CAN FD and Beyond - CANoe's New Simulation Concept
Agenda

- **Automotive Industry Trends**
  - SoA Middleware Requirements
  - The CANoe Function Bus
  - Test Concepts
  - Summary
Automotive Industry Trends

Mega Trends

**Connectivity (IoT)**
- Infotainment
- V2X
- Cloud services
- OTA update & diagnostics
- always on
- new business models

**Electrification**
- Efficiency
- Start-stop, sailing, ...
- Hybrid → Battery Electric Vehicle
- Multi voltage
- broad range of solutions
- standards needed

**Automated Driving**
- Different sensor types
- On/off-board information fusion
- Complex algorithms
- security is a must
- fail operational needed

Major Changes in E/E Architecture
# Automotive Industry Trends

## Embedded Software Becomes Software

### Technology Drivers

- ADAS
- Autonomous Driving
- Connected Car
- Car-to-X

### Implications

<table>
<thead>
<tr>
<th>Platform</th>
<th>AUTOSAR/OSEK</th>
<th>AUTOSAR Adaptive, Linux et al.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication</td>
<td>Signal flow</td>
<td>SoA, complex data</td>
</tr>
<tr>
<td>Topology</td>
<td>Static</td>
<td>Dynamic</td>
</tr>
<tr>
<td>Test focus</td>
<td>Device</td>
<td>Function</td>
</tr>
</tbody>
</table>

![Signal flow diagram](image)
Automotive Industry Trends

It’s All About Middleware

AUTOSAR Basic Approach
© AUTOSAR.org

AUTOSAR the Next Generation – The Adaptive Platform, Simon Fürst, ARS@EDCC 2015

http://portals.omg.org/dds/omg-dds-standard/
Agenda

Automotive Industry Trends

- **SoA Middleware Requirements**
  
  The CANoe Function Bus
  
  Test Concepts
  
  Summary
Asynchronous Remote Procedure Calls

Client

Middleware

ECU

Ethernet cable

Middleware

Server

GetItem(12)

Return(42)
**SoA Middleware Requirements**

**SOME/IP**

- **Services** are a composition of
  - **Methods** (RPCs)
  - **Events** (Notifications)
  - **Fields** (Events with getter/setter methods)
- **Announce** availability and location of services
- **Publish/subscribe** model for events

- Dynamic approach to communication
  - Service discovery
  - Activate/deactivate communication paths
  - Intelligent usage of uni-, multi-, broadcast
  - Service content is statically defined

- Save bandwidth and computing resources
- Easy relocation of services
SoA on CAN FD?

**SOME/IP** is for **Ethernet**, *not CAN FD*... right?

- SoA communication requirements
  - High data rate ✓
  - Bigger datagrams ✓
  - Dynamic payload ✓

Is CAN FD also a choice for SoA communication?
SoA Middleware Requirements

Prototyping and Integration

Real ECU
Virtual ECU
Simulation node
Prototyping and Integration

SoA Middleware Requirements

- Real ECU
- Virtual ECU
- Simulation node

CANoe
Prototyping and Integration

SoA Middleware Requirements

Real ECU

Virtual ECU

Simulation node
Prototyping and Integration

SoA Middleware Requirements

Real ECU

Virtual ECU

Simulation node

CANoe
The CANoe Function Bus

Converging point: Function Bus

Support services and their discovery

SOME/IP <SoA>

Address dynamic topologies

AUTOSAR Adaptive

Easy switching between real and simulated components

Abstraction client/server communication

AUTOSAR VFB

Virtual prototyping
Any pair of endpoints may
...connect distinct physical devices, or
...share a MAC address on the same PHY, or
...share an IP address, or
...connect abstract simulation code.
Communication Object: Details

**CameraServiceInterface**

```java
CameraServiceInterface {
  Operations:
  GetCamStatus()
  SetVideoFormat()
  Events:
  NewFrame
}
```

**System.arxml**

**Dyn. Clients**
- HeadUnit
- ParkAssist
- FrontLeftCAM
- FrontRightCAM
- RearCAM

**Dyn. Servers**
- HeadUnit
- ParkAssist

**Virtual Target**
- HeadUnit: Virtual Target
- ParkAssist: Real
- FrontLeftCAM: Simulated
- FrontRightCAM: Simulated
- RearCAM: Real
The CANoe Function Bus

Communication Object: Layers
The CANoe Function Bus

Communication Object: Layers

App. API
SOME/IP
TCP/UDP/IP
Ethernet

Binding block

App. API
SOME/IP
TCP/UDP/IP
Ethernet
Communication Object: Layers

The CANoe Function Bus

App. API

Signal

PDU

CAN FD Frame

App. API

Signal

PDU

CAN FD Frame

Binding block
Communication Object in Real Environment I

The CANoe Function Bus

Sim. node

App. API
SOME/IP
TCP/UDP/IP
Ethernet

Binding block configured

ECU
Communication Object in Real Environment II

The CANoe Function Bus

App. API
SOME/IP
TCP/UDP/IP
Ethernet

ECU

App. API
SOME/IP
TCP/UDP/IP
Ethernet

VN56xx

ECU

Measurement
The CANoe Function Bus

Communication Object in Simulated Environment

Sim. node A

Abstract transmission

Virtual ECU

App. API

SOME/IP

TCP/UDP/IP

Ethernet

App. API

SOME/IP

TCP/UDP/IP

Ethernet

No need to configure those layers

Abstract timing behaviour by transmission models
Agenda

Automotive Industry Trends
SoA Middleware Requirements
The CANoe Function Bus

Test Concepts

Summary
Test Concepts

Testing: Choosing the Right Layer

Classic domain: PDUs and signals

Service-oriented Architecture domain

<table>
<thead>
<tr>
<th>Test</th>
<th>Test</th>
<th>Test</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>send(msg1) on msg2 { ... }</td>
<td>$sig1 = 1; On sig2 {...}</td>
<td>@sig1 = 1; On sig2 {...}</td>
<td>SetStatus(42)</td>
</tr>
<tr>
<td>IL</td>
<td>NM</td>
<td>IL</td>
<td>SD</td>
</tr>
<tr>
<td>TP</td>
<td>TP</td>
<td>TP</td>
<td>SOME/IP</td>
</tr>
<tr>
<td>IL</td>
<td>NM</td>
<td>IL</td>
<td>SD</td>
</tr>
<tr>
<td>Application</td>
<td>Application</td>
<td>Application</td>
<td>Application</td>
</tr>
</tbody>
</table>

SUT (real or virtual)

SUT (real or virtual)

SUT (virtual only)

SUT (real or virtual)

SUT (virtual only)

CANoe Function Bus

CANoe

System variables

Eth

Re-use

Abstract

Test

Test

Test

Test

SetStatus(42)
Test Concepts

APIs for Testing SoA Applications

Service Discovery
- Abstraction of SOME/IP SD
- Configurable automatic offers
- Observable discovery states

Service Mockup
- Answering Machine
- Configurable automatic answers
- Dynamic endpoints

New wait functions
- Await service being discovered
- Await function call on server side
- Await function return on client side
- Await event subscription

New handlers
- On service discovered
- On function called
- On function returned
- On event subscribed
Test Concepts

Programming Examples

```c#
public override void Start()
{
    CAM.HeadUnit.FrontLeftCAM.GetCamStatus
        .CallAsync((result) => {
            Output.WriteLine("returned");
        });
}

[Testcase X]
TestWaitForCall(...);
...LatestCall.Result = 1;
...LatestCall.Return();
TestWaitForReply(...);
```

```capl
on key 'f' {
    CAM[HeadUnit, FrontLeftCam].GetCamStatus.[CallAsync()];
}

on called CAM[HeadUnit, FrontLeftCam].GetCamStatus {
    this.status = CAM_RECORDING;
    this.return = 1;
}

on returned
CAM[HeadUnit, FrontLeftCam].GetCamStatus {
    //...
}
```

New concept: Waiting handlers
Summary

Conclusion

- E/E architectures keep on changing
- All major OEMs design service-oriented architectures
- CANoe function bus key concepts:
  - Providing a simulation backbone for all relevant communication paradigms
  - Supporting virtual and real integration of SoA
  - Prototyping and integration of classic (signal-based) systems
  - Simplifying test cases by providing the right level of abstraction
- CAN FD fits the function bus paradigm
For more information about Vector and our products please visit

www.vector.com

Author:
Köllner, Christian
Vector Germany