Distributed Model Based Development For Car Electronics Scenarios and Examples
Outline

- Examples of Distributed Systems
- Scenarios by Example
Examples of Distributed Systems
Adaptive Cruise Control (ACC) + Brake By Wire

Source Renault
Brake By Wire ñ Sensotronic Brake Control System (SBC)

Block Diagram

- Electric Brakes
- Active Body Control
- Brake Pedal
- Windshield Wiper
- SBC Computer
- Engine/Transmission Control Unit
- ABS
- Steering Wheel Sensors
- Electronic Stability Program
- Front Wheel speed
- Engine Torque
- Gear Selection
- Engine Braking
- Brake Cylinder Pressure
- Transversal Acceleration
- Rotational Speed
- Delta Time
- Brake Pressure
Brake Project Ö From Centralized Redundancy

- Fault tolerant Pedal Module
- Fault tolerant Electronic Brake Module (fail-safe)
- Fault tolerant Power Management Module (fail-safe)
- Phase Converter for each actuator
- 4 High End Microcontrollers (incl. RAM, ROM, IO), 4 for the Brake Module
- 3 Low End Controller for the Pedal Module (incl. Sensor, Power Supply, RAM, ROM)
- 8-16 Midrange Microcontroller/DSP for the Phase Converters, Power Management (incl. Power stages, sensors, RAM, ROM)
- Totals:
  - 15-23 microcontrollers
  - 11 separate electronic control units
  - 9 communication links.
  - fault tolerant if 4 wheel braking is required after a
- Total of 4 ECUs and 3 communication links must be operational to allow braking on one wheel (fault-tolerant)
Brake Project ..to Distributed Redundancy

- Pedal Module
- Power Management Module (fail-silent ECUs)
- Wheel Brake ECU for both brake control and the phase converter functions

Totals:
- 11 microcontrollers (incl. resources)
- 9 separate electronic control units
- 8 communication links

3 ECUs and 2 communication links must be operational to allow braking at one wheel

Source: Infineon-Delphi-Volvo-WindRiver
Brake Project: (variant of) Unidirectional Redundant Ring Structure

- 2 travel sensors and one force sensor to determine driver intent (each sensor connected to a different wheel node)
- Sensor values communicated over the network → Consistency checks
- Wheel node calculates the actuation commands for all four wheels
- Commands communicated via network → each of the four wheel nodes compares their own actuation commands with those calculated by the other wheel nodes
- Voting mechanism in the network layer of each wheel node can then disable the power to individual actuators in case of a fault.
- If a node needs to be shut down the brake force is redistributed to prevent the vehicle from yawing.
- The advanced brake functions (ABS) are executed in the two front wheel nodes.
- If the front wheel nodes do not calculate the same output commands for these advanced brake functions, the function will be deactivated. This provides fail-safe operation
- Redundant power supply.

Source: Infineon-Delphi-Volvo-WindRiver
Scenarios by Example
Objectives

1. To implement the SBC system on the Infineon-Delphi distributed redundancy architecture integrating the basic brake function with ABS, etc.Ø
   - Assumption1: designer has only the SBC spec ñ executable spec not yet available for the control algorithms (typical INTEGRATOR CASE!)
   - Assumption2: distributed architecture is defined (number of ECUs, Communication Protocols)

2. To add the ACC functionality (derivative design) to the existing SBC system (virtual) platform
Scenarios/Design Steps for Objective 1

1.0: UML Description for SBC, ABS, etc
   - Product Marketing Manager

1.1: Functional Network Integration with coarse models/un-Timed Validation (SBC + ABS + ..)
   - System Architect

1.2: Control Algorithms/Plants Development and Un-Timed Validation
   - Algorithm Developer

1.3: Functional Network with finer models/un-Timed Validation (SBC + ABS + ..)
   - System Integrator

1.4: Functional Network with finer models/Timed Validation (SBC + ABS + ..)
   - System Integrator

Sw (re)-Distribution

SBC Virtual Prototype Sign-off

Protocol (re)-Configuration
Scenarios/Design Steps for Objective 1

SBC Virtual Prototype Sign-off

1.5: System Aware ECU SW RT Validation

1.6: Protocol Configuration Detailed Analysis

1.7: Export (SW to Target, Comm Layer, etc.)

Sw Real Time Validation

Cycle Accurate Validation

SW Developer

System Integrator

System Integrator

4: Functional Network

SBC Physical Prototype Ready for Testing

System Integrator
Scenarios/Design Steps for Objective 2

1. **UML Description for ACC**
   - Product Marketing Manager

2. **Functional Network Integration w/ coarse ACC model/ un-Timed Validation (SBC Virtual Platform + ACC)**
   - System Architect

3. **ACC Control Algorithms/Plants Development and Un-Timed Validation**
   - Algorithm Developer

4. **Functional Network w/ Finer ACC model/ un-Timed Validation (SBC Virtual Platform + ACC)**
   - System Integrator

5. **Functional Network w/ Finer ACC model/ Timed Validation (SBC Virtual Platform + ACC)**
   - System Integrator

6. **Sw (re)-Distribution**

7. **SBC + ACC Virtual Prototype Sign-off**
   - Protocol (re)- Configuration

---

**Cadence Confidential**

**Automotive Tiger Team**
Scenarios/Design Steps for Objective 2

SBC + ACC Virtual Prototype Sign-off

2.5: System Aware ACC ECU SW RT Validation

Sw Real Time Validation

2.6: Protocol Configuration Detailed Analysis

Cycle Accurate Validation

2.7: Export (SW to Target, Comm Layer, etc.)

4: Functional Network

SBC + ACC Physical Prototype Ready for Testing

System Integrator

System Integrator

SW Developer

System Integrator
Scenarios/Design Steps for Objective 1

1.0: UML Description for SBC, ABS, etc
   - Product Marketing Manager

1.1: Functional Network Integration w/ coarse models/un-Timed Validation (SBC + ABS + ..)
   - System Architect

1.2: Control Algorithms/Plants Development and Un-Timed Validation
   - Algorithm Developer

1.3: Functional Network w/ finer models/un-Timed Validation (SBC + ABS + ..)
   - System Integrator

1.4: Functional Network w/ finer models/Time Validation (SBC + ABS + ..)
   - System Integrator
   - Protocol (re)-Configuration

Sw (re)-Distribution

SBC Virtual Prototype Sign-off
1.1: Functional Network Integration with Coarse Grain Models

- Windshield Wiper
- Brake Pedal
- Engine/Transmission Control Algorithm
- Steering Wheel Sensors
- Electronic Stability Program

SBC Function

- Brake1 Pressure
- Brake1 Cylinder Pressure
- Wheel1 speed
- ABS
- Wheel1 Sensors
- Brake2
- ABS
- Wheel2 Sensors
- Brake3
- Brake4

Delta Time
Brake Pedal Speed
Engine Torque
Engine Braking
Gear Selection
Steering Angle
Rotational Speed
Transversal Acceleration
1.1: Functional Network Integration with Coarse Grain Models
1.1: Functional Network Validation with token-level (messages) un-timed Simulation

- Windshield Wiper
- Brake Pedal
- Engine/Transmission Control Algorithm
- Steering Wheel Sensors
- Electronic Stability Program

Simulation w/ Probing
1.1: Functional Network Validation with token-level un-timed Simulation
1.1: Functional Network Validation with token-level un-timed Simulation

Display Simulation Results...
Scenarios/Design Steps for Objective 1

1.0: UML Description for SBC, ABS, etc

1.1: Functional Network Integration w/ coarse models/ un-Timed Validation (SBC + ABS + ..)

1.2: Control Algorithms/Plants Development and Un-Timed Validation

1.3: Functional Network w/ finer models/un-Timed Validation (SBC + ABS + ..)

1.4: Functional Network w/ finer models/ Timed Validation (SBC + ABS + ..)

Sw (re)-Distribution

SBC Virtual Prototype Sign-off

Product Marketing Manager
System Architect
Algorithm Developer
System Integrator
System Integrator
Protocol (re)-Configuration
1.2: Control Algorithms/Plants Development and un-timed validation

Export of Plant Model Interface to Mathworks/Simulink (not available in the current offer)

Plant Development and Un-Timed Validation

Control Algorithm Development and Un-Timed Validation
Scenarios/Design Steps for Objective 1

1.0: UML Description for SBC, ABS, etc

1.1: Functional Network Integration w/ coarse models/ un-Timed Validation (SBC + ABS + ..)

1.2: Control Algorithms/Plants Development and Un-Timed Validation

1.3: Functional Network w/ finer models/un-Timed Validation (SBC + ABS + ..)

1.4: Functional Network w/ finer models/ Timed Validation (SBC + ABS + ..)

Sw (re)-Distribution

SBC Virtual Prototype Sign-off

Product Marketing Manager

System Architect

Algorithm Developer

System Integrator

System Integrator

Protocol (re)-Configuration

Cadence Confidential

Automotive Tiger Team
1.3: Functional Network Integration with Finer Grain Models

- Import of Plant Model from Mathworks/Simulink
- Import of Control Algorithm Model from Ascet-SD (From Target Link in the future)
1.3: Functional Network Integration with Finer Grain Models

Refinement from C++ Models to Hierarchical Finer Grain Detailed Models
1.3: Functional Network Validation with token-level un-timed Simulation

Simulation with Finer Grain Models Finished....
1.3: Functional Network Validation with token-level un-timed Simulation
Scenarios/Design Steps for Objective 1

1.0: UML Description for SBC, ABS, etc

1.1: Functional Network Integration w/ coarse models/ un-Timed Validation (SBC + ABS + ..)

1.2: Control Algorithms/Plants Development and Un-Timed Validation

1.3: Functional Network w/ finer models/un-Timed Validation (SBC + ABS + ..)

1.4: Functional Network w/ finer models/ Timed Validation (SBC + ABS + ..)

Sw (re)-Distribution

SBC Virtual Prototype Sign-off

Product Marketing Manager
System Architect
Algorithm Developer
System Integrator
System Integrator
Protocol (re)- Configuration
1.4: Functional Network Validation with token-level timed Simulation
1.4: Functional Network Validation with token-level timed Simulation
1.4: Functional Network Validation with token-level timed Simulation
1.4: Functional Network Validation with token-level timed Simulation
1.4: Functional Network Validation with token-level timed Simulation

- UCM Bus Controllers Configured with Periodic Frame Activation Policy
- UCM Bus Controllers Configured with Dynamic Asynchronous Frame Activation Policy
- UCM Communication Cycle is Configured as Time-Triggered Communication (no arbitration)
- UCM Communication Cycle as High Speed CAN (arbitration)
1.4: Functional Network Validation with token-level timed Simulation

Mapping of SW and Zero-Time Communications: Automatic Configuration of Buses and SW Tasks
1.4: Functional Network Validation with token-level timed Simulation

Coarse Grain Analysis via Simulation...are the signals still OK?
1.4: Functional Network Validation with token-level timed Simulation


Transition from idle to Queued
Transition from Accessing to Idle
Transition from Blocked to Accessing

Finer Grain Analysis via Simulation
1.4: Functional Network Validation with token-level timed Simulation

SW Task Scheduling Validation: Task Priorities? Cooperative? Preemptive?

Finer Grain Analysis via Simulation
Scenarios/Design Steps for Objective 1

1.0: UML Description for SBC, ABS, etc

1.1: Functional Network Integration w/ coarse models/ un-Timed Validation (SBC + ABS + ..)

1.2: Control Algorithms/Plants Development and Un-Timed Validation

1.3: Functional Network w/ finer models/un-Timed Validation (SBC + ABS + ..)

1.4: Functional Network w/ finer models/ Timed Validation (SBC + ABS + ..)

Sw (re)-Distribution

SBC Virtual Prototype Sign-off

Product Marketing Manager

System Architect

Algorithm Developer

System Integrator

System Integrator

Protocol (re)- Configuration

Cadence Confidential

Automotive Tiger Team
Scenarios/Design Steps for Objective 1

SBC Virtual Prototype Sign-off

1.5: System Aware ECU SW RT Validation

Sw Real Time Validation

1.6: Protocol Configuration Detailed Analysis

Cycle Accurate Validation

1.7: Export (SW to Target, Comm Layer, etc.)

4: Functional Network

SBC Physical Prototype Ready for Testing

SW Developer

System Integrator

System Integrator

System Integrator
1.5: System Aware ECU SW RT Validation

ECU1 SW Real Time Validation: Simulation with Probing of ECU1 Traffic

Functional Network

ECU1 TT Bus Probing

ECU1 CAN Bus Controller Collected Input Trace

Architectural Model

ECU1 TT Bus Controller Collected Input Trace
1.5: System Aware ECU SW RT Validation

ECU1 SW Real Time Validation: Export of ECU1 SW Task Configuration/ RT Constraints
1.5: System Aware ECU SW RT Validation

- LF Base Brake Model
- LF ABS Model
- Event Driven
- Periodic 50 ms
- RT Constraints
- dSpace TL
- ETAS ASCET-SD
- SW Debugger/ Cycle Accurate Timing Estimator
- Application SW Validation vs. RT constraints
- ECU1 TT Bus Controller/ CAN Bus Controller Traffic Traces
Scenarios/Design Steps for Objective 1

SBC Virtual Prototype Sign-off

1.5: System Aware ECU SW RT Validation

Sw Real Time Validation

1.6: Protocol Configuration Detailed Analysis

Cycle Accurate Validation

1.7: Export (SW to Target, Comm Layer, etc.)

4: Functional Network

SBC Physical Prototype Ready for Testing

SW Developer

System Integrator

System Integrator

System Integrator

Automotive Tiger Team

Cadence Confidential
1.6: Protocol Configuration Detailed Analysis

Export of Bus Configuration
1.6: Protocol Configuration Detailed Analysis

- Co-Simulation with CANoE
- Export CAN Bus Configuration
- XML DB
Scenarios/Design Steps for Objective 1

1.5: System Aware ECU SW RT Validation

1.6: Protocol Configuration Detailed Analysis

1.7: Export (SW to Target, Comm Layer, etc.)

4: Functional Network

SBC Virtual Prototype Sign-off

Sw Real Time Validation

Cycle Accurate Validation

SBC Physical Prototype Ready for Testing

SW Developer

System Integrator

System Integrator

System Integrator
Scenarios/Design Steps for Objective 2

2.0: UML Description for ACC

2.1: Functional Network Integration w/ coarse ACC model/ un-Timed Validation (SBC Virtual Platform + ACC)

2.2: ACC Control Algorithms/Plants Development and Un-Timed Validation

2.3: Functional Network w/ Finer ACC model/ un-Timed Validation (SBC Virtual Platform + ACC)

2.4: Functional Network w/ Finer ACC model/ Timed Validation (SBC Virtual Platform + ACC)

Sw (re)-Distribution

SBC + ACC Virtual Prototype Sign-off

- Product Marketing Manager
- System Architect
- Algorithm Developer
- System Integrator
- System Integrator
- Protocol (re)- Configuration

Cadence Confidential

Automotive Tiger Team
2.1: Functional Network Validation with token-level timed Simulation

Derivative Design: Adding ACC...do they work together?
Scenarios/Design Steps for Objective 2

2.0: UML Description for ACC

2.1: Functional Network Integration w/ coarse ACC model/ un-Timed Validation (SBC Virtual Platform + ACC)

2.2: ACC Control Algorithms/Plants Development and Un-Timed Validation

2.3: Functional Network w/ Finer ACC model/ un-Timed Validation (SBC Virtual Platform + ACC)

2.4: Functional Network w/ Finer ACC model/ Timed Validation (SBC Virtual Platform + ACC)

SBC + ACC Virtual Prototype Sign-off

Sw (re)-Distribution

Product Marketing Manager

System Architect

Algorithm Developer

System Integrator

Protocol (re)-Configuration

Skipped these two Steps in slides yet they do exist!!
2.4: Functional Network Validation with token-level timed Simulation

Coarse Grain Analysis via Simulation...Existing Architecture is sufficient?
Scenarios/Design Steps for Objective 2

SBC + ACC Virtual Prototype Sign-off

2.5: System Aware ACC ECU SW RT Validation

2.6: Protocol Configuration Detailed Analysis

2.7: Export (SW to Target, Comm Layer, etc.)

SBC + ACC Physical Prototype Ready for Testing

Sw Real Time Validation

Cycle Accurate Validation

These three Steps are left to the reader as exercise 😊

4: Functional Network

SW Developer

System Integrator

System Integrator

System Integrator

Cadence Confidential

Automotive Tiger Team
Backup Slides
Brake Project â€” High Level Objectives

- Distributing vehicle control over multiple subsystems from different vendors
- Enabling vehicle control by defining open interfaces between chassis control systems
- Providing a fault-tolerant safety architecture based on system distribution
- Enhancing the OSEK operating system to create a distributed operating system
- Distributed Approach is focused on the base brake function: most critical for safe and reliable brake system
- Shift from Fail-Safe to Fail-Silent:
  - Fail-Safe: In case of an error within the electronic control, the performance of the mechanical system is reduced, while still providing minimum functionality
  - Fail-Silent: When one task or node fails, another is already running and the system employs the alternative
Brake Project ñ Main Features

- Dynamic allocation of control tasks to multiple electronic control units
- Time-triggered scheduling of the tasks enabling task encapsulation
- "Silent State" Operating system:
  - time-triggered concept
  - deterministic system behavior
  - complete separation of tasks (task encapsulation): in case of faulty task, the operating system enters a "silent state" (fail silent)
- "Fault Tolerant" Communication System:
  - no central master and ECU local redundancy → Reduced Number of ECUs
  - Local redundancy achieved by inherent redundancy of a braking system
- Definition of Open Interface Standards:
  - between the brake actuatorís control circuitry and the brakeís electronic control unit
  - to the communication system
  - between upper control layers
Brake Project 🟤 Challenges

- Global time synchronization between component ECUs and processes to ensure accurate calculation of vehicle states from sensor inputs
- Deterministic actuation of wheel brakes with the respect to brake demands
- Composability of brake system elements
- Encapsulation of sub-systems and faults
- Acceptable vehicle performance in the presence of fault through redundancy, including:
  - Dynamic reconfiguration and redistribution of brake function
  - Optimization to minimize number of redundant subsystems
  - Extended and optimized degraded modes to achieve full potential of vehicle performance
Brake Project ñ Communication Medium

- Connects and correlates the distributed control units in the control path
- Encapsulates the distributed control units to prevent fault propagation (time and value domain)

Requirements:
- Application level features
- Synchronized and distributed global time synchronisation
- Failure tolerance
- Encapsulation at the protocol and at the physical level
- Compatibility with other systems already in production

Need open standard for deterministic failure-tolerant communication protocol (FlexRay)