e.g. partitioning, deployment, scheduling, design space exploration)
- Higher degree of automation in the development due to tool support
Subproject 4: Industrial Platforms for Multicore Systems

Development and extension of established industrial platforms for the use in multicore-based systems

Investigation of basis software, middleware and operating systems

Evaluation and development of lightweight fail-operational concepts for multicore platforms
Involved Domains for the Validation of the Results

- Automotive
- Avionics
- Industry Automation
Subproject 5: Use Case Implementation and Evaluation
• Coordination: Karlsruhe Institute of Technology (KIT)
• Duration: 10/2016 – 09/2019
• Consortium: 33 Partner
• Budget: > 26 Mio.€
• Web: www.aramis2.de
• Publications up to now: >40 (www.aramis2.de/publikationen)
Thank you for your attention!

Prof. Dr.-Ing. Dr. h. c. Jürgen Becker (becker@kit.edu), Dr.-Ing. Falco Bapp (bapp@kit.edu)
Institut für Technik der Informationsvereinbarung (ITIV)
Karlsruher Institut für Technologie (KIT)
Engesserstrasse 5, 76131 Karlsruhe