2nd Conference
**Active Safety through Driver Assistance**
April 4-5, 2006
Garching near Munich, Germany

A. Sulzmann / DaimlerChrysler Truck, DaimlerChrysler AG, Stuttgart
G. Spiegelberg / Siemens VDO Automotive AG, Regensburg

**Motivation**

- Scalability
- Operational Reliability
- Simple Aggregate Interfaces
- Economic efficiency

[Diagram showing various applications and aggregates connected through sensors and interfaces.]
Table of contents

1. SPARC
2. Platform approach
3. Reconfiguration of redundant systems
4. Generic behavior model of aggregates
5. Conclusion and prospects

X-by-Wire Platform developed in SPARC

- **Driver**
  - Objective: Wish Control Vector

- **Data Fusion & Auto Pilot**
  - Objective: Redundant Auto Control Vector

**Decision Level**
- Objective: Safe Control Vector

**Execution Level**
- Objective: Control Signals to Aggregates

SPARC: EU program coordinated by DCAG
Table of contents

1. SPARC
2. Platform approach
3. Reconfiguration of redundant systems
4. Generic behavior model of aggregates
5. Conclusion and prospects

Requirements of the Platform

Key Requirements of the Platform
- No Single Point failure in the Platform independent of its probability
- Scalability to failures in terms of F/S, F/O-F/S, F/O-F/O...-F/S
- Capability of Dissimilarity / Diversity
- Minimum Wiring between Central Unit and Aggregates
  - Information via Bus
  - Power via Energy Bus
- Simple Interfaces in Aggregates

Detailed Key Requirements of the Platform
- System is split into two Sides
- Each Side has its own independent Bus (FlexRay)
- Only the Processing Units of the Central Computer have access to both FlexRay Busses
- Simplex Aggregates are linked to one FlexRay Bus only
Structure of the Platform

Two FlexRay Buses
Two Platform Computers
Simplex Aggregates are linked to one bus only
Redundant Aggregates are linked to both busses (MMI)

Drive-by wire system architecture

Each Platform Computer consists of two XCC
Each XCC is built up as a dual Unit
Each XCC runs in strict Fail/Passive Mode
XCC/R run synchronously to Bus R
XCC/B run synchronously to Bus B
Each Aggregate receives Data from Computer B and Computer R
Which XCC runs specific application is transparent to external aggregates
Each XCC can receive Data from all Aggregates resp. Sensors, MMIs

EBS... Elect. Brake
StW... Steer by Wire
ESP... Elect. Stab. Program (Sensor)
XCC... Universal Control Computer
Redundancy-Management
Basic SW-Modules

- Application 1
- Application 2
- ...

Functional Control-SW

- Application-Management
  Sensor/actuator resource-management, application [partition] -control ...

- Platform-Management
  Generic Management of the redundant control -platform ...

- IO-Management
  Interactive data-consistency

OS
  Scheduler, Device-Driver, Partition-Management, ...

Table of contents

1. SPARC
2. Platform approach
3. Reconfiguration of redundant systems
4. Generic behavior model of aggregates
5. Conclusion and prospects
Reconfiguration of redundant systems

- Application A1 is more important than Application A2
- Both Applications are available on each XCC (Universal Control Computer), but only one is running on one XCC
- Restart with same functional configuration will cause collisions on FlexRay
- Power Interrupt or Restart after internal error
- Communication mechanism required to detect wrong configuration and to force right reconfiguration

![Diagram of reconfiguration process]

Reconfiguration – collision detection

- Data_XCC:
  - Hardware dependant timeslot
  - Fixed to exact one node in cluster
- Data_APPLI:
  - Function dependant timeslot
  - Fixed to function running on XCC
  - Hardware dependant timeslot required to guarantee collision-free start-up even in case of wrong configuration
  - In case of running the same functionality on both XCCs, collision is detected by wrap-up of duplex HW.

Nullframe
Filled Frame
Generic behavior model of aggregates

General Requirements

- Minimum wiring
- Flexible
- Simple interface to redundant platform
- Degree of redundancy platform shall not be visible to aggregates
- Valid for all kinds of aggregates (sensors, actuators, etc.)
- Scalable in terms of amount of actuators and sensors
- Platform shall be able to manage the behavior of aggregates
Considered aggregate states

Communication: aggregates → platform

Aggregate states:
- Current state of aggregate (e.g. STBY, NOB, BIT, etc.)

Message counter:
- Increased by aggregate
- Indicates failure (e.g. frozen host-CPU of aggregate)

BIT (Built-in-Test):
- BIT-Status
- BIT-Type
- BIT-Result

FlexRay Configuration Version:
- Current version number of the FlexRay configuration file (*.chi)

Failure Code:
- Internal error of aggregate (e.g. Communication on one FlexRay channel failed)
Communication: platform ↔ aggregates

Platform state:
- Current state of XCC (Master or Slave)

Message counter:
- Increased by XCC
- Used for Wrap-up (XCC-internal check-mechanism)

Mode command:
- XCC commands mode to the aggregate, which the aggregate should stay in or should take over (e.g. STBY, BIT, NOP, etc.)

General message-format

Institut für Luftfahrtsysteme
Scalability and Standardisation

The scalability requires standardisation with respect to:
- Behaviour-Model of
  - Sensors
  - Actuators
- Rating of resources
  - Sensors
  - Actuators

Platform-Scalability

Within SPARC the platform-approach is scalable with respect to:
- Number of sensors/actuators
- Amount of I/O-data

Scalability means that changes of the scalability-domains will not lead to a change-request of the concept.

Transferability of platform between vehicles
Conclusion

- X-by-wire Platform approach
  - Scalability
  - Economic efficiency (Minimum wiring, Efficient Use of Resources)

- Reconfiguration
  - Intelligent reconfiguration mechanism by using available HW in an intelligent way

- Generic behaviour model for aggregates
  - Simple Interfaces in Aggregates
  - Control functionality
  - Error detection

Table of contents

1. SPARC
2. Platform approach
3. Reconfiguration of redundant systems
4. Generic behavior model of aggregates
5. Conclusion and Prospects
Prospects

- Application of XCP-Protocol for calibration and diagnosis purposes
- Flashing of SW-updates via FlexRay-Network

Thank you for your attention!

Universität Stuttgart
- Institut für Luftfahrtsysteme -
Eduard Zimmer
Eduard.Zimmer@ils.uni-stuttgart.de