From AUTOSAR Adaptive to a Safe Level 4 ADAS Platform

Christopher Helpa
Functional Safety Engineer
Christopher.Helpa@tttech-automotive.com
Contents

✔️ ADAS ECU - TTIntegration Platform Approach and Validation Support
✔️ Experience from Level 3 series projects
✔️ TTIntegration and AUTOSAR Adaptive
✔️ The fail-operational Level 4 challenge
✔️ Putting it all together
ADAS Domain ECU Architecture – Scalable and Flexible Architecture
Ad Hoc Software Architecture

- Different APIs, no SWC portability
- Inhomogeneous basic services, tool and integration landscape
- "Manual" interface adaptation between SWCs
Integrated Software Architecture – Integration and Validation Support

- Uniform basic services on all Hosts
- SWCs can be moved between Hosts
- Integrated Tooling and SWC integration process
PC-based SW-in-the-Loop Simulation (SIL)

- Identical SWC-Code on PC and ECU – SWC only validated once
- Timing accurate PC-Cosimulation (ADTF, Matlab/Simulink)
Debugging & Tracing – Validation Support using Open Standards

Absolut observability with standard tooling without interference

IEEE 802.1 AS / gPTP synchronization

IEEE TSN

Deterministic Ethernet Switch

Autosar OS

Safety RTOS

Middle-
ware

MCAL

RTE

SWC 1

SWC 2

SWC 3

…

SWC n

…

SWC k

…

SWC l

General Purpose µC

GPU

RTE
Software Integration of Complex Real-Time Systems

1. Integration of platform without configuring execution frames

2. Applications are integrated and tested individually by SWC suppliers without any timing restrictions

3. All applications are integrated by the SW-integrator on the platform; conflicts start immediately as it is not clear who is causing problems and why

4. Conflicts are reported back to function SW suppliers, applications have to be modified to meet the system’s timing restrictions
Modular Platform - Reduced Integration and Validation Efforts

Integration process massively accelerated

- Iterations are avoided

1. Platform configuration and application scheduling
2. Single SWC test within configured schedule
3. SWCs are instantly running together ("composability")
Insights from Serial Development

Providing a clean platform architecture requires only very little HW-Overhead (Switch)

Middleware features are intensively used (flexible SW allocation, PC co-simulation, debug features)

A clearly defined integration process is very helpful for the work split between suppliers, dependencies and release organization
Next Generation – Diverse Application Interfaces with AUTOSAR Adaptive

- RTE
- Autosar OS
- BSW
- MCAL
- Automotive μC

- RTE
- Middle-ware
- POSIX compatible OS (Linux, VxWorks)
- MCAL
- Performance SoC
- Core 0
- Core 3

- RTE
- ara::com
- Middle-ware
- Adaptive ASR stack

- Deterministic Ethernet Switch
Next Generation – Diverse Application Interfaces with AUTOSAR Adaptive

- For absolute determinism and highest safety requirements
- QM: Best Effort/Rate Constraint
- For “some” safety:
  - Dynamic behaviour during development
  - Static configurations for increased determinism and safety in the vehicle
Next Generation – Diverse Application Interfaces with AUTOSAR Adaptive

- Enables mixed criticality network traffic
  (coexistence of best effort and safety related traffic)
Automated Driving - System Classification by VDA

- **On the market today**
  - Level 0: Driver only
  - Level 1: Assisted
  - Level 2: Partly automated
  - Level 3: Highly automated
  - Level 4: Fully automated
  - Level 5: Driverless

- **SOP 2017**
  - Planned for 2020 ff
  - System handles all situations automatically throughout the trip. No driver needed.

Delegation of responsibility from driver to the car.
Level 4 - Safety Requirements

- Level 4 automated driving requires ASIL D
- No high performance compute chips available for ASIL D
- High speed Level 4 automated driving required fault tolerance ASIL D

![Diagram showing Level 4 automated driving systems and safety requirements]
The Fail-Operational Requirement

**Level 4:** “System can handle all situations automatically in the specific application case” [VDA] → Even in the case of component failures!

- Autonomous lane change to the side and stopping
- Or Stop the car (avoiding obstacles)
Why is it Difficult?

Airplanes are fly-by-wire and fail-operational …
… but they use triple or quad redundant diverse electronics
And they have very strict software development requirements

TTTech provides the fly-by-wire network for Embraer E-Jet E2 Family
Why is it Difficult?

- Boeing 777 Flight Computer
- 3 x 3 Architecture
- Hardware diversity
- "Perfect" Software – No ASIL decomposition like mechanisms
ISO 26262 Approach

Fault avoidance according to ASIL

Systematic Faults

Software
- Processes more strict for higher ASIL
  - Examples
    - Strict traceability
    - MC/DC test coverage
    - Formal methods

Hardware
- Processes more strict for higher ASIL
  - Examples
    - Formal reviews
    - Fault injection tests
    - Analysis

Random Faults

Hardware
- Lower fault probability for higher ASIL
  - Examples - self checking design
    - Dual core lockstep
    - ECC
    - Clock supervision

ASIL: Automotive Safety Integrity Level
Design Corners For Random Hardware Faults

1oo2
one out of two

- Components need to be fail-silent or self-checking (i.e., deliver correct result or no result at all)

2oo3
two out of three

- Components can fail arbitrarily
- The voter becomes the critical element and a single point of failure
- Components can be ASIL B (high performance) but the voter becomes ASIL D (embedded safety)
Solution Sketch Systematic Faults – Fail-operational ASIL-D and Diversity

- In most cases diversity is more costly and more effort compared to a single version with a higher ASIL level
- Unfortunately many COTS SW/HW components are not available as ASIL D
- But many COTS components are available from multiple suppliers (Operating Systems, SoCs etc.)
  - No need to raise to fail-operational ASIL D due to diversity opportunity
- The “right” compromise is clearly application specific
The New Problem for fail-operational Architectures:
Example insufficient ASIL D Decomposition

Bugs in QM components are detected but message is still unusable

System is only “QM” concerning availability of the software
The Fail-Operational Challenge

- Find the optimal architectural compromise for cost, safety and performance by selecting
  - the best suited hardware fault-tolerance mechanisms,
  - the most appropriate ASIL decomposition
  - considering diversity where necessary or beneficial
  - based on the best suited set of sensor and SoCs
  - considering scalability
Fail-operational Redundancy Integration: Fail-operational as a Simple Add-in (1)

High Integrity Main Node enables (fail-silent) ASIL D commands

Integration Method: „Apply output from Partition A unless output is unusable - only then use output from Partition B“
Random/Systematic HW Fault Solution: Partition Independence (1)

Sensors

Independent Fault-containment regions with electrical decoupling, diverse sensors and diverse power supply
Random HW Fault Solution: Partition Independence (1)

Sensors

System architecture allows use of “high FIT” performance SoCs
Systematic Software Faults – Example Application Architecture Solution

TTTech

www.tttech.com
Platform Architecture for Level 4 Automated Driving (1)

- Flexible combination of SoCs according to the necessary performance and ASIL level
- Flexibility to support different redundancy, and diversity schemes

TTIntegration Middleware

- Autosar OS
- RT-OS: VxWorks Autosar Adaptive
- Rich OS Linux, Autosar Adaptive
- RT OS Vision Acceleration

Communication Backbone (Deterministic Ethernet TSN, 1 Gbit/s)
Platform Architecture for Level 4 Automated Driving (2)

- Flexible combination of different operating systems according to needed features and ASIL level
- Possibility for diverse operating systems use
Next Generation Platform Architecture (3)

- Flexible combination of SWCs with different ASIL levels
- Coexistence of fail-silent and fail-op SWCs without interference
- Diverse application interfaces (POSIX, AUTOSAR (Classic, Adaptive))
- Diverse and replicated SWCs will be supported
- Middleware will fulfill fail-op ASIL D

Communication Backbone (Deterministic Ethernet TSN, 1 Gbit/s)
TTIntegration for fail-operational Level 4 Autonomous Driving

We do not know what the OEM’s Level 4 application will look like -
But the next generation TTIntegration will support it by providing a
flexible fail-operational ASIL D execution environment

Thank you for your attention