The Safe State: Design Patterns and Degradation Mechanisms for Fail-Operational Systems

Alexander Much
2015-11-11
Agenda

• About EB Automotive
• Motivation
• Comparison of different architectures
• Concept for an 1oo2D software systems
• Summary
EB: Software and Services

**In-Car Infrastructure**
- EB tresos – integrated software and tools, based on AUTOSAR standards
- **Solutions for: operating systems, middleware, dependable communication**
- **Solutions for high integrity systems: reliability, functional safety and security**
- Test & simulation

**Infotainment**
Connected navigation software
- HMI tools for in-dash, digital instrument clusters and head-up displays
- Global software integration and engineering services

**Connected**
- Connected experiences around urbanization and electrification
- Online diagnostics
- Software and content updates

**Driver Assistance**
- Software development for driver assistance functions
- Electronic horizon and test drive recording solutions
- Driver assistance algorithms and functions
Agenda

- About EB Automotive
- Motivation
- Comparison of different architectures
- Concept for an 1oo2D software systems
- Summary
Different Types of Systems

According to Laprie (1992)
Current Automotive Systems are Fail-Safe

**Failure Detected?**

- Deactivate / degrade function
  → Safe State
- Inform the driver
- Report a diagnostic error

Standard approach in many safety relevant systems:
- Airbag, ESP, air conditioning, battery charging, ...
- Driver assistant functions such as adaptive cruise control, lane assist, ...

Some functions provide a degraded mode, sometimes limited in time:
- Electronic Power Steering
- Braking
Levels of Autonomous Driving (AD)

<table>
<thead>
<tr>
<th>driver in the loop</th>
<th>yes (required)</th>
<th>not required</th>
</tr>
</thead>
<tbody>
<tr>
<td>time to take control back</td>
<td>~ 1s</td>
<td>several seconds</td>
</tr>
<tr>
<td>other activities while driving</td>
<td>not allowed</td>
<td>specific</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>examples</th>
<th>FCW, LDW</th>
<th>ACC, LKA</th>
<th>Traffic Jam Assistant</th>
<th>Highway Chauffeur</th>
<th>Valet Parking</th>
<th>Robot car</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FCW ... Forward Collision Warning</td>
<td>ACC ... Adaptive Cruise Control</td>
<td>LKA ... Lane Keeping Assistant</td>
<td>Source: SAE, NHTSA, VDA</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Goal: Autonomous driving

Safe State could mean:

- Continue driving until driver is in the loop
  - approx. 7-15s for conditional autonomous driving
  - Several minutes for high and full autonomous driving
  - **Precondition**: driver monitoring including a black-box

- Perform an autonomous „safe-stop“ (stand-still at a non-hazardous place)
  - Main issue is to get the driver attention focused on the situation
  - Several minutes, depending on the situation
  - **Precondition**: legal acceptance, approved and certified black-box
Agenda

• About EB Automotive
• Motivation
• Comparison of different architectures
• Concept for an 1oo2D software systems
• Summary
2 channels with comparison

Redundant ECUs calculate using redundant data, output is compared.

A 2 channels with comparison system is fail-safe since you cannot distinguish between “ECU1 not ok” and “ECU2 not ok”.

The safe state is a complete system shutdown.
2oo3 Systems

Three redundant ECUs calculate on redundant data, output data is voted upon.

If one of the ECUs fails the system can continue with the remaining two ECUs.

Failures in the input data can be detected by an “Input-Voter”.
2003 Systems and the Automotive Domain

Applicable for automotive?

• More ECUs
• More wiring
• More weight
• More power consumption
• Higher complexity to manage

Will we as a customer accept that?

• Different opinions and market studies
• Referring to several studies, customer will pay 1500 - 3000€ more for autonomous driving car (mid-size car).

2003: An Example (Nissan Steer-by-Wire)

Source: http://www.caranddriver.com/features/electric-feel-nissan-digitizes-steering-but-the-wheel-remains-feature
Agenda

- About EB Automotive
- Motivation
- Comparison of different architectures
- Concept for an 1oo2D software systems
- Summary
1oo2D Systems

If an ECU fails in one of the two channels, the system does not shut down but continues to operate with only one channel.

The best policy is not to operate on a single channel for a “long” period of time.

Controlling this time and matching it with complete system behavior is the key.

**Precondition:** very high diagnostic coverage needed for each ECU to detect failures.
The Safe State: Design Patterns and Mechanisms for Fail-Operational Systems

From Detection to Prevention

Integrity mechanisms:
- Memory partitioning
- Data protection
- Temporal protection

Software Engineering:
- Plausibility checks
- Functional monitoring
- Defensive programming
- Semantic analysis
- Robustness

Car Infrastructure:
- Fault tolerant Ethernet
- Service orientated communication

Integrity mechanisms:
- Memory partitioning
- Data protection
- Temporal protection

Software Engineering:
- Plausibility checks
- Functional monitoring
- Defensive programming
- Semantic analysis
- Robustness

Car Infrastructure:
- Fault tolerant Ethernet
- Service orientated communication

Safety OS
- Microkernel
- QM Functions

Safety RTE
- BSW
- OEM modules
- QM CDD
- ASIL CDD
- MCAL
- MCAL (ASIL)
- Wdg

Safety OS
- Data Protection
- Stack Protection
- Context Protection
- OS Protection
- Hardware Error management

Safety E2E Protection
- Safe communication

Safety TimE Protection
- Alive supervision
- Deadline Monitoring
- Control flow monitoring
Dynamic 1oo2D

1oo2D
- Normal operation

1 channel
- Still Operational
- Handover to driver
- Failure recovery
- Internal recovery

< 10s

1oo2D*
- Rebuilding 2 channel system
- Disabling of comfort functions
1oo2D - Normal operation
The Safe State: Design Patterns and Mechanisms for Fail-Operational Systems

1002D – 1 channel

1oo2D system

- ECU1
  - Func1
  - Func2
  - Func3
  - Diagnostics
- ECU2
  - Func1
  - Func2
  - Func3
  - Diagnostics

Fault tolerant Ethernet

Sensors /Actuators

ECU3

- Func4
- Func5
- Func6
- Func3
- Func1

Critical

Non-critical
disabled
Requirements for Reconfiguration

- Req. 1: Functions can be dynamically relocated
- Req. 2: Sensor/Actuators are redundant or accessible via network
Dynamic Re-Configuration

Req. 1: Functions can be dynamically relocated
- Application information based on AUTOSAR xml description available
- Runtime environment (RTE) supporting reconfigurable software components
- Threads can restarted with e.g. Safety OS

Req. 2: Sensor/Actuators are redundant or accessible via network
- Service orientated communication
- Multi-cast fault-tolerant Ethernet
The Safe State: Design Patterns and Mechanisms for Fail-Operational Systems

Agenda

• About EB Automotive
• Motivation
• Comparison of different architectures
• Concept for an 1oo2D software systems
• Summary
Summary

• Re-use of available integrity mechanisms from fail-safe systems is the basis for building fail-operational systems.
• Software systems that are designed to achieve a high diagnostic coverage are available today and can be extended to fail-operational.
• Fault tolerant Automotive Ethernet is available today.
• Established concepts for fail-operational systems are available and can be reused in automotive systems with budget constraints.

Let’s build the next generation software systems for autonomous driving!
Contact us!

http://automotive.elektrobit.com
alexander.much@elektrobit.com
rudolf.grave@elektrobit.com