Embedded System Development for Automotive Applications: Trends and Challenges

Werner Damm
OFFIS

OFFIS

SafeTRANS

ARTIST

Acknowledgements

- This presentation reports on
  - Results of the NoE Artist, notably the Results of the Workshop “Beyond Autosar” (co-organized with Albert Benveniste, INRIA)
  - Roadmapping activities within the SafeTRANS competence cluster (see www.safetrans-de.org)
  - Workshops and discussions within the Technology Platform Artemis (see www.artemis-office.org)
  - Research carried out in preparation to the Integrated Project Speeds
    - INRIA, OFFIS, Parades, Verimag
    - Airbus, Bosch, DaimlerChrysler, Israeli Aircraft Industries, Magna Powertrain, Knorr Bremse, Saab
    - Esterel Technologies, Extesy, Telelogic, TNI
- All rights rest with the contributors – see individual acknowledgements in sections of presentation

Structure of Presentation

- The Automotive Market
- The Autosar Approach
- Impact and Challenges on Real-Time Analyis
- Impact and Challenges on Control
- Impact and Challenges on Safety Analysis
- The Speeds Answer

The Automotive Sector

Facts and Trends

OFFIS

SafeTRANS

ARTIST

Automotive Sector: Facts and Trends[1]

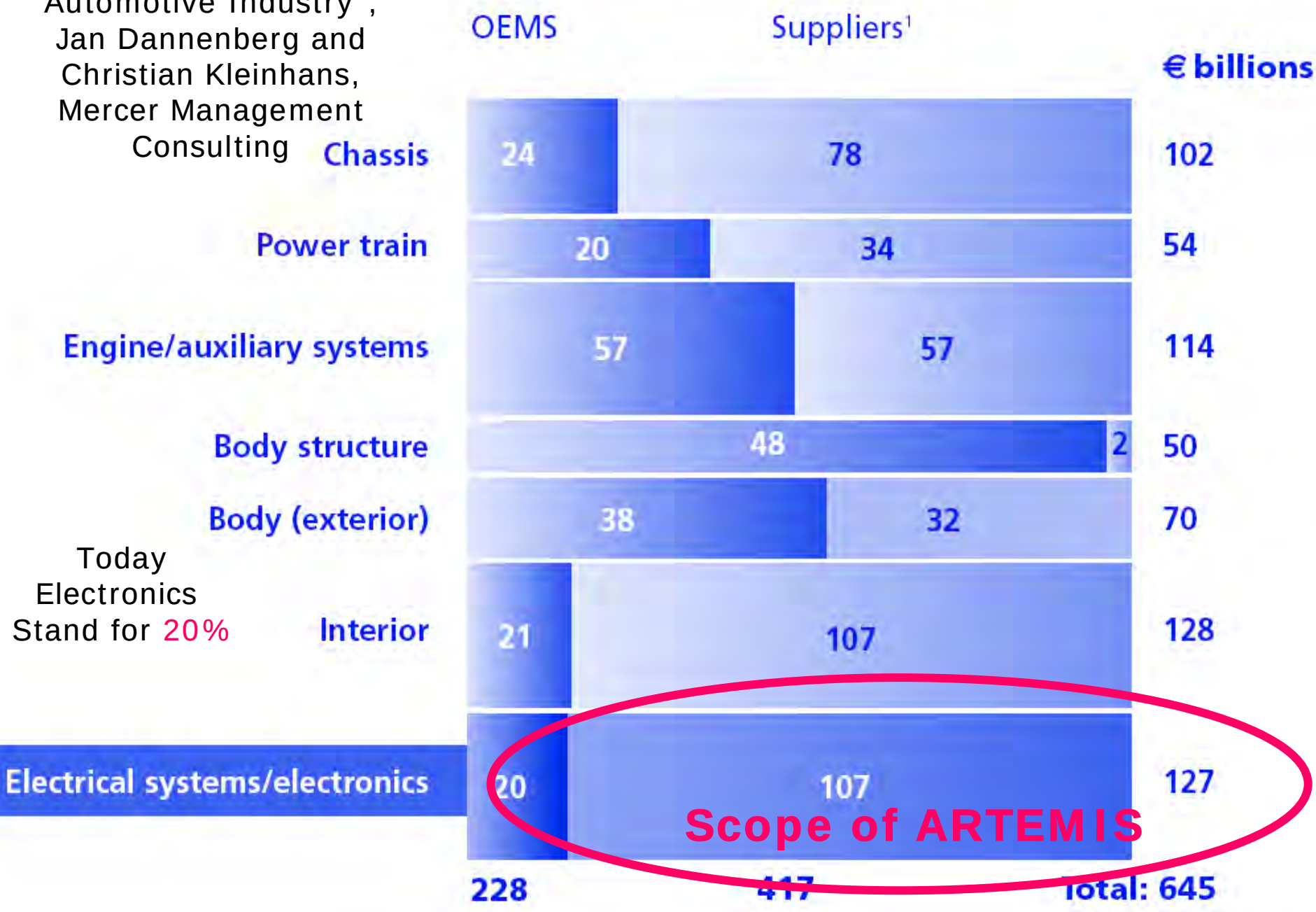
➢Share of World-Wide GDP	15%
➢Total GDP	645 Billion €
- Europe	204 Billion €
➢Direct jobs OEMs & Supplier	8.8 Million
- Europe	2,71 Million
➢Number of Units light vehicles/year	57 Million
➢Will be pushed to	76 Million

by 2015 through 2 Trillion € investment

[1]Quoted from "The Coming Age of Collaboration in the Automotive Industry", Jan Dannenberg and Christian Kleinhans, Mercer Management Consulting

©"The Coming Age of Collaboration in the Automotive Industry", Jan Dannenberg and Christian Kleinhans, Mercer Management Consulting

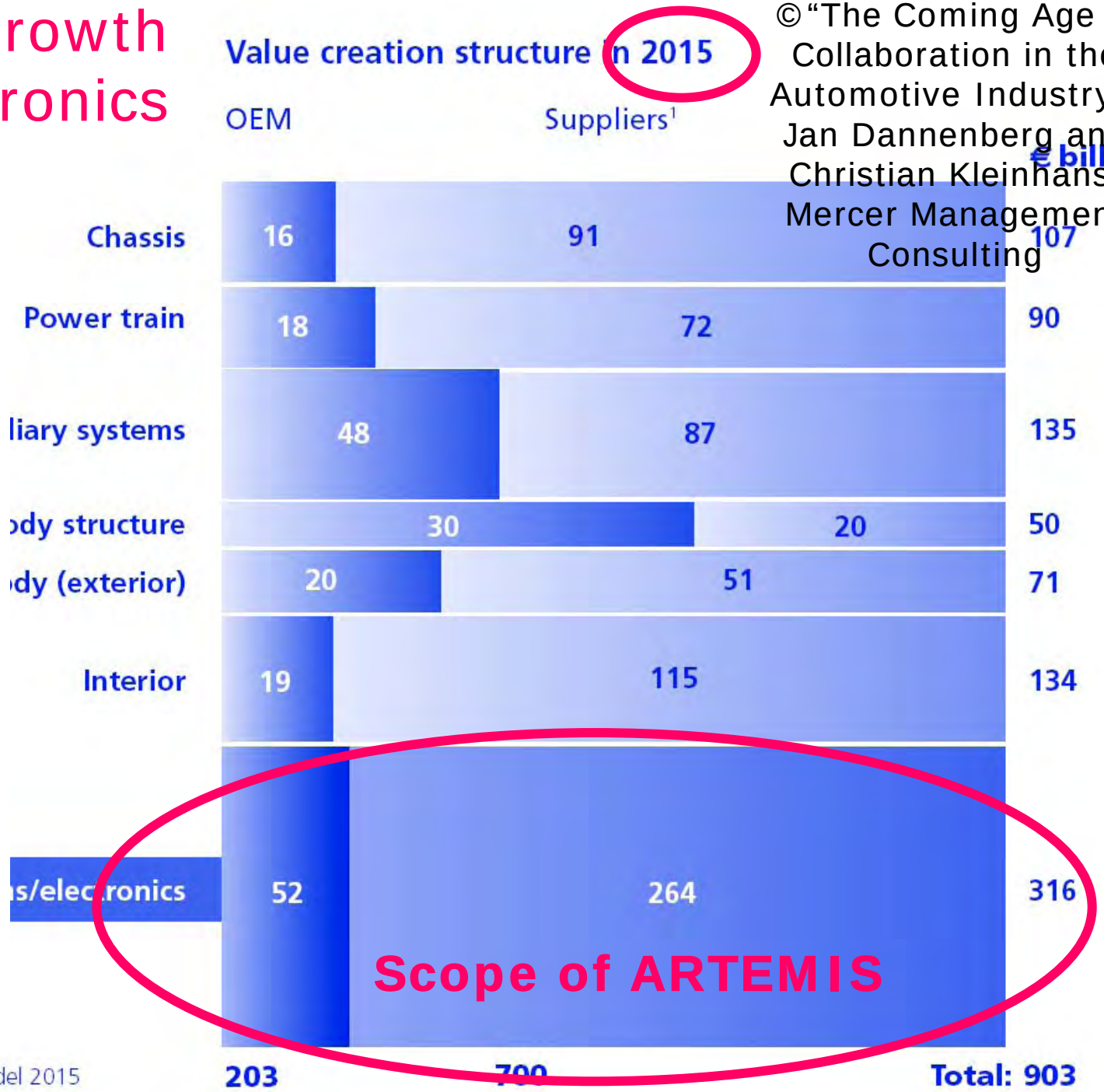
Value creation structure in 2002



Today Electronics Stand for 20%

Strongest Growth in Electronics

- Overall growth rate 40%
- **Electronic growth rate 150%**
- Increases **average** share of electronics to 35% from current 20%
- **More than 600 000 new jobs only in Automotive Electronics in Europe**



[1] Includes service providers
Source: Mercer Value Creation Model 2015

©"The Coming Age o Collaboration in the Automotive Industry" Jan Dannenberg and Christian Kleinhans, Mercer Management Consulting

Automotive SW
Workshop
San Diego, USA
H.-G. Frischkorn
BMW Group
10 - 12. Jan. 2004
Page 3

Automotive Software
An Emerging Domain

Up to 90% of future vehicle innovations will be based on electronic embedded systems.

Electronic Injections
Check Control
Speed Control
Central Locking
...

Electronic Gear Control
Electronic Air Condition
ASC Anti Slip Control
ABS Anti Blocking Sys.
Telephone
Seat Heating Control
Autom. Mirror Dimming
...

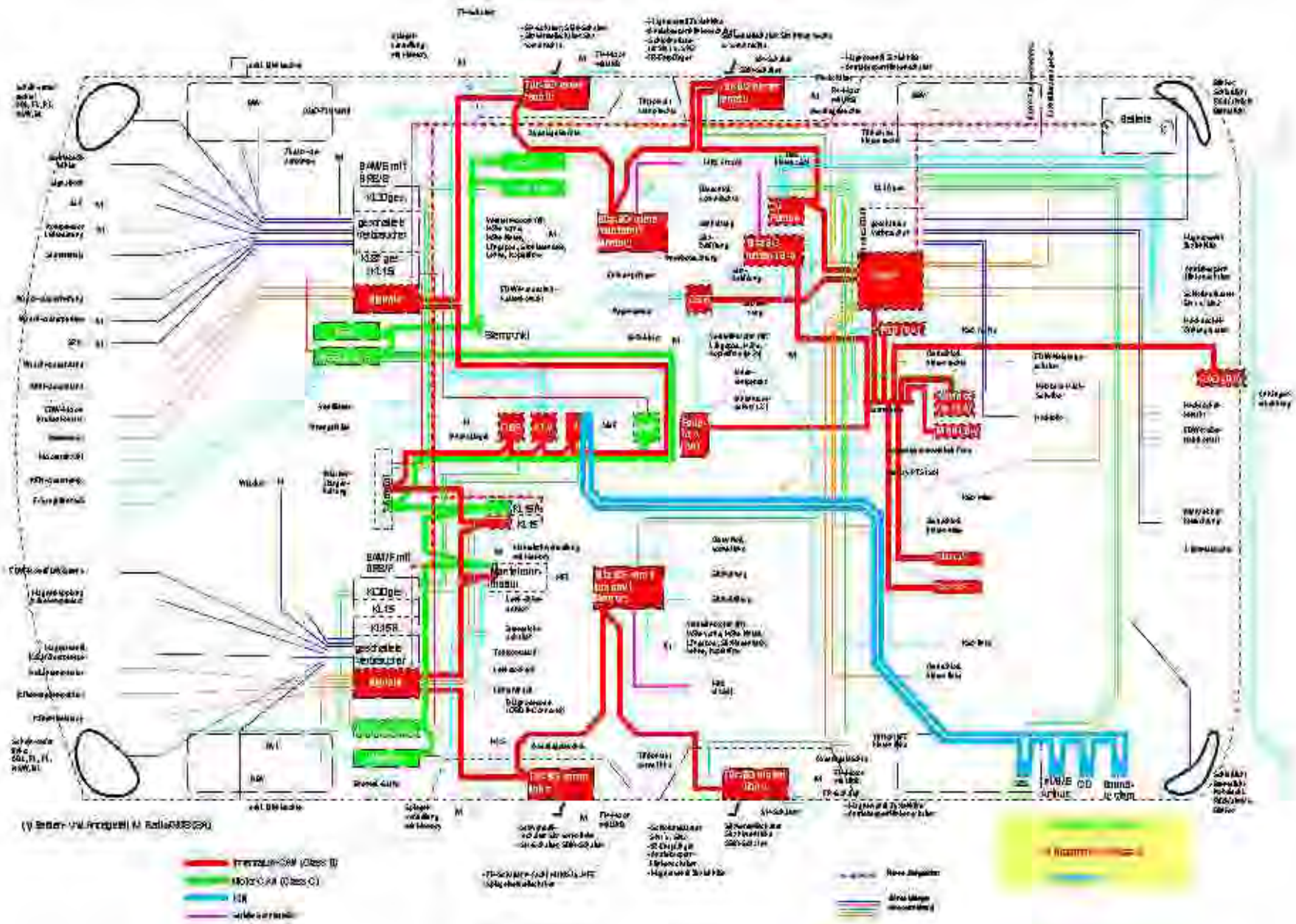
Navigation System
CD-Changer
ACC Active Cruise Control
Airbags
DSC Dynamic Stability Control
Adaptive Gear Control
Xenon Light
BMW Assist
RDS/TMC
Speech Recognition
Emergency Call
...

ACC Stop&Go
BFD
ALC
KSG
42-Voltage
Internet Portal
GPRS, UMTS
Telematics
Online Services
Blue-Tooth
Car Office
Local Hazard Warning
Integrated Safety System
Steer/Brake-By-Wire
I-Drive
Lane Keeping Assist.
Personalization
Software Update
Force Feedback Pedal

1970 1980 1990 2000

Degree of Interdependence

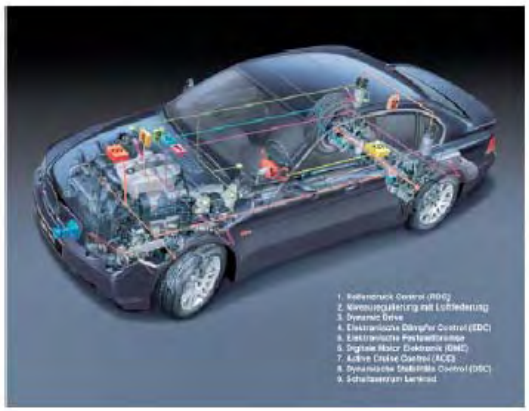
Electronic components are pervasive in today‘s vehicles



Automotive Software: An Emerging Domain

A Software Perspective

- Up to 40% of the vehicles' costs are determined by electronics and software
- 90% of all innovations are driven by electronics and software
- 50 – 70% of the development costs for an ECU are related to software
- Premium cars have up to 70 ECUs, connected by 5 system busses

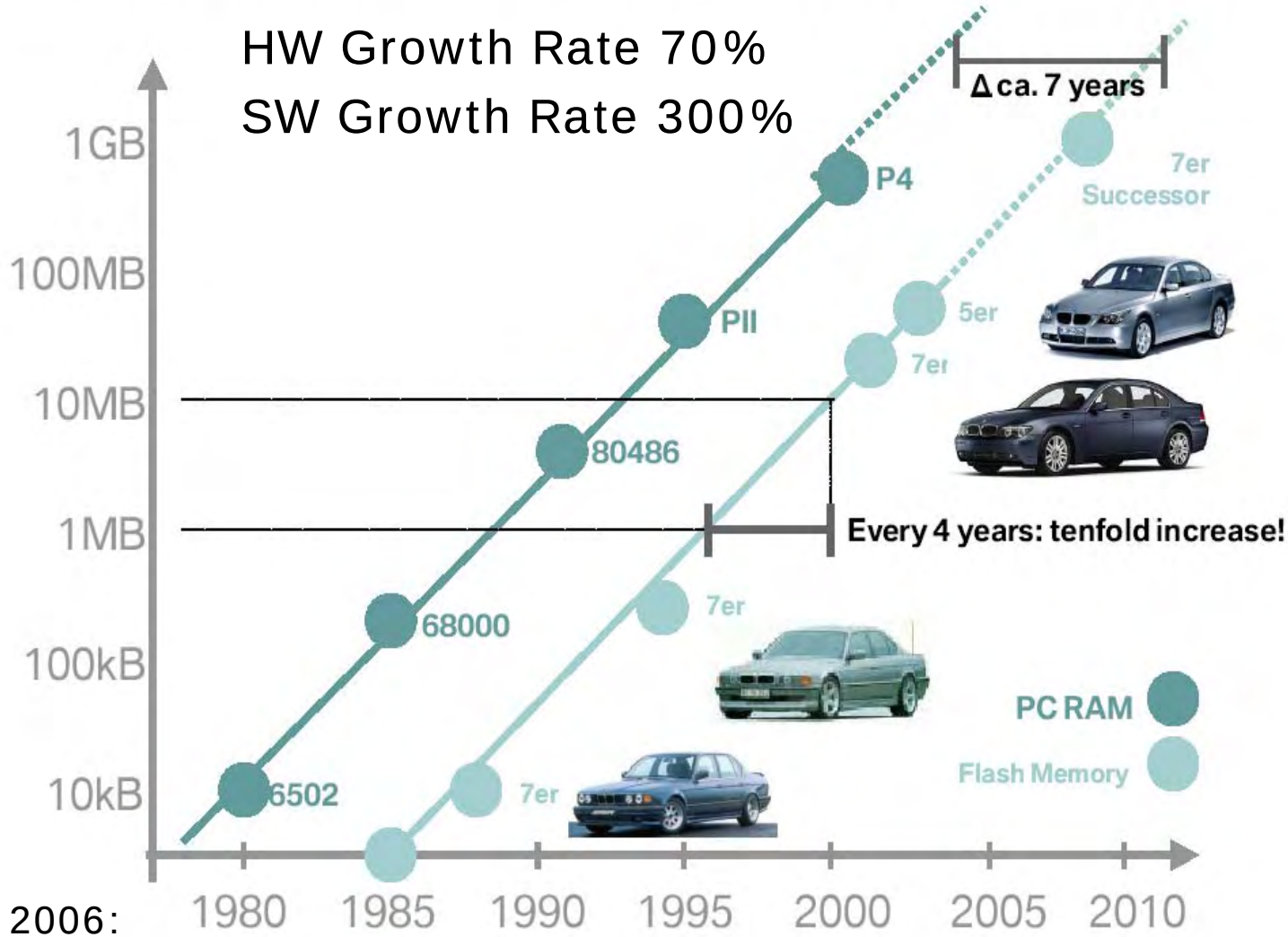


- **Growing system complexity**
- **More dependencies**
- **Costs play significant role**

Automotive SW
Workshop
San Diego, USA
H.-G. Frischkorn
BMW Group
10 -12. Jan. 2004
Page 8

Automotive Software Systems Today

Moore's Law: Memory Use in Vehicles



Premium Cars 2006:
200-300 MB SW – growing towards 1 GB by 2010
Up to 60 ECUs – not growing

The Autosar Approach

Based in part on presentation by

Christian Salzmann, BMW CarIT
at workshop *Beyond Autosar*

OFFIS

SafeTRANS

ARTIST

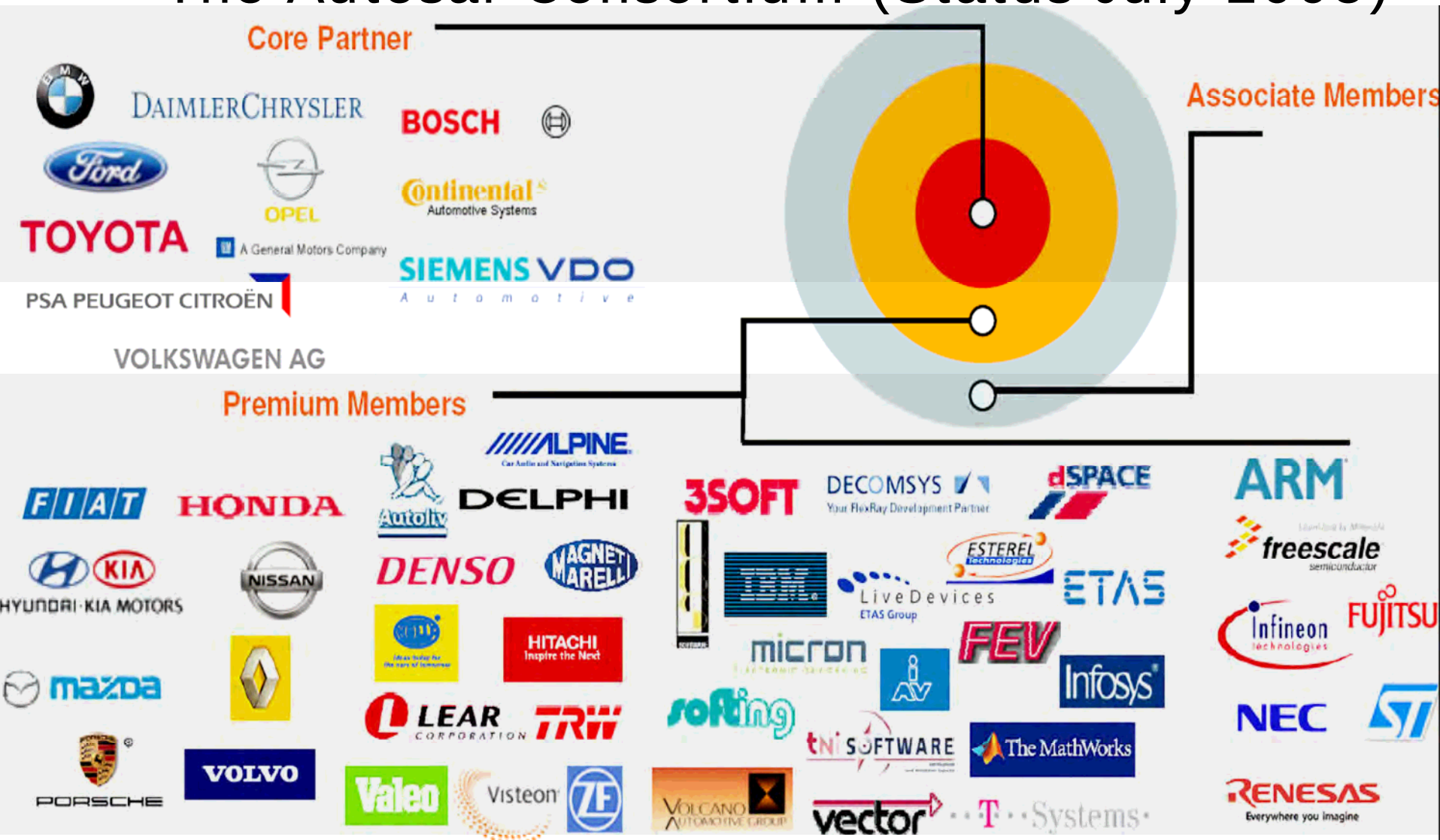
Drivers for Change

- Flexibility
  - Decouple growth rate of #functions from growth rate of #electronic components
  - Freedom in choosing boundary of in-house and external development
- Adaptability
  - Towards emerging technologies
  - Towards emerging hardware platforms
  - Maintainability : at life-time
- Cost
  - Decouple growth of #functions from growth rate of development costs
  - Decouple growth rate of number of supported platforms from development costs
- Quality
  - Maintain/Improve Quality while allowing growth of #functions

Anticipated Changes in Processes

- Strong push to virtual subsystem models (function-level) for time reduction
  - Target independent
  - Topic in Autosar
- Strong push towards component based development
  - Topic in Autosar
  - Requires component characterizations dealing with non-functional aspects (e.g. real-time, safety, ...)
- Need to boost quality
  - to support IEC 61508 customized to automotive domain – safety cases
  - Reduce number of re-calls
  - Topic in Autosar
- Deployment analysis capabilities will be key competence
  - for price-competetive offerings of tier 1 supliers
  - For realizability analysis of new functions for innovator OEMs

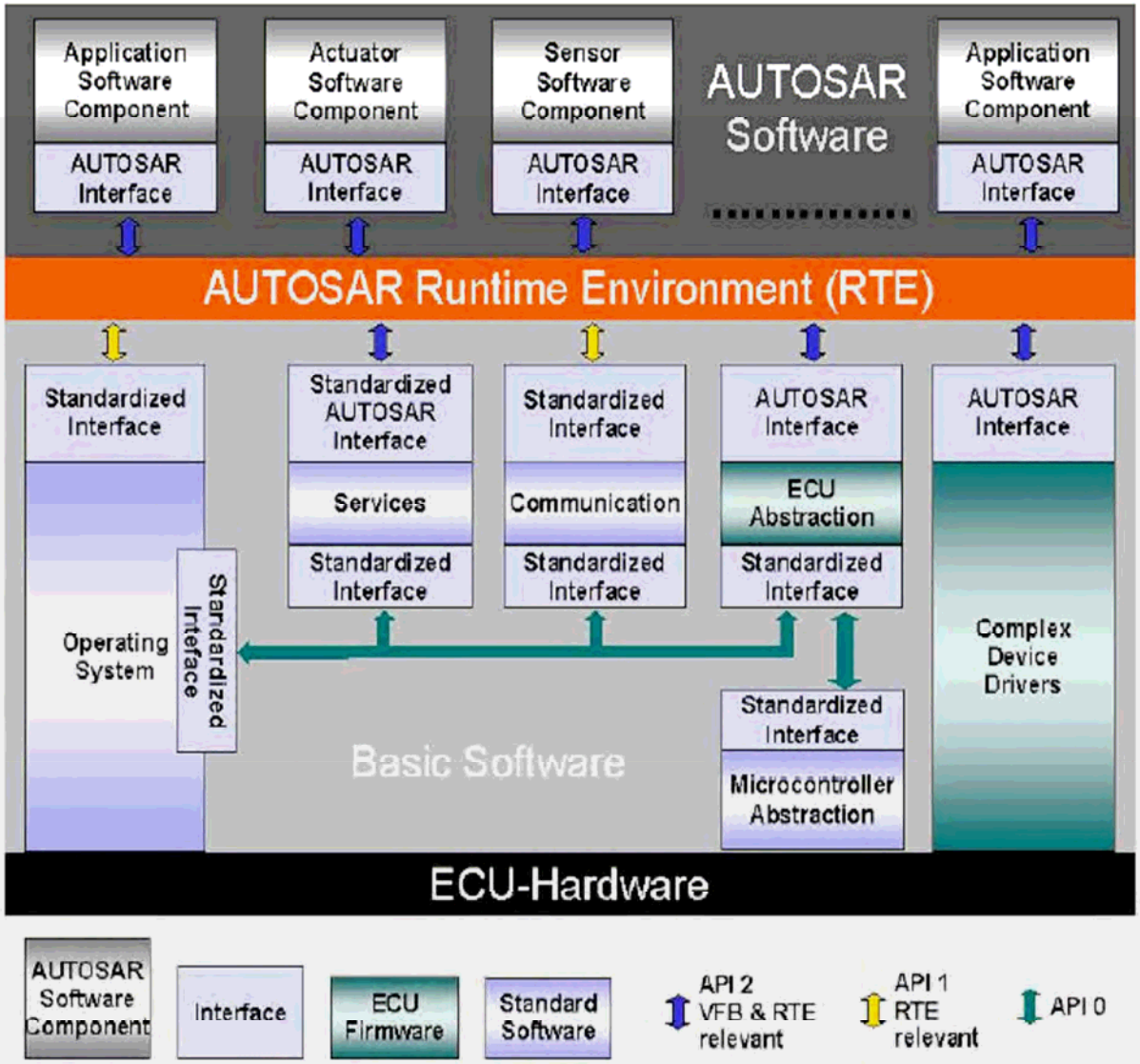
The Autosar Consortium (Status July 2005)



General OEM	Generic Tier 1	Standard Software	Tools and Services	Semi-conductors

Standardization

AUTOSAR – ECU Software Architecture



Automotive Open System Architecture (AUTOSAR):

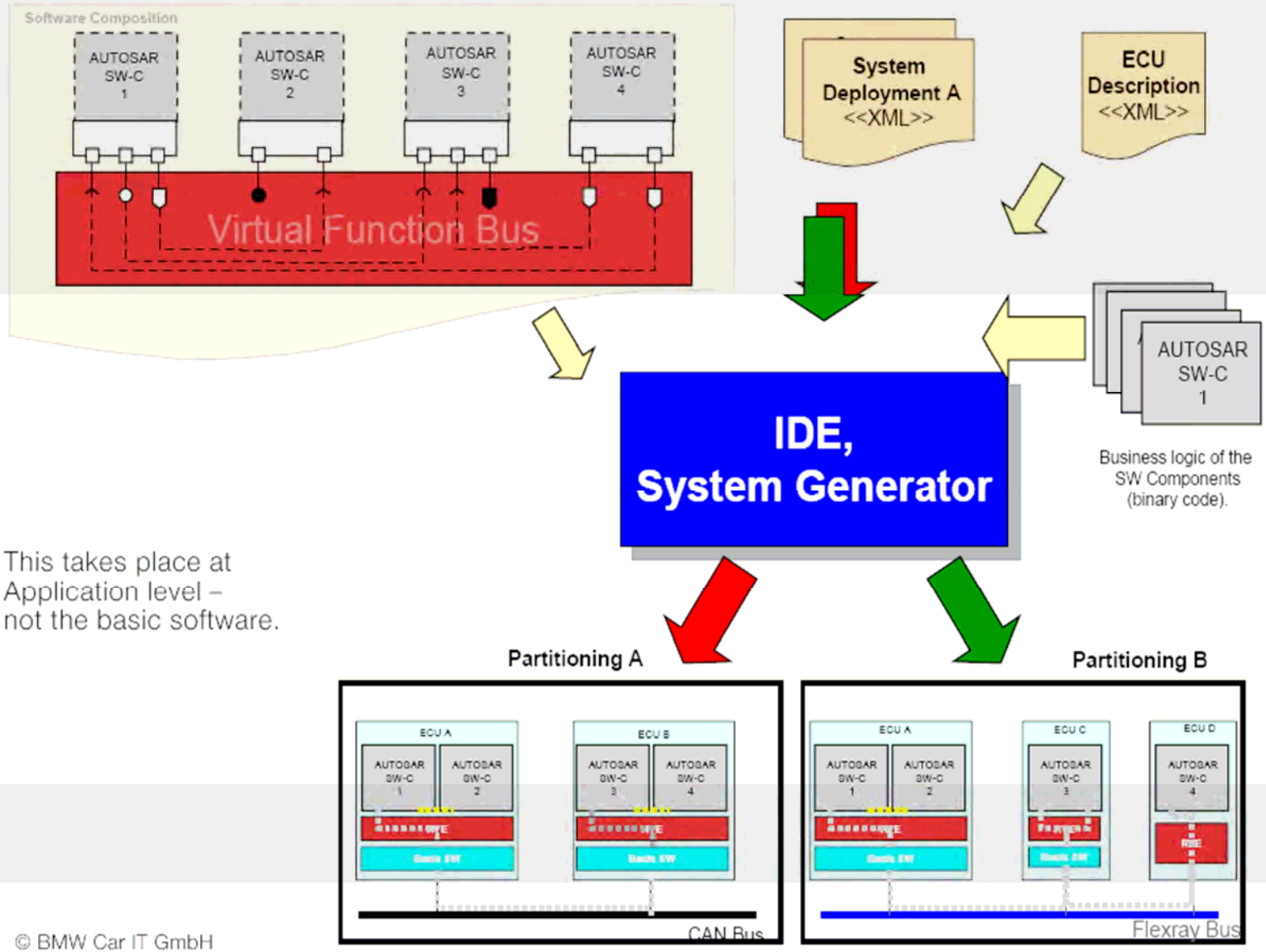
- Standardized, openly disclosed interfaces
- HW independent SW layer
- Transferability of functions
- Redundancy activation

AUTOSAR RTE:

by specifying interfaces and their communication mechanisms, the applications are decoupled from the underlying HW and Basic SW, enabling the realization of Standard Library Functions.

AUTOSAR - First Experiences.

Model based development under AUTOSAR.

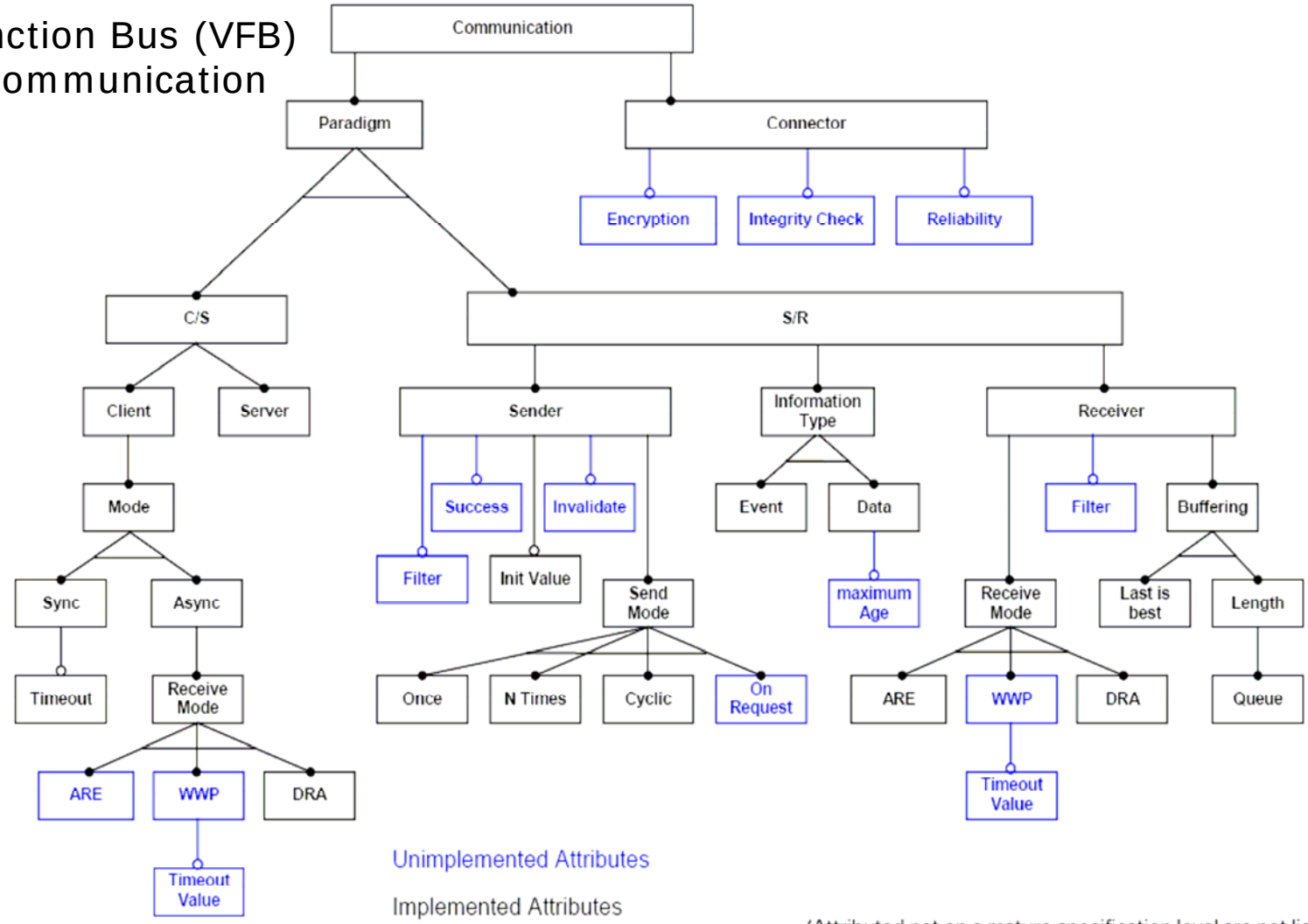


This takes place at Application level – not the basic software.

AUTOSAR – Migration Strategy.

A stable subset of the VFB for migration.

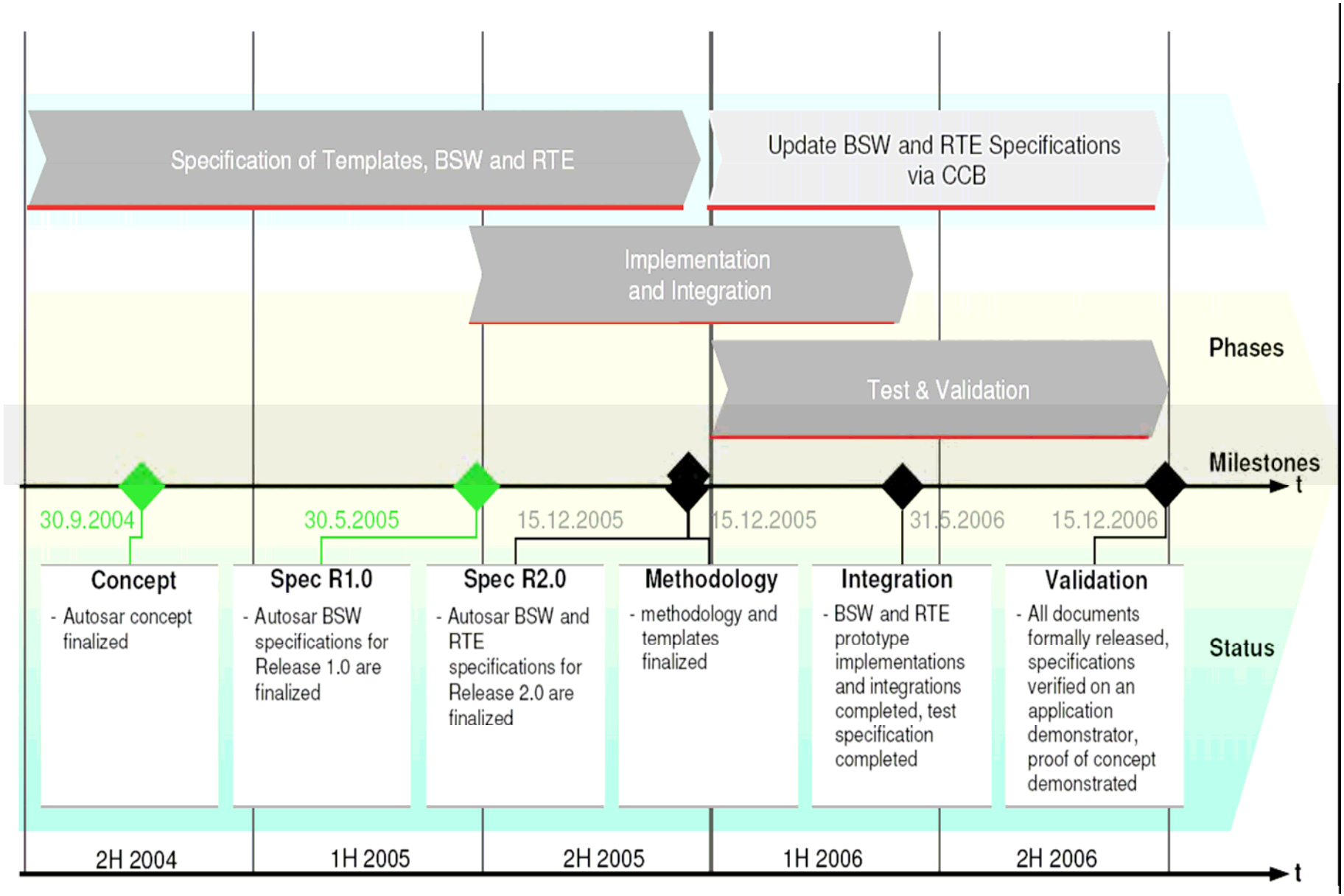
Virtual Function Bus (VFB) offers 48 communication Schemes



(Attributed not on a mature specification level are not listed)

T. Scharnhorst et al, VDI report 1907, 2005

Autosar Schedule



Highlights

- Strong industrial take up
  - Large privat investment: equivalent to 175 full time staff
  - Accepted on international scale
  - Strong vendor involvement
- Autosar Metamodel defined in UML/OCL
  - description of SW-Cs, their interfaces and resource needs
  - description of HW resources, network topologies and communication matrices (covering CAN, LIN and FlexRay).
- Pilot Powertrain demonstration 2005 demonstrated complete flow with minimal overhead against conventional implementation
  - Key to success is to be able to compile away RTE for given configuration (similar to OSEK approach)
- Phase 2 will push towards strong deployment

Impact and Challenges on Real-Time Analysis

based in part on presentation of

Kai Richter, Symtavision GmbH

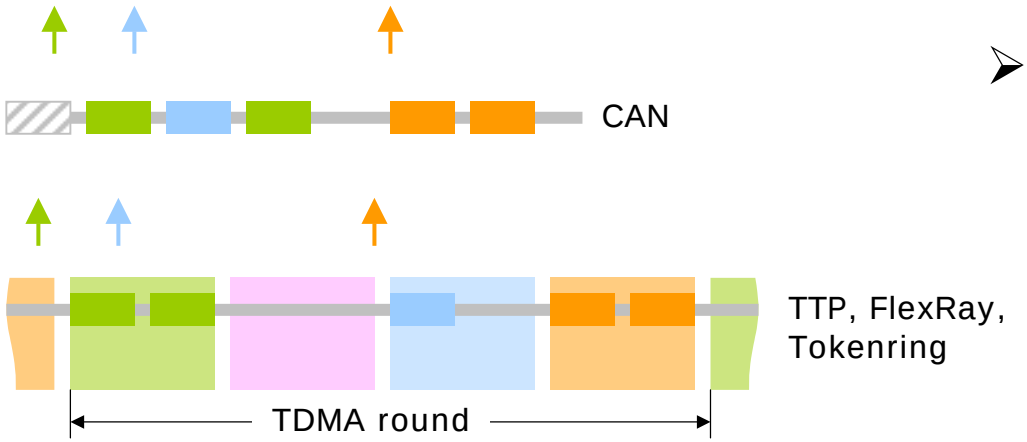
Workshop Beyond Autosar

OFFIS

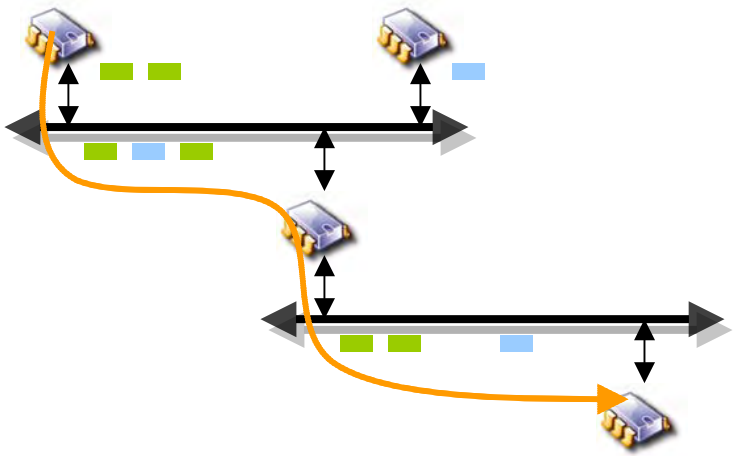
SafeTRANS

ARTIST

Real-Time view: Communication Issues



Real-time analysis for message transmission similar to response time analysis of task systems



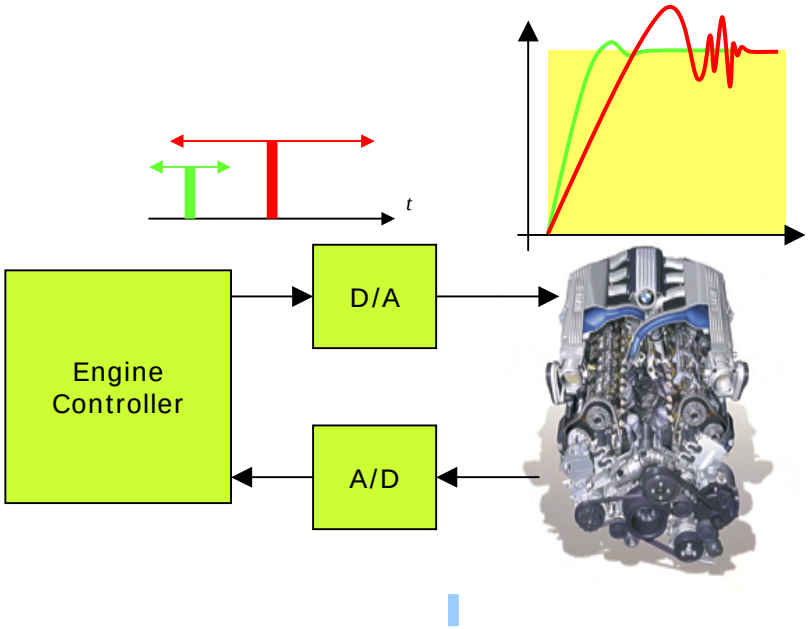
- Different paradigms of bus access for event triggered messages:
  - Event triggered (similar to preemptive, priority driven scheduling) e.g. CAN
  - Time triggered (nodes are allowed to send only at fixed time instants) e.g. TTP, FlexRay
- Fragmentation of messages into packages
- Access to bus controller
- Physical latency of transmission
- Latencies on gateway nodes

Roles

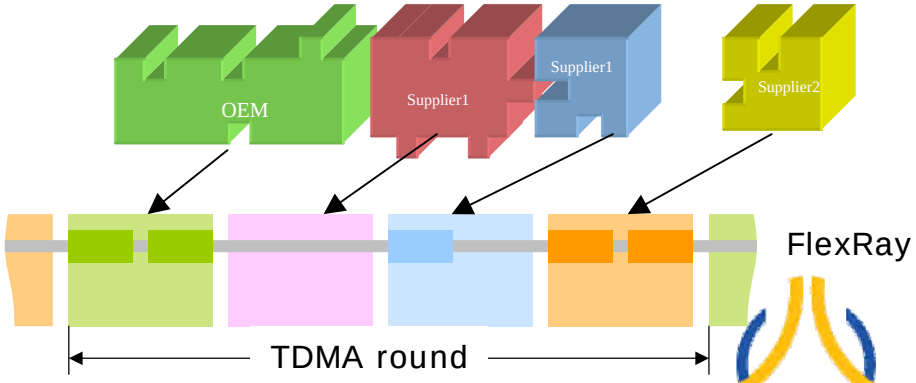
- System Specification
  - Responsibility for design of new subsystems realizing new function
  - Autosar approach: target independent design using function networks
  - Assessment of realizability of new function using sufficiently detailed abstractions of implementation space
  - Pass validated specs to multiple suppliers for design and implementation
  - Pass hardware requirements (e.g. FlexRay based, number and class of ECUs, ...)
  - Integrate provided solutions
- System Design & Implementation
  - Develop detailed spec
  - Derive task structure
  - Explore Design Space for cost-optimal solutions
  - Allocate tasks and messages
  - HW/SW Implementation
- Key issues:
  - Target hardware shared across multiple suppliers
  - Sharing across bus-systems: need to pre-budget communication
  - Sharing of ECUs: HW/SW integration must be done by OEM or single trusted supplier
  - Entails need to support incremental allocation (per supplier)

Real-Time view – Key issues (cont.)

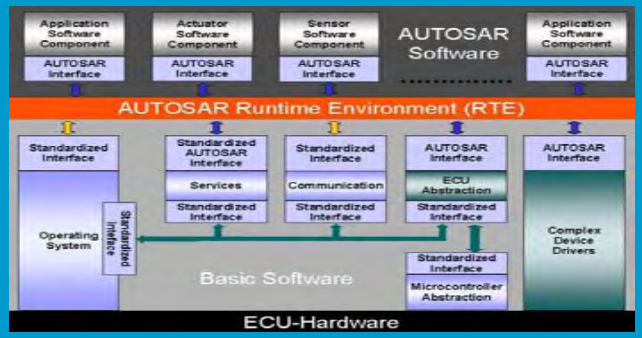
- Incremental Allocation must guarantee component specific maximal jitter
- Typical issue: loss of stability



- Introduction of TDMA based bus systems forces OEMs to perform pre-allocation of TDMA slots to suppliers
- Wrong estimates are potentially expensive: may force re-scheduling of messages of other suppliers
- Key weakness of TDMA based solutions



Integrating Real-Time Analysis Techniques into Autosar Based Design Processes



Hierarchical Task Graphs
End-to-End Latencies

Real-Time Systems Community

Tasks

Messages

Schedulability Analysis

??

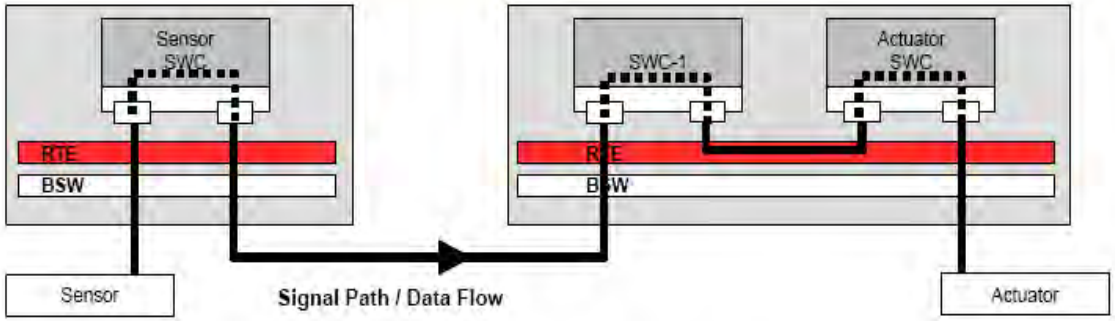
The Basic Dilemma

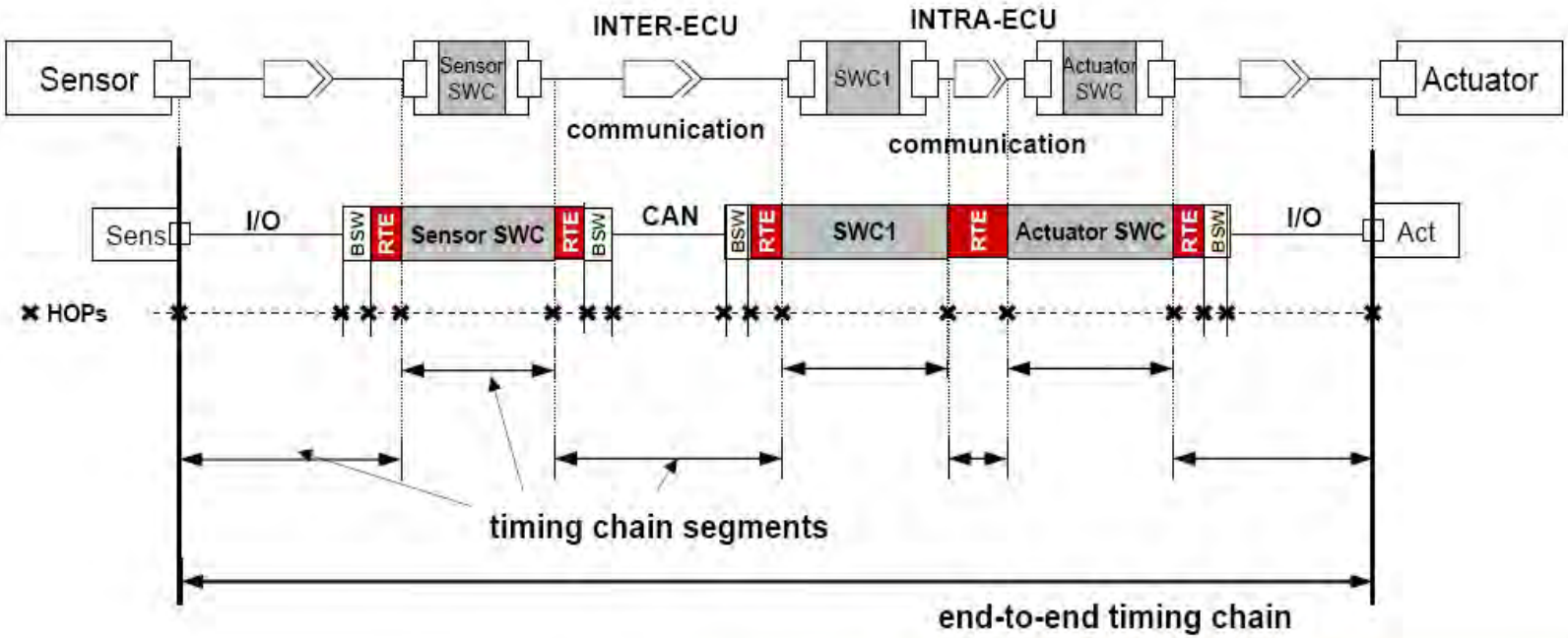
- Autosar is all about decoupling functional design from architecture
- However, response-time analysis is inherently impacted by architectural choices
- Depending on allocation decisions taken late in designs, end-to-end latencies vary drastically from local single ECU implementations to hierarchical distributed designs

Introduction of Timing Effects: Framework

- Function development imposes timing constraints
- High-level specification based on SW components
- AUTOSAR goal: break down the software structure into "manageable" blocks
  - timing chains and timing chain segments
  - connected at hand-over points (HOPs)
  - consider each segment / HOP individually
- Goals:
  - divide and conquer "timing analysis" top-down
  - assignment of responsibilities
  - locally verifiable, then result composition bottom-up

Timing Chains and Hand-Over Points (HOPs)



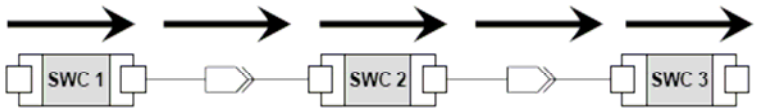


Introduction of Local Timing Effects

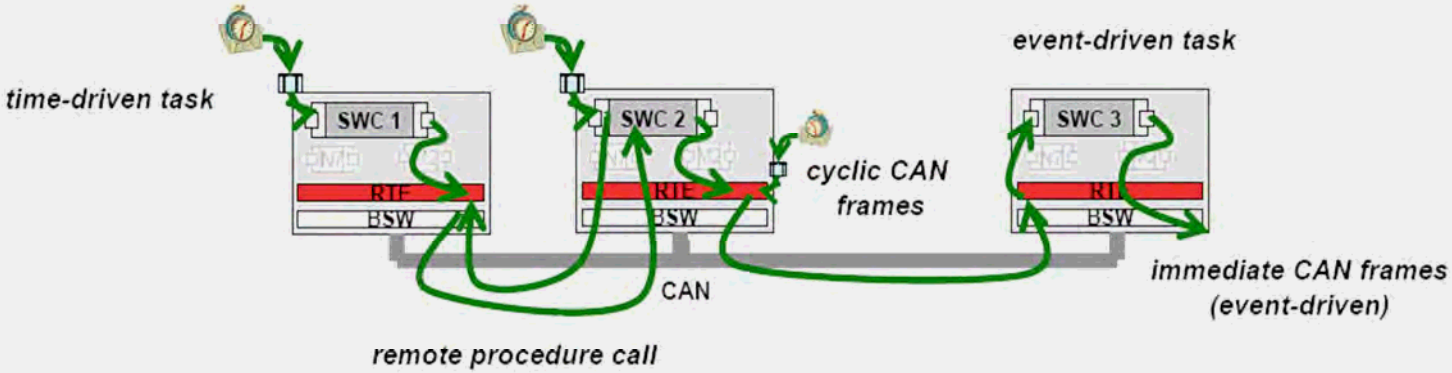
- Reasoning about timing requires considering two views:
  static **software components**
  vs. dynamic **execution platform** behavior
  - operating systems and scheduling;
    SW components vs. runnables and tasks
  - communication semantics;
    SW-C structure vs. timing dependencies
  - middleware / driver structure;
    standardized protocols vs.
    non-standardized implementation & BSW

Software Component Structure vs. Timing Dependencies

- Software component view captures "logical" dependencies (data flow)



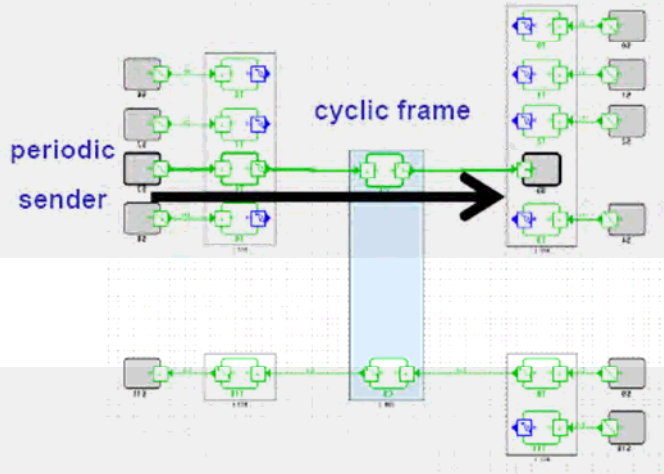
- Implementation timing dependencies can be very different!!!
  - time-driven and event-driven activation
  - send/recv and client/server communication (remote procedure call)
  - over- / undersampling



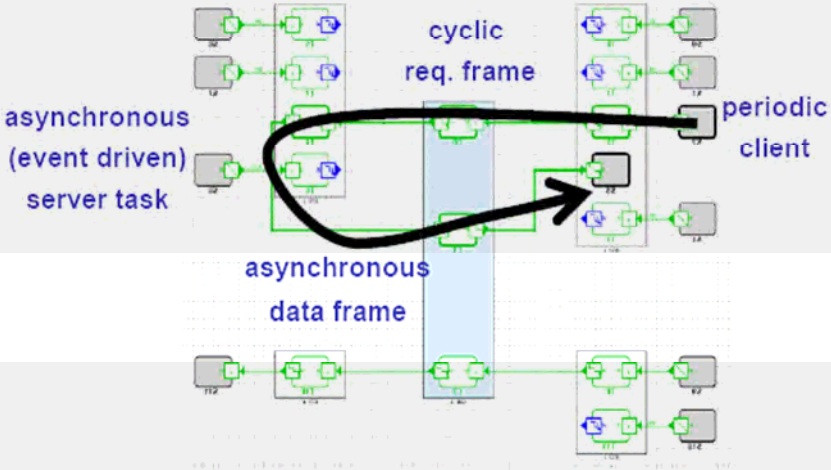
Sender-Receiver vs. Client-Server I

- INTRA-ECU communication: both SW-Cs on one ECU
  - merely an issue of software structure
  - global register vs. local variable (with get Method)

- INTER-ECU communication: SW-Cs on different ECUs
  - has large influence on bus / ECU timing



sender-receiver



client-server

Screenshots by SYMTA

Research Challenges I: Bridging the timing gap

- How can we assess early the impact of architectural choices on key system timing characteristics, such as end-to-latencies, so as to assess the feasibility to realize new automotive functions?
- What architectural abstractions are required to perform such assessments with sufficient precision, thus allowing to narrow down the design space?

Research Challenges II: Bridging the timing gap

- How can we decompose overall timing analysis both horizontally and vertically taking into account responsibilities and roles of OEMs and suppliers?
- Can we develop compositional timing analysis methods allowing to decouple global timing analysis into local analysis within the scope of OEMs/suppliers?
- Which expressiveness for timing interface specifications of components is required to support compositional timing analysis?

Some Links

- Marek Jersak, Kai Richter, Rolf Ernst Performance Analysis for Complex Embedded Applications,
- S. Wang, S. Rho, Z. Mai, R. Bettati, and W. Zhao. Real-time component-based systems. In RTAS, 428-437. IEEE Computer Society, 2005
- I. Shin and I. Lee. Compositional real-time scheduling framework. In RTSS, pages 57.67. IEEE Computer Society, 2004.
- Thomas A. Henzinger, Slobodan Matic, An Interface Algebra for Real-Time Components, In Proc. RTAS, 2006.
- Arindam Chakrabarti, Luca de Alfaro, T.Henzinger, and Marielle Stoelinga, Resource Interfaces, Proceedings EMSOFT 2003, Lecture Notes in Computer Science 2855, Springer, 2003, 117-133.
- Luca de Alfaro and Thomas A. Henzinger. Interface Based Design, In: Engineering Theories of Software-intensive Systems (M. Broy, J. Gruenbauer, D. Harel, and C.A.R. Hoare, eds.), NATO Science Series: Mathematics, Physics, and Chemistry, Vol. 195, Springer, 2005, pp. 83-104.
- E. Wandeler and L. Thiele. Real-time interfaces for interface-based design of real-time systems with priority scheduling. In Proc. EMSOFT 2005, 80-89. ACM, 2005.
- F. Eisenbrand, W. Damm, A. Metzner, G. Shmonin, R. Wilhelm, and S. Winkel. Mapping Task-Graphs on Distributed ECU Networks: Efficient Algorithms for Feasibility and Optimality. In Proceedings of the 12th IEEE Conference on Embedded and Real-Time Computing Systems and Applications. IEEE Computer Society, 2006.
- NoE Artist, www.artist-embedded.org

Impact and Challenges on Control

based in part on presentation of

K.-E. Arzen, Lund University

Workshop Beyond Autosar

OFFIS

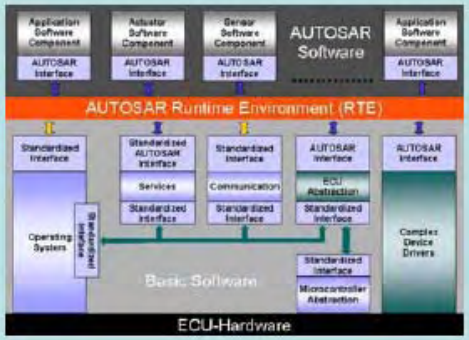
SafeTRANS

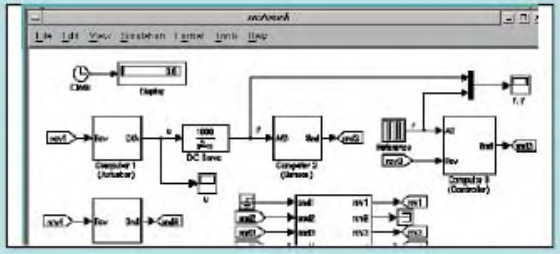
ARTIST

The Role of Control

- Advanced control is absolutely essential in modern cars
  - Powertrain, emissions, vehicle dynamics, safety systems, …
  - ECU rather than CPU
- Control gives performance, safety, and low emissions
- The quality and performance of the control systems must be a top priority

Integrating Control Design in Autosar Based Development Processes





Simulink components

Control Community

Stateflow

Real-Time Workshop

Model-based design

Robustness

Stability

??

"Autosar"

"Matlab/Simulink"

The Basic Dilemma

- Autosar is all about decoupling functional design from architecture
- However, control design is inherently impacted by architectural choices
- Depending on allocation decisions taken late in designs, control-loop implementation varies drastically from tight closed loop control to hierarchical distributed control

Control Loop Timing

- Classical control assumes deterministic sampling
  - in most cases periodic (not engine control)
  - too long sampling interval or too much jitter cause poor performance or instability
    - but, anomalies exist
- Classical control assumes negligible or constant input-output latencies
  - if the latency is small compared to the sampling interval it can be ignored
  - if the latency is constant it can be included in the control design
  - too long latency or too much jitter cause poor performance or instability
    - but, anomalies exist

Networked Embedded Control Timing

- Embedded control systems with limited computing resources may cause temporal non-determinism
  - multiple tasks competing for computing resources
  - preemption by higher-priority tasks, blocking when accessing shared resources, varying computation times, non-deterministic kernel primitives, priority inversion, …
- Networked control systems with limited communication resources may cause temporal non-determinism
  - network interface delay, queuing delay, transmission delay, propagation delay, link layer resending delay, transport layer ACK delay, ...
  - lost packets

Research Challenges

- How can we assess early the impact of architectural choices on stability and controllability? What architectural abstractions are required to perform such assessments with sufficient precision?
- How can we design control strategies sufficiently robust so as to “smoothly degenerate” when implemented in a distributed fashion? Can we learn from the analogy to QoS requirements in soft real-time vs hard real-time?
- What degree of determinism must be provided by interconnects? E.g. trade off between latency and determinism between time-triggered and event triggered solutions.
- How can we re-use control-components in spite of possible drastically varying architectural choices in given implementations (tied to Q1)?
- How can we assure key control properties such as stability (or stronger variants) in a compositional way? C.f. also work on distributed implementation of self-stabilizing algorithms.

Some Links

- See http://www.control.lth.se/documents/2005/hen+05survey.pdf for survey of tools for real time control systems co-design
- See NoE HyCon http://www.ist-hycon.org/
  - E.g. A. Balluchi, L. Benvenuti, S. Engell, T. Geyer, K.H. Johansson, F. Lamnabhi-Lagarrigue, J. Lygeros, M. Morari, G. Papafotiou, A.L. Sangiovanni-Vincentelli, F. Santucci, O. Stursberg, Hybrid control of networked embedded systems, Special Issue of the European Journal on Control, Fundamental Issues in Control, vol. 11 no. 4-5, pp. 478--508, 2006.
- See German Priority Research Theme on Distributed Control http://spp-1305.atp.rub.de/
- See NSF program on embedded and hybrid systems, e.g. CHESS at UCB

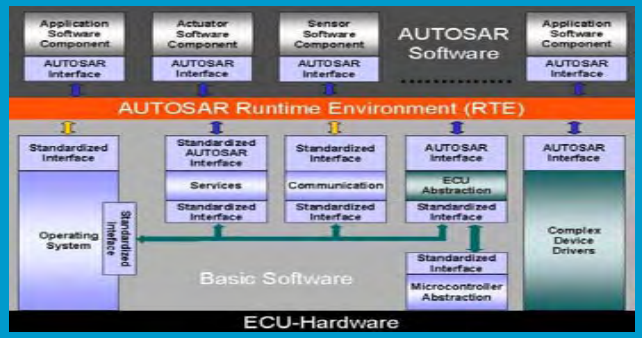
Impact and Challenges on Safety

OFFIS

SafeTRANS

ARTIST

Integrating Safety Analysis Techniques into Autosar Based Design Processes



ISO WD 26262
ASIL Levels

Safety Plan
Safety Cases

FMEA, Fault Trees
Common Cause Analysis

Failure Hypotehesis

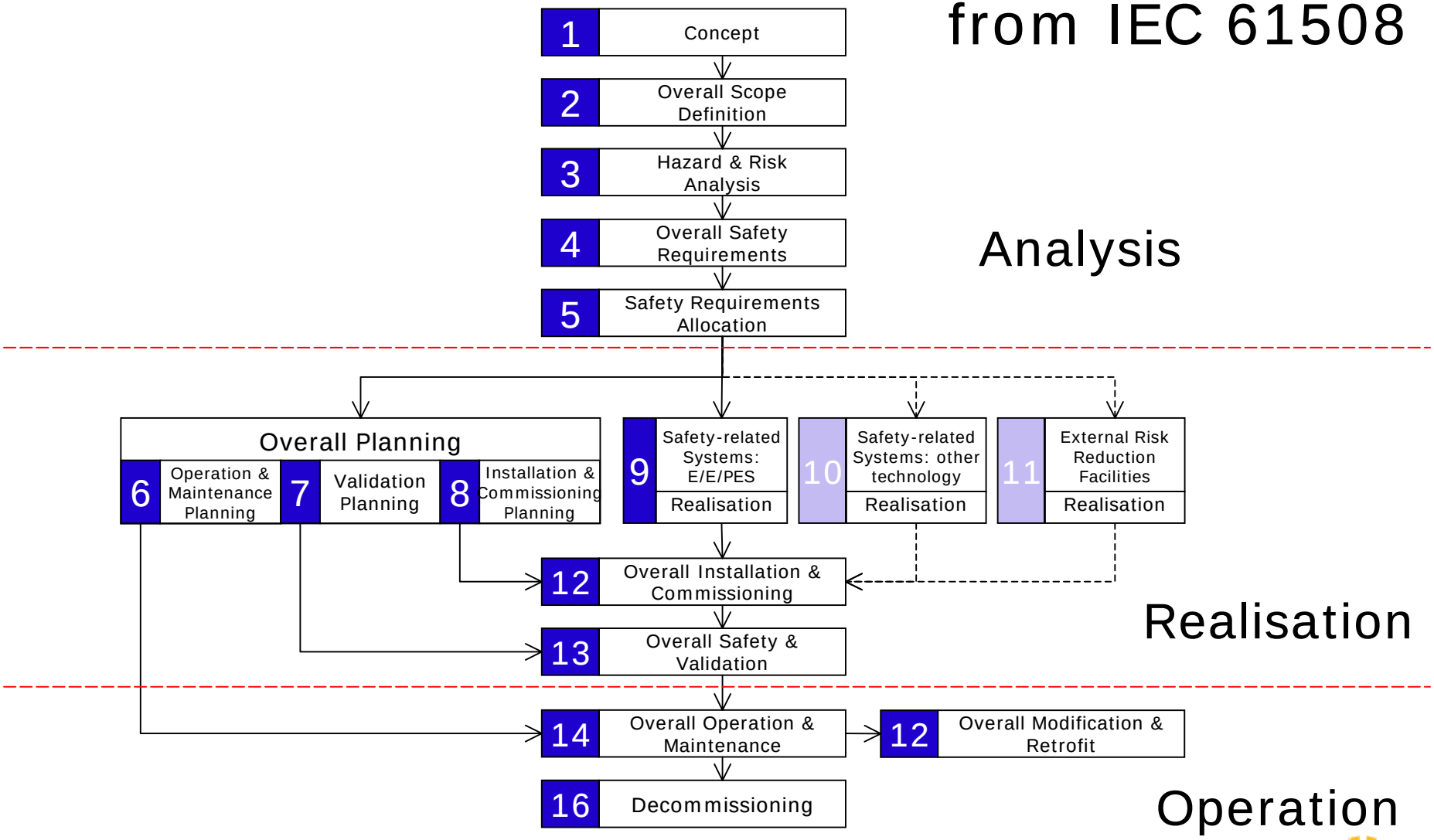
Functional Safety

??

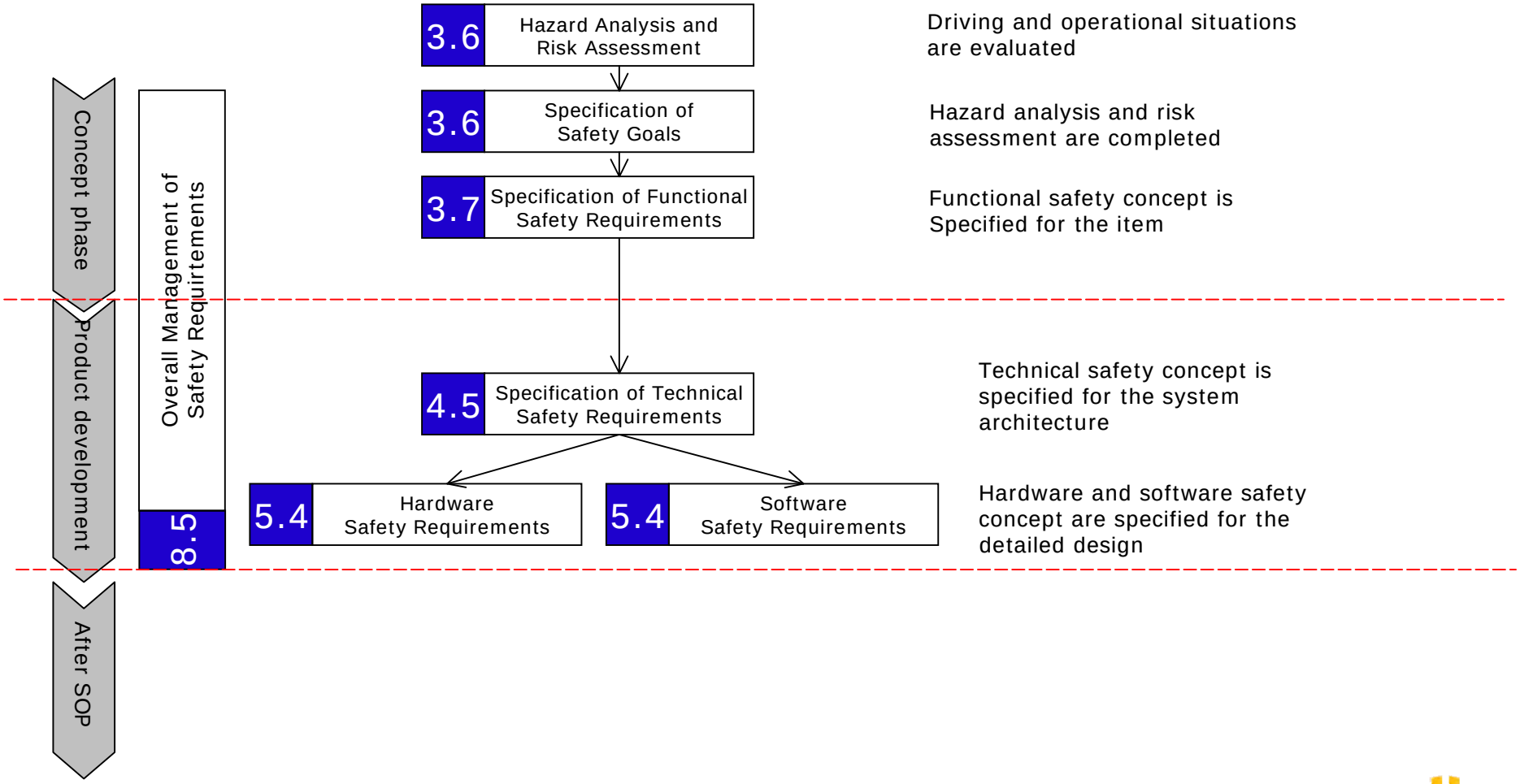
ISO WD 26262 – a forthcoming safety standard for the automotive industry

- IEC 61508 Metanorm for Safety Critical Systems
- Many application domains have derived domain specific versions of this metanorm
  - E.g. CENELEC EN 50126, 50128, 50129 for Railway Systems
- Ongoing initiative to establish harmonized derivation of IEC 61508 for automotive applications
  - No public draft available
- Calls for establishment of safety cases
- Consideration of availability and safety top priority in Autosar

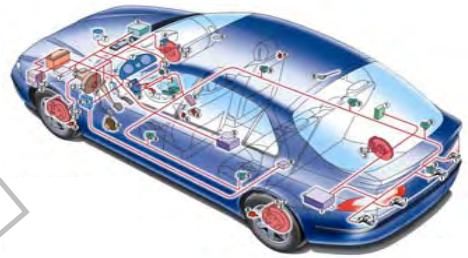
Functional Safety Life Cycle from IEC 61508



Derivation of Safety Requirements



Relevant analysis types



Human Error

Will an erroneous driver action lead to a safety requirement violation?

Testability

Is it possible for a fault to occur undetected?

Operational Reliability

Is it possible to continue driving in a failed configuration?

Common Cause Analysis

Will a list of impacted items violate independency assumptions of a functional model?

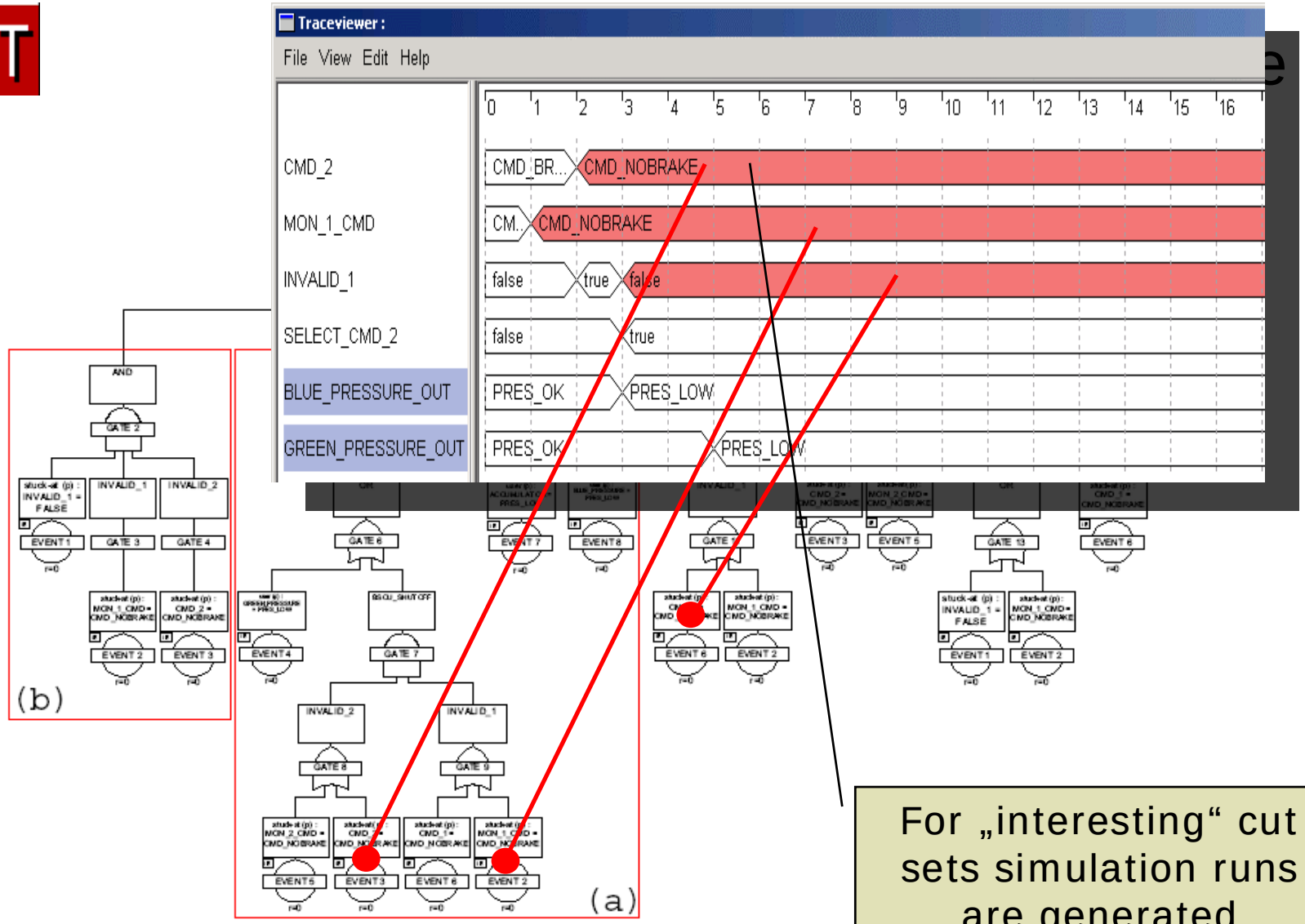
Safety

Is it possible to violate a certain safety requirement?

Experimental Infrastructure
DLR IFS View Car

- Research car with extensive equipment for the analysis of drivers‘ behaviour in different situations.
- Sensors for driver, vehicle and environment

ARTIST



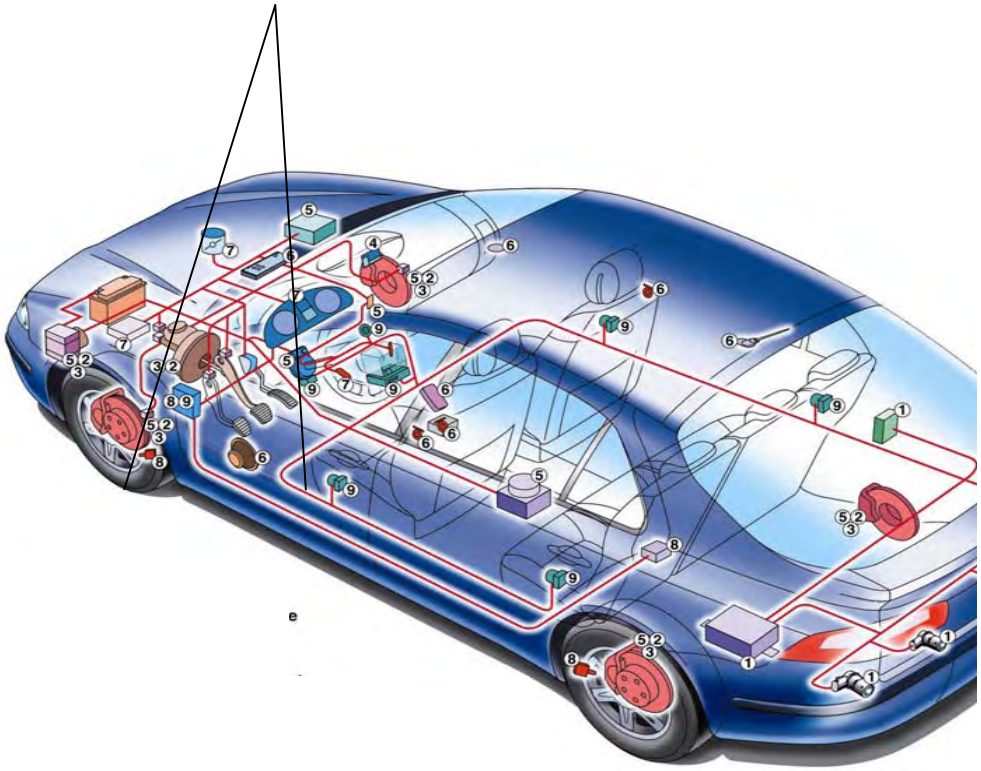
For „interesting“ cut sets simulation runs are generated

Loss of braking should not occur

Example: Common Cause Analysis

- Redundancy is an important architectural feature for safety-critical systems
- Redundancy ensures that failures are (stochastically) independent
- Common cause failures invalidate independency assumptions

Example: Short circuit deactivates redundant sensors



Example: Testability

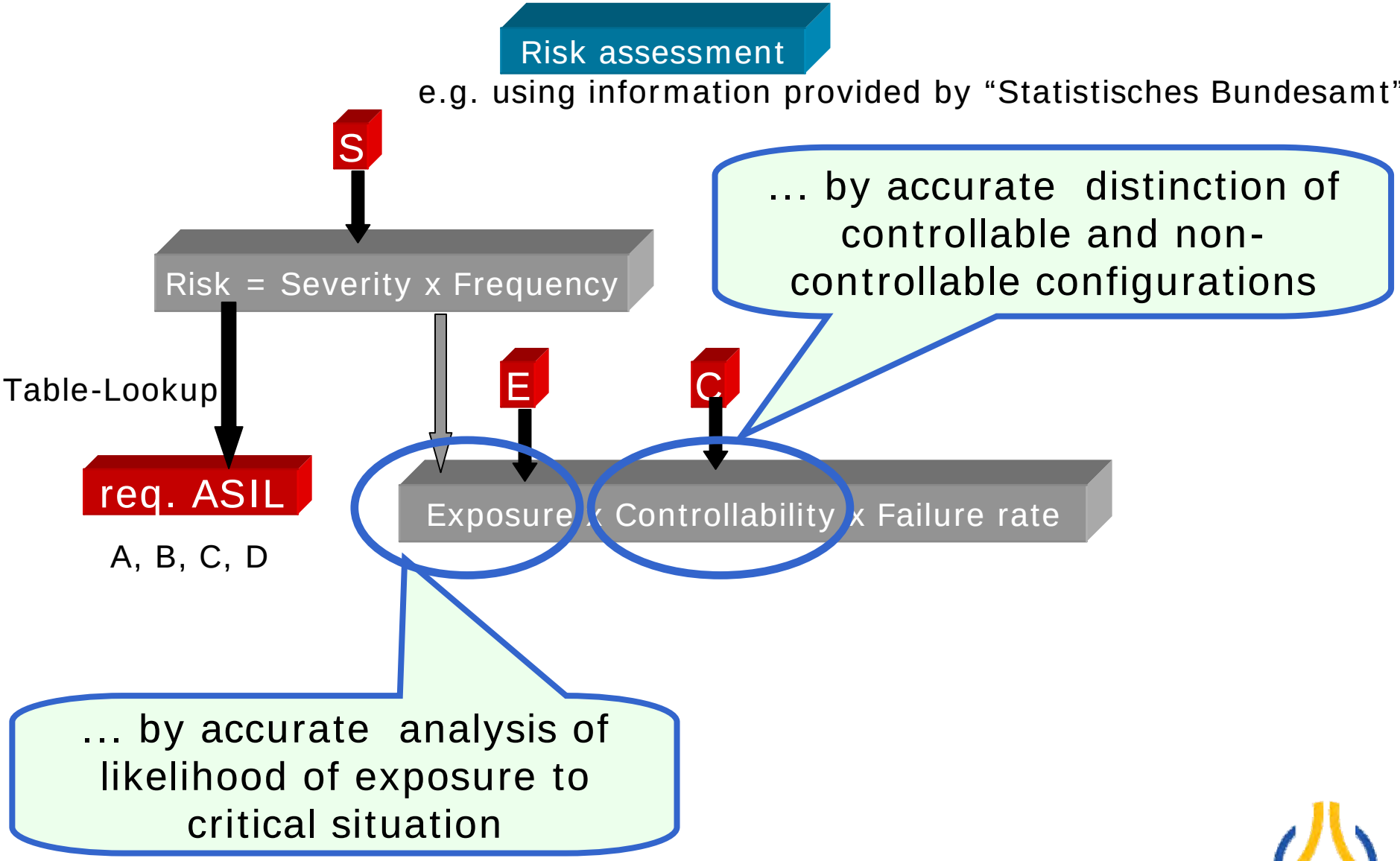
- Subject of the analysis is the Built In Test Equipment (BITE)

- The testability of the BITE is the capability of the system
  - to detect its safety critical fault configurations,
  - to alert the driver about the occurrence of unsafe system operating conditions through the generation of appropriate warnings and, possibly,
  - to take (or suggest) corrective actions

- The goal of the testability analysis is to check to what extent the above objectives are met

- The need is to design and verify testability properties of a system in an effective way

Quantitative Aspects



Research Challenges I:

- How can we assess early the impact of architectural choices on key system safety aspects, so as to assess the feasibility to realize new automotive functions?
- What architectural abstractions are required to perform such assessments with sufficient precision, thus allowing to narrow down the design space?

Research Challenges II:

- How can we decompose overall safety analysis both horizontally and vertically taking into account responsibilities and roles of OEMs and suppliers?
- Which expressiveness for safety interface specifications of components is required to support compositional safety analysis?

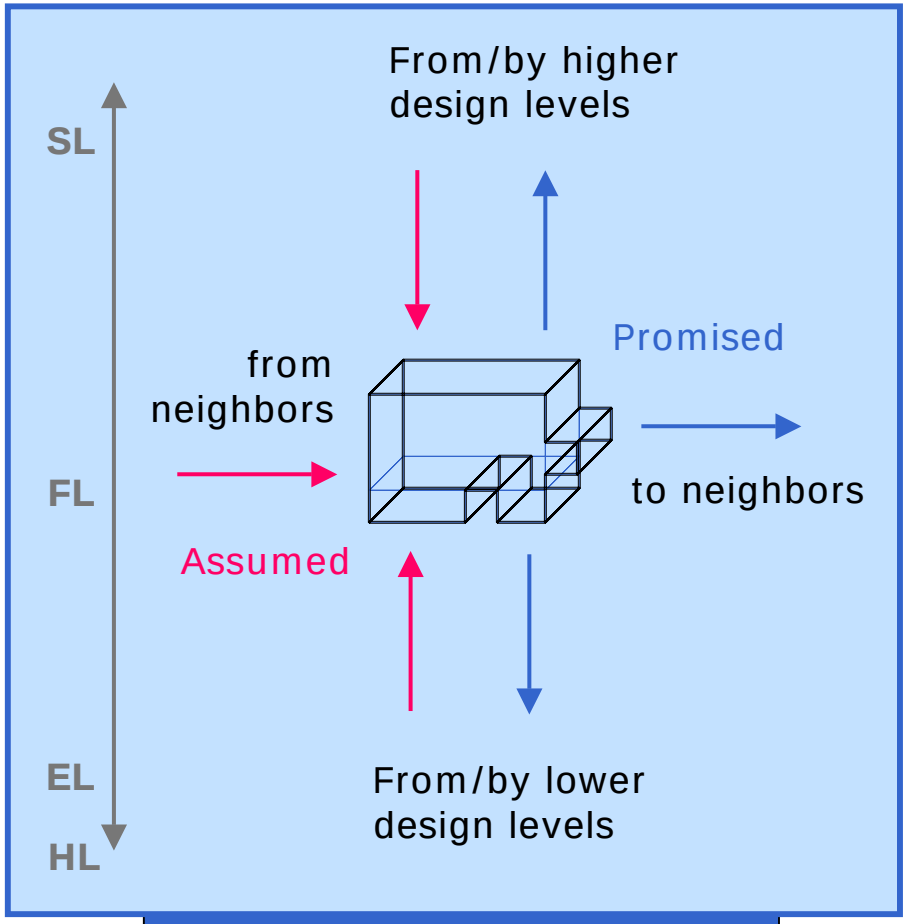
The Speeds Answer

Technical Highlights of the IP Speeds

- Speeds provides
  - The capability of Modeling and Integration of Architectural Abstractions at all System Design Levels for multiple viewpoints including real-time and safety
  - A Rich Component Model allowing to completely encapsulate functional and non-functional aspects of a design in an assume-guarantee style with cross viewpoint dependencies, including the capability of expressing assumptions on lower design levels captured as architectural abstractions
  - A harmonized meta-model allowing a semantic integration of industry standard system- and software design tools supporting rich components based on an open tool integration standard, compatible with the Autosar Metamodel
  - A suite of compositional analysis and design space exploration methods supporting real-time and safety analysis

Rich Components

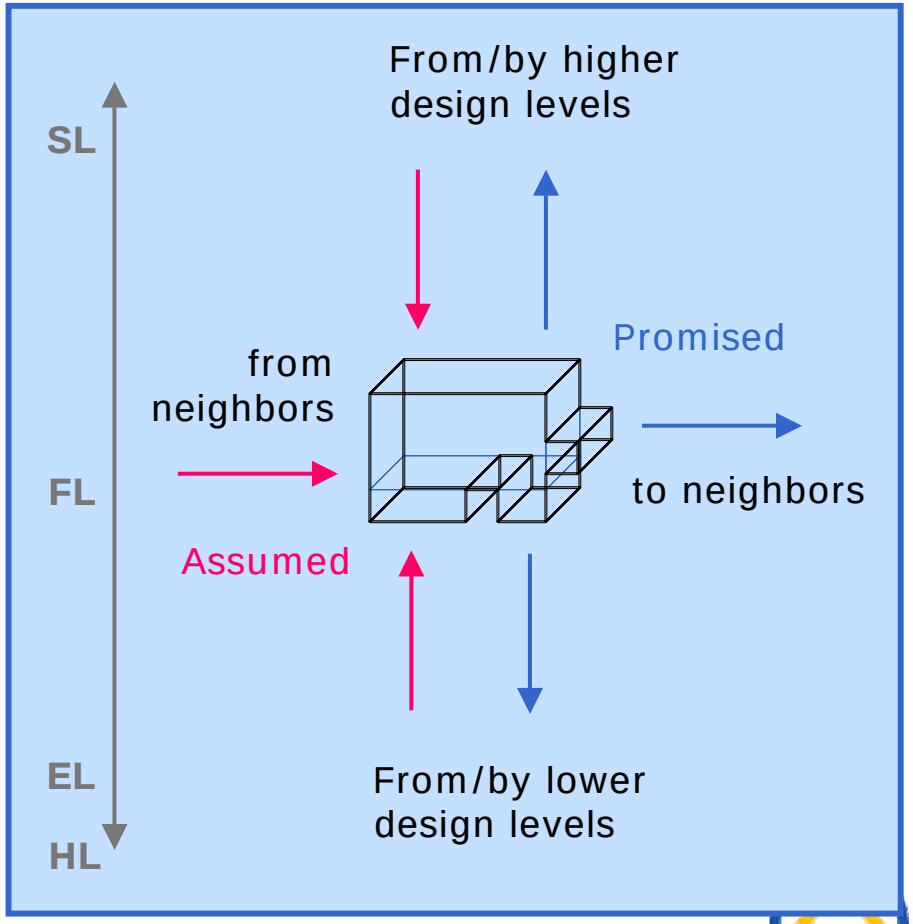
- A component : fully re-usable design artifact providing a well defined functionality
  - Application level functionality
    - "features" of application level functions – level of granularity determined by need to customize application level function
  - Middleware components
  - Hardware components
- „Rich"
  - Explicates all assumptions and/or dependencies on its design context
  - Such that assessment to functional and non-functional characteristics can be made without assessing component itself
- Component Characterization
  - For all viewpoints
    - Safety, Reliability, Real-Time, Power, Bandwidth, Memory consumption, behaviour, protocols



SL:	System Layer
FL:	Functional Layer
EL:	ECU Layer
HL:	Hardware Layer

Rich Component Model

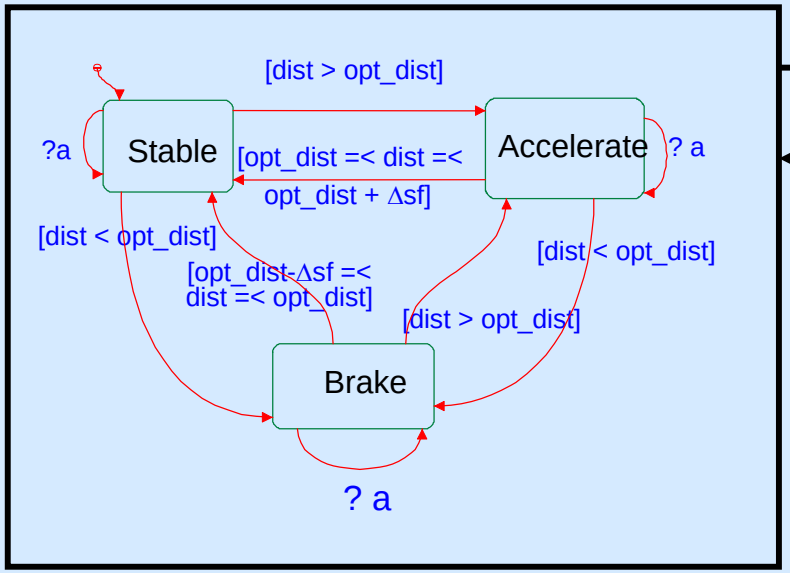
- Assumptions
  - reflect incomplete knowledge of actual design context
  - Determine boundary conditions on actual design context for each view-point under which component is promising its services
  - are decorated with confidence levels
- Promises
  - Are guaranteed if component is used in assumed design context
- Tradeoff
  - Accuracy of promises dependent on stringency of assumptions
  - High accuracy restricts implementation space
- Viewpoint specific models
  - Explicate dependency of promises on actual guarantees by design context
- Component characterization
  - For all viewpoints
    - Safety, Reliability, Real-Time, Power, Bandwidth, Memory consumption, behaviour,

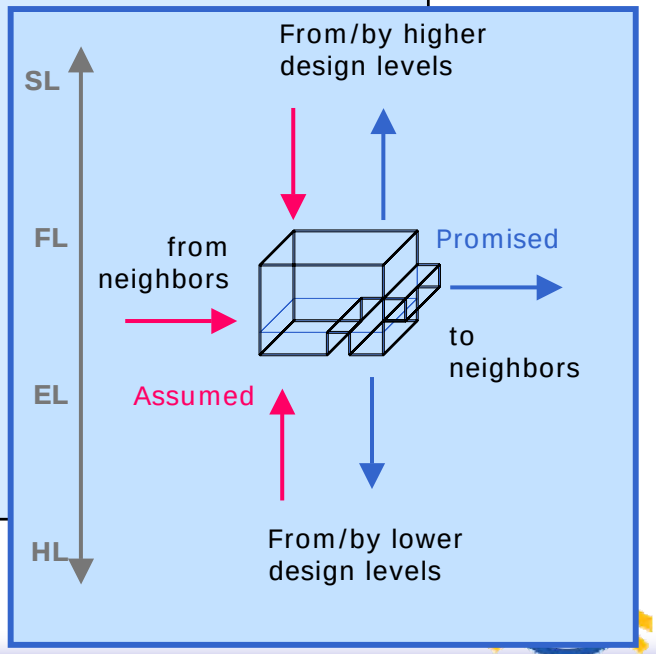


nponent Models (Functional View)

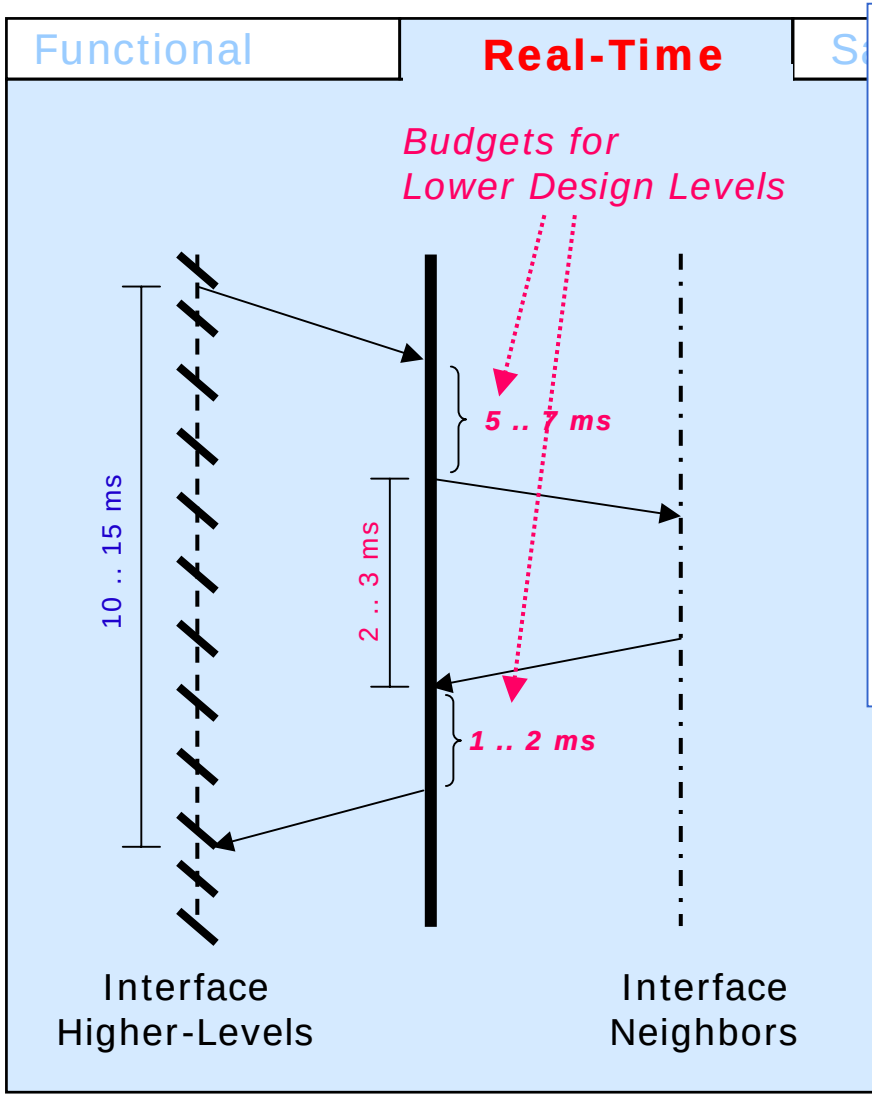
Functional | Real-Time | Safety | ...

- Horizontal assumption:
  Environment will provide the requested data
- Vertical assumption:
  The communication layer guarantees the transmission of every message





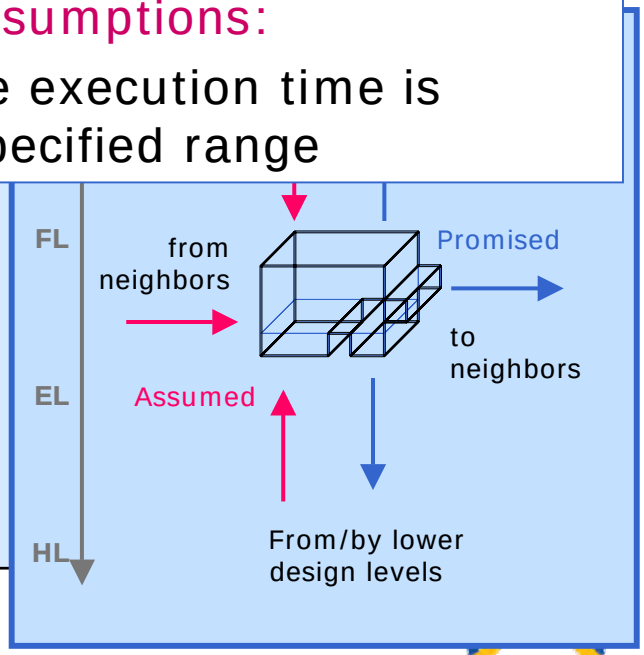
mponent Models (Real Time View)



- explicate dependency of promises on actual guarantees by design context for **real-time** properties using Live Sequence Charts
- Horizontal Assumptions:

  Requested information will be delivered within a specified time frame
- Vertical Assumptions:

  Worst case execution time is within a specified range



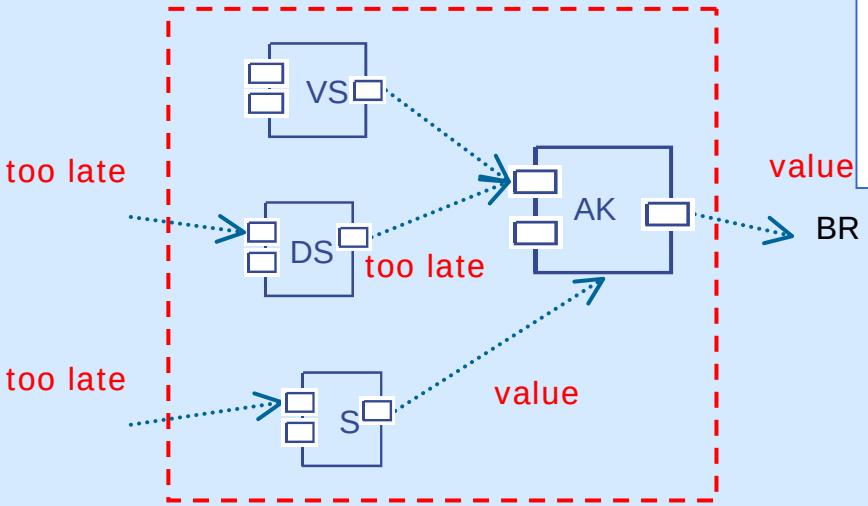
Component Models (Safety View)

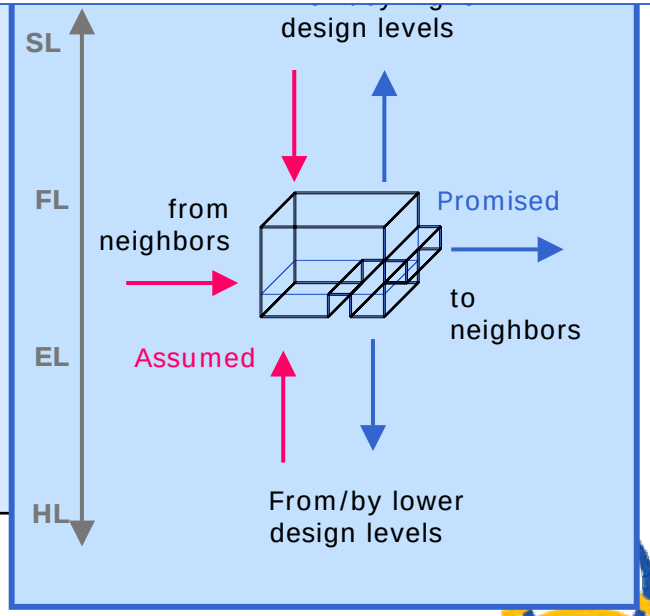
Functional	Real-Time	**Safety**	...

- explicate propagation of failures in a conceptual model for a preliminary safety assessment
- Horizontal Assumptions:

  Failure modes/rates for required information
- Vertical Assumptions:

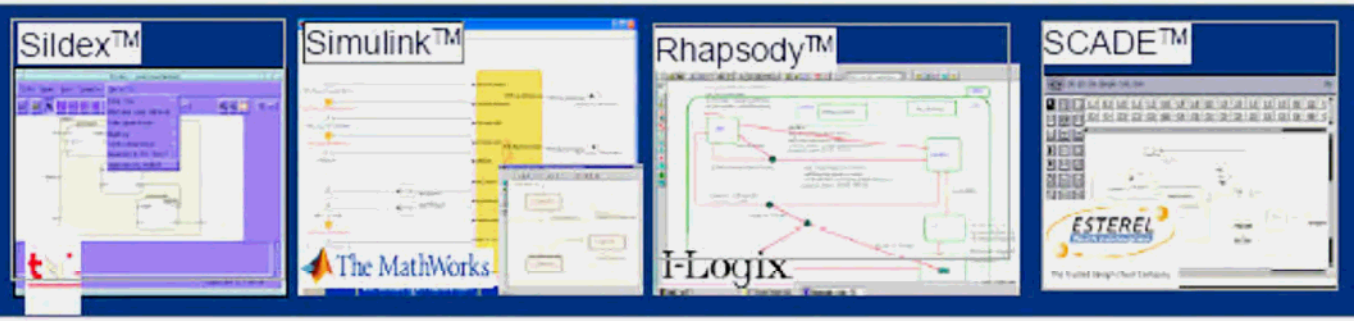
  Failure rates for each failure mode



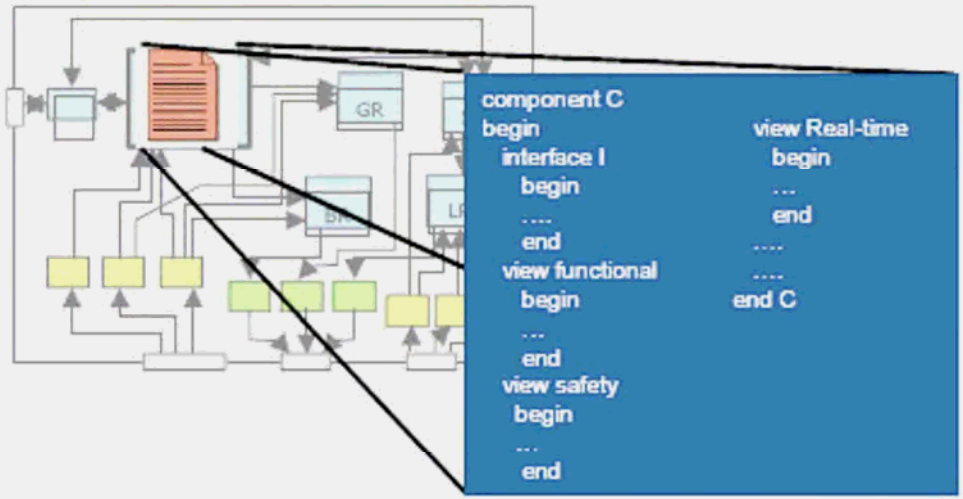


Speeds: Semantic Based Integration

User's View: COTS modeling tools



SPEEDS Meta-model



SPEEDS Semantic Foundation

for all viewpoints v:

$\cap L(A(OutI.v.pr_j)) \subseteq$

$\cap L(A(InI.v.assm_i))$

Speeds Affiliated Partners

- Speeds offers key users, vendors, and research organizations the capability to become “affiliated partners”
  - Participate in requirement analysis phase
  - Early access to project results
  - Participate in evaluation activities
- Current affiliated partners include
  - BMW, Carmeq, Continental, VW
- Request with profile of the applying institution and relevance to Speed should be directed to the projected coordinator:

Gert Döhmen
Airbus
Kreestlag 10
D-21129 Hamburg
Gert.doehmen@airbus.com

speeds
EUROPE'S BEST IN SYSTEM DEVELOPMENT

Conclusion

OFFIS

SafeTRANS

ARTIST

Conclusion

- Autosar opens the way to significant reductions in automotive electronic systems development time and costs
- The separation from function and implementation
  - Is a key enabler towards this objective
  - Induces significant challenges in establishing seamless processes addressing real-time, control, and safety
- The integrated project Speeds addresses these challenges
  - Focus on behavioral modeling, real-time and safety
  - Rich Component Model extendible to other viewpoints