New Vehicle Architectures - New CANoe?

Testing the Future with CANoe - Vector Congress November 21th 2018
<table>
<thead>
<tr>
<th>Agenda</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Introduction</td>
</tr>
<tr>
<td>2.</td>
<td>AUTOSAR Adaptive Platform</td>
</tr>
<tr>
<td>3.</td>
<td>Impact on CANoe</td>
</tr>
<tr>
<td>4.</td>
<td>Outlook</td>
</tr>
</tbody>
</table>
Introduction

Current Trends and Requirements in Automotive Applications

- **ADAS**
  - Camera/LIDAR & Machine Learning

- **Connectivity**
  - Connection to non-AUTOSAR services
  - Security

- **Infotainment**
  - Integration of end-user devices
  - File handling

- **Dynamic Software**
  - Install and start applications during runtime
  - Enable 3rd party applications
Central Idea: “Smartphone on Wheels”

Key Features: Updateability and connection to backend infrastructure

Service-oriented architectures allow update and upgradeability during lifecycle of vehicle

Forward and backward compatibility

Ethernet and SOA enabling “End-to-End Architecture” from vehicle to the backend

New architectures will introduce high-performance nodes

Connection of high-performance nodes is realized with Ethernet as communication technology
In Detail: AUTOSAR Classic is Supplemented by AUTOSAR Adaptive

<table>
<thead>
<tr>
<th>Classic Automotive Requirements</th>
<th>New Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>▶ Hard real-time</td>
<td>▶ Support of high-performance µCs</td>
</tr>
<tr>
<td>▶ Safety</td>
<td>▶ Dynamic configuration</td>
</tr>
<tr>
<td>▶ Cost efficiency</td>
<td>▶ Secure and efficient link to the cloud</td>
</tr>
</tbody>
</table>

**AUTOSAR Classic Platform - CP**

- For deeply embedded control systems
  - Number of ECUs: ~50-120
- OSEK OS / Cooperative multitasking
- Developed in C
- Signal-oriented communication:
  - CAN, LIN, FR, (Ethernet)
  - Configured at compile time

**AUTOSAR Adaptive Platform - AP**

- For powerful computing nodes
  - Number of nodes: <10 / hypervisor
- POSIX OS / Preemptive multitasking
- Developed in C++
- Service-oriented communication:
  - Ethernet
  - Configured at runtime

**Foundation - FO**

Common requirements
The primary use cases of CANoe is to test ECUs and networks:
- During the development to verify individual development steps
- Test prototypes
- Perform regression and conformance tests
- CANoe services the System Under Test at all interfaces

Main focus of CANoe:
- Network specific elements ("CAN frame")
Impact on CANoe

Service Orientation

- New approach for built-in “service-oriented communication” instead of “network specific elements” was required
- Solution:
  - Communication Objects (CO) to model any type of communication
  - New CO layer fully integrated in the existing tool
  - Mixed operation with classic network specific elements possible
Impact on CANoe

Testing the Future

- Usage scenario: Testing algorithms on a platform for autonomous driving
  - Access to service-oriented software is directly possible from within CANoe
  - Analysis, simulation and automated test can be performed on Communication Objects
What’s still right about the CANoe approach

- Cars are not just another IT software
  - Testing of all production variants highly desired
  - Tests must be performed on various integration levels
    - Software component level
    - ECU level
    - Subsystem level
    - Entire vehicle network level
    - Test drive
  - CANoe can be used on all these levels

- Most important CANoe concepts
  - Simultaneous operation of all networks
  - Same time base for all networks and application layer objects
    - Allows testing of gateway applications
  - Scalability (distributed operation on multiple PCs)
Outlook

And Yet – Software Testing will become more important

- New product in planning: CANoe Server
  - Offering parallel and scaled computing
For more information about Vector and our products please visit

www.vector.com

Author:
Schwager, Mark
Vector Germany