

Memorandum and Post parMERASA Vision

Prof. Dr. Theo Ungerer,

parMERASA Project Coordinator
University of Augsburg

- Target of Memorandum based on Q&A to industrial experts
 - reason about post parMERASA research topics
 - survey research topics in embedded real-time systems
 - influence future national and EC research funding

- Timeline
 - June 2014: Questions defined and industrial experts selected
 - July 2014: Answers collected and refined
 - August 2014: First internal draft of Memorandum compiled
 - September 23, 2014 at Dissemination Event:
First presentation of Memorandum to public, discussed and refined again

Six questions for future of safety-critical systems

1. Time predictable vs. COTS-based multi/many-cores.
2. Timing analysis techniques: static WCET, hybrid methods, probabilistic analysis, or just testing and adding a safety margin.
3. Application characteristic justifying parallelization of hard real-time programs.
4. Limits of multi-core scalability: Multi- vs. many-cores (>8 cores); heterogeneous or homogeneous multi-cores.
5. Start multi-core program development from scratch or reuse or transform legacy programs; parallel design patterns.
6. Research topics in safety-critical domain "beyond parMERASA".

- **15 industrial experts answered:**
- **Avionics domain:** **Airbus**, Toulouse, France; **Honeywell** International S.R.O., Bruno, Czech Republic; **Honeywell** EOOD, Sofia, Bulgaria; **Thales** Research & Technology
- **Automotive domain:** **AUDI** AG, Germany; **Daimler**, Germany; **Delphi** E&S, Sweden; **DENSO** AUTOMOTIVE Deutschland GmbH, Germany; **Elektrobit** Automotive GmbH, Germany; **Infineon** Technologies UK, Bristol, UK; (**Infineon** Technologies AG, Munich, Germany; **Volvo** Group, Sweden
- **Other:** **AIT** Austrian Institute of Technology GmbH , Austria; **BAUER** Maschinen GmbH, Schrobenhausen, Germany; **Sensortechnik Wiedemann** STW, Germany

Q1: Time predictable vs. COTS-based multi/many-cores

Time predictable multi-core systems are very important, however, not provided by major semiconductor vendors

- Analyze the obstacles for semiconductor vendors to produce time predictable hardware architectures and suggest a way forward.
- Propose predictability techniques in hardware, system software and tools to increase predictability of COTS multi-core processor.

Q2: Future of timing analysis techniques

No single generally applicable timing analysis method exists.

- Safety standards are not asking for a particular technique only requiring some level of quality.
- The well-known approach of adding a safety margin of 20% on top what is measured might not be valid for future applications.
- A mixture of several tools/methods is expected also depending on the design phase in which the tool is used.
- Tool support for WCET analysis is critical.

Q3: Application characteristic justifying parallelization of hard real-time programs

Hard real-time applications require high computing power.

- Doing more of the existing: acquiring more sensors, driving more actuators, etc.
- Implementing more elaborate control laws (safety, fuel efficiency, passenger comfort).
- Complex mathematical operations may be applied for online physical simulation.
- Increasing productivity and lowering engineering costs by relying on more abstractions that in turn “waste” some processing power.

Q4: Limits of multi-core scalability: Multi- vs. many-cores (>8 cores); heterogeneous or homogeneous multi-cores.

Having a dual-core or a quad-core in a safety-critical system is already an open problem, so the problem of scalability for the moment is not an issue.

- SW is way behind the HW and multi-core programming is still a complex topic.
- Heterogeneous systems and hardware accelerators are already used in automotive controllers.
- Need for a corresponding eco-system with tooling, debugging, compilers, ...
- Multi-/many-cores with a high core number are a research opportunity, but are not expected to be applied in near future.

Q5: Start multi-core program development from scratch or reuse or transform legacy programs; parallel design patterns

“It is always intended to reuse legacy code, and it is rarely avoided rewriting non-negligible amounts of it”

- Employ model based development and autocoders.
- Reuse legacy programs for migration to multi-core to avoid high (re-)certification cost.
- Parallel design patterns are a means to employ structured parallelism, however, suitable tools should be developed.

Q6: Research topics in safety-critical domain “beyond parMERASA”

- Develop better programming paradigms in particular languages.
- Develop tool support for assisting parallelization, mapping tools, debuggers, dynamic design assistants and so on. Tools should address combined safety/security/resilience issues.
- Investigate heterogeneous architectures supporting beyond 1000 GMACs (Giga Multiply-Accumulate Operations per Second), many-cores should be qualitatively different from today's.
- Find ways to implement parMERASA solutions on today's or tomorrow's hardware. Touch each one of the mentioned topics with more emphasis on the COTS processors.

- parMERASA showed that parallelisation of hard real-time control algorithms is feasible.
- parMERASA is the first step in this direction
 - Results can be improved
 - New more complex control algorithms can be developed
- Research perspectives summarized:
 - Research topics for future safety-critical systems concern tooling and languages for current COTS hardware, but also new multi-core architectural solutions for future high-performance applications.
 - Envisioned autonomous and connected cars, remotely piloted aircrafts, and formation flying create high-performance demands from interactions with physical environment beyond the systems available today.