Automotive System and Software Architecture

Yanja Dajsuren

2IW80 Software Specification and Architecture

March 24, 2015

Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

Boeing 787, the most electronic airliner

Chevrolet Volt, an example modern day car

~8,000,000 LOC

~40,000,000 LOC

Why more software?

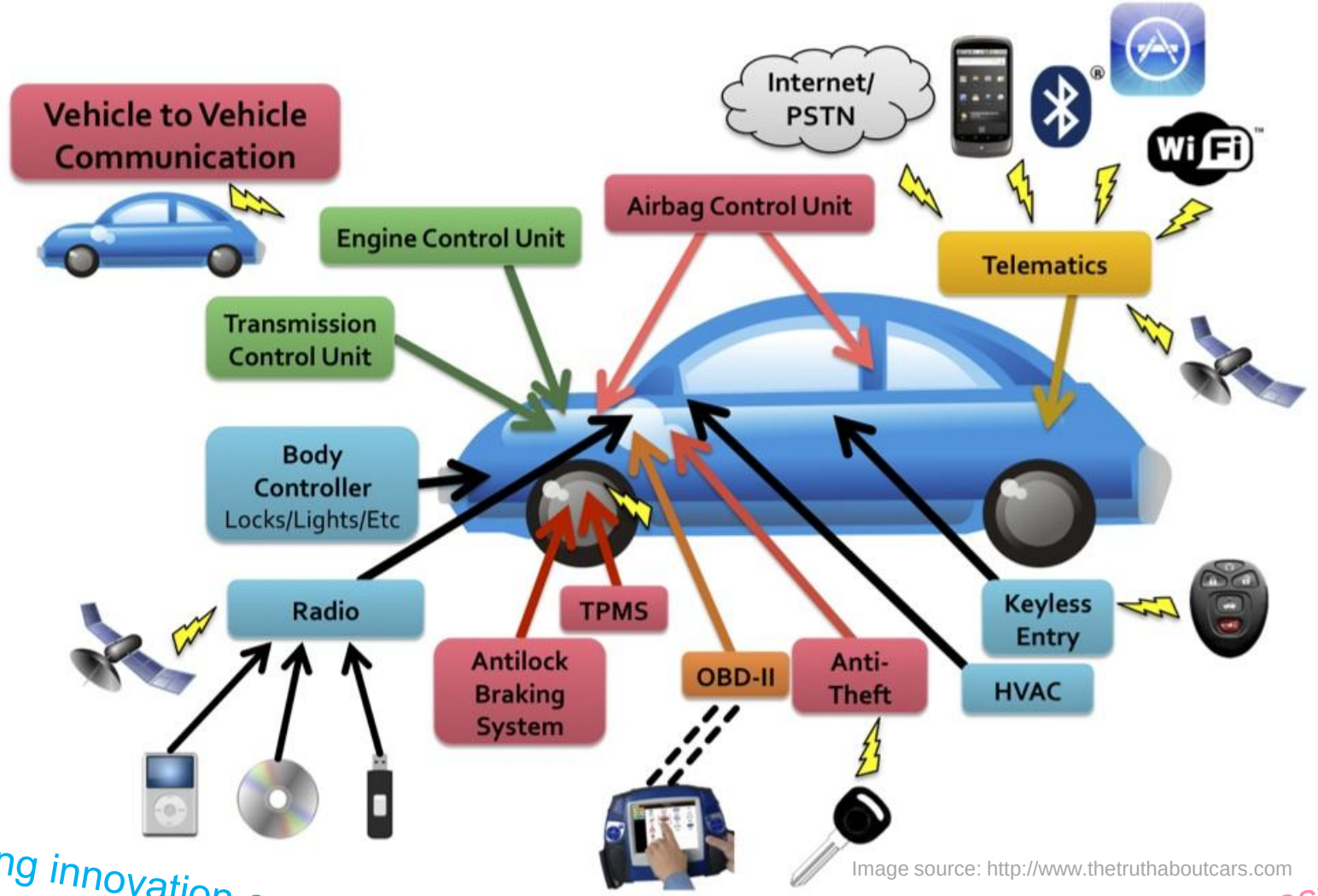


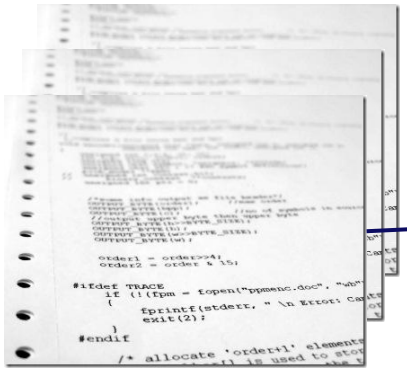
Image source: http://www.thetruthaboutcars.com

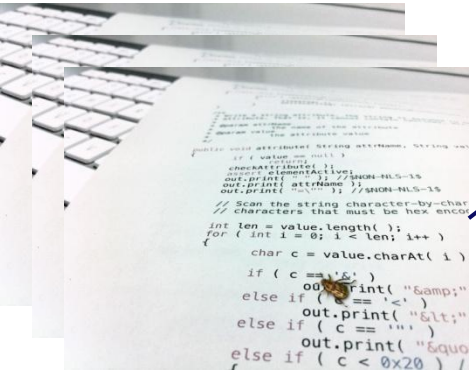
Enabling innovation easier and cheaper

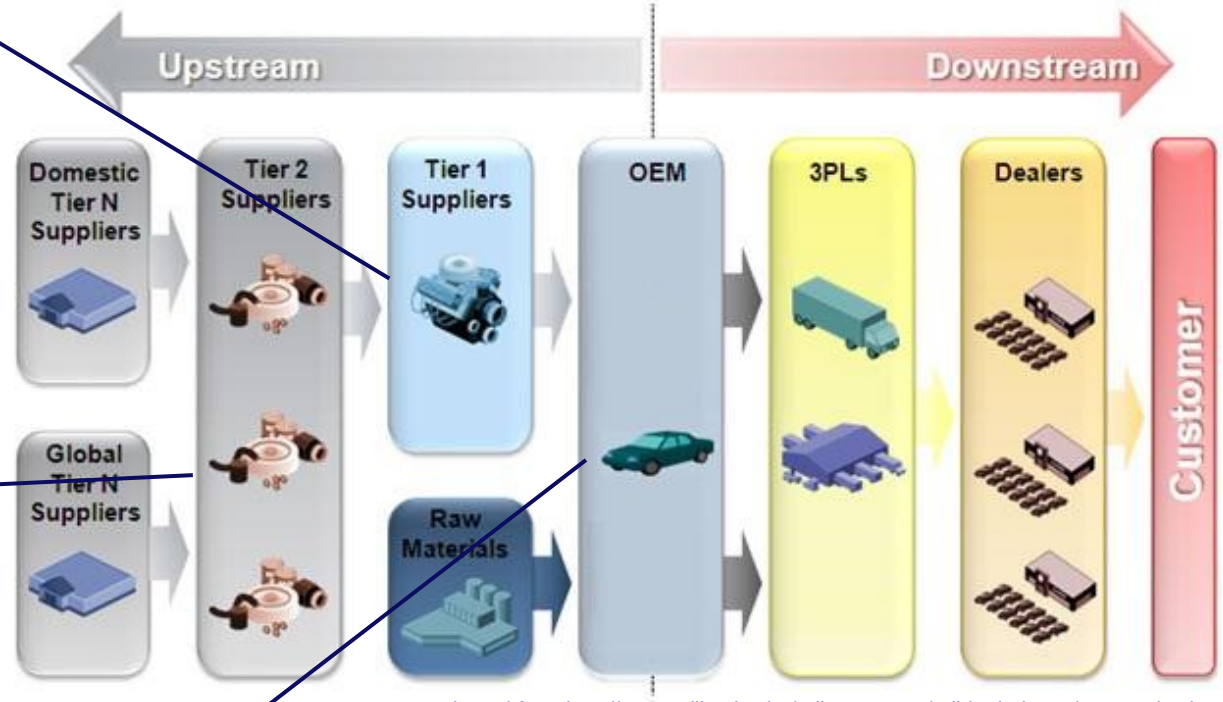
Faster time to market

Decreasing development cost

Automotive supply chain software integration





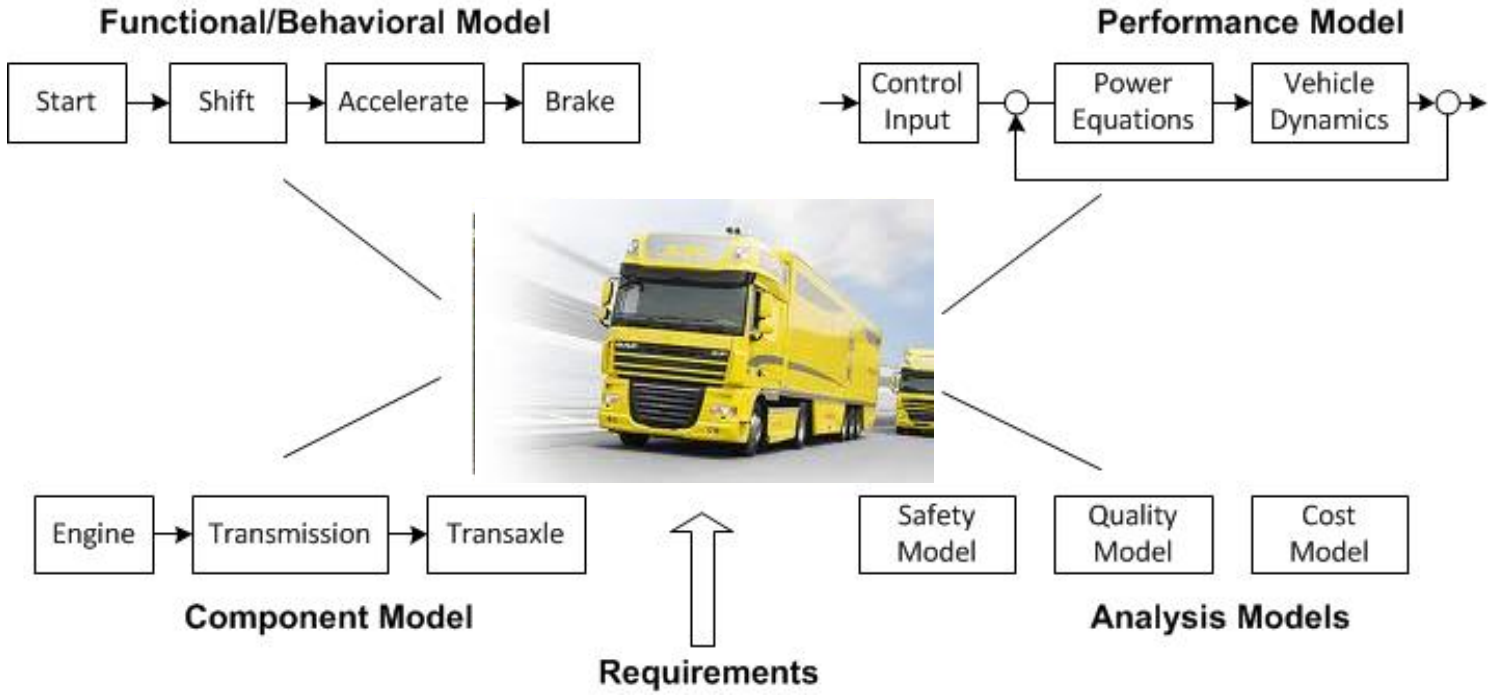


Adapted from http://www.edibasics.hu/edi-resources/edi-by-industry/automotive.htm

Document-centric:
- Manual
- Error prone
- Costly to change

Architecture-driven:
- (Partially) Automated
- Early detection of errors
- Less effort/cost to change

Automotive architecture modeling



- **Top-down system development i.o. bottom up**
- **Separation of concerns in different architectural models/views**
- **Model-driven i.o. document-centric approach**
- **Improved design quality by detecting errors early**
- **…**

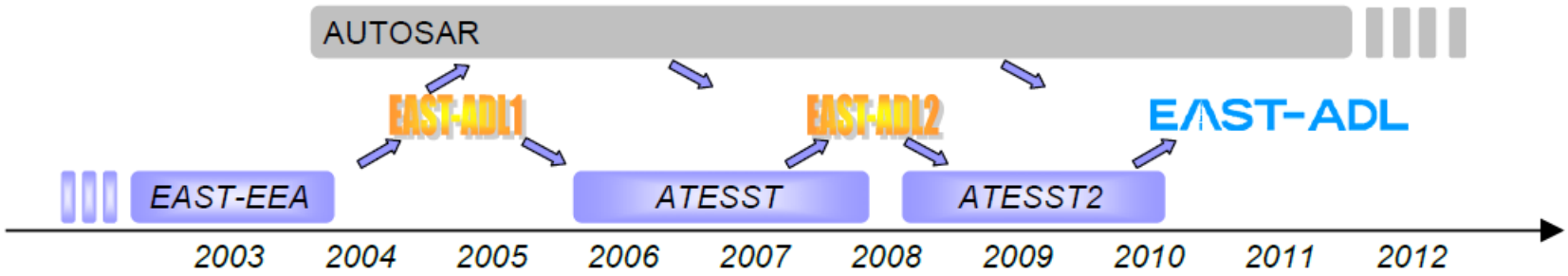
Automotive companies and ADLs

- Automotive Modeling Language (**AML**)
- COmponent Language (**COLA**)

- **EAST-ADL**
- Timing Augmented Description Language (**TADL**)

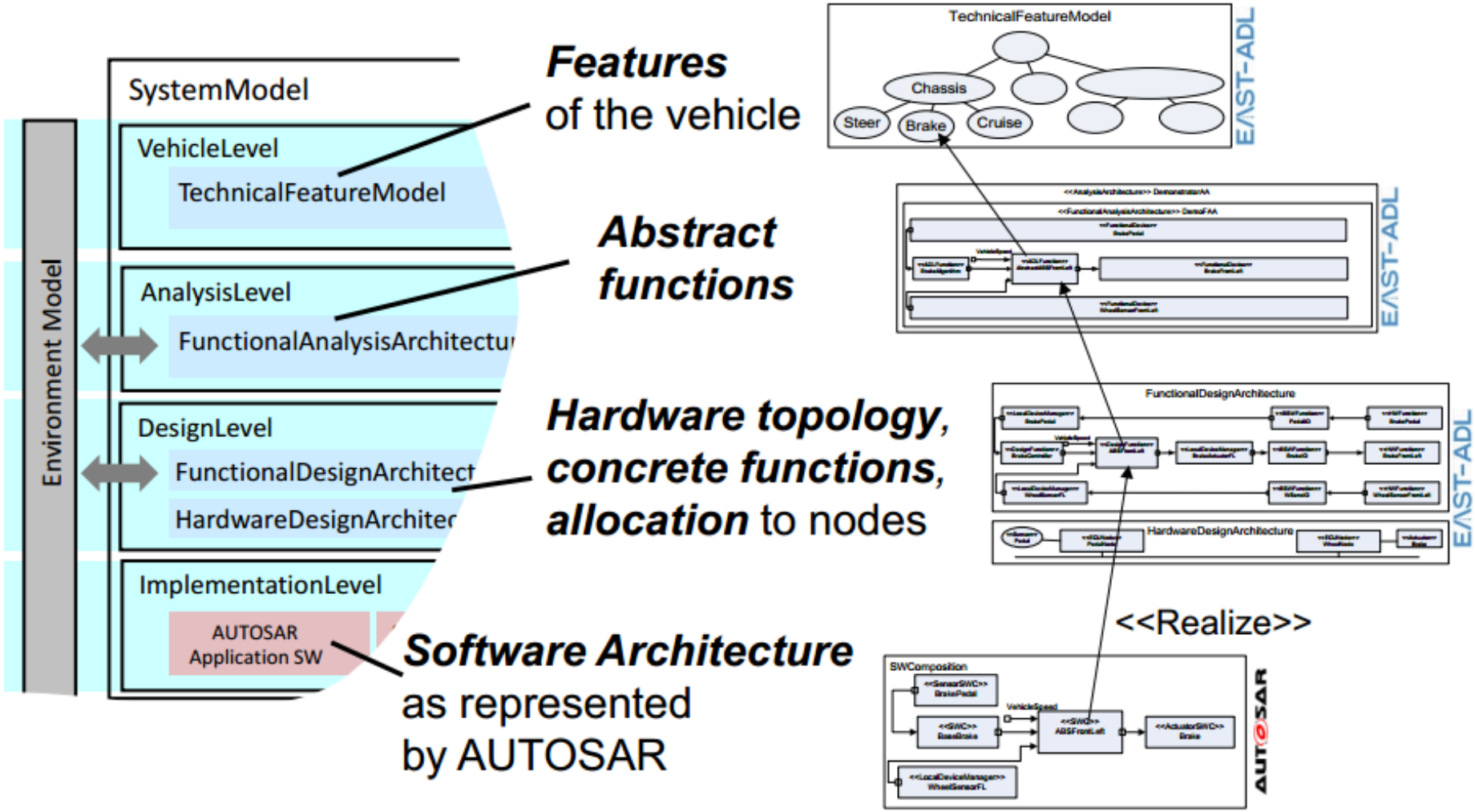
- The ICT MAENAD project **EAST-ADL2**

EAST-ADL



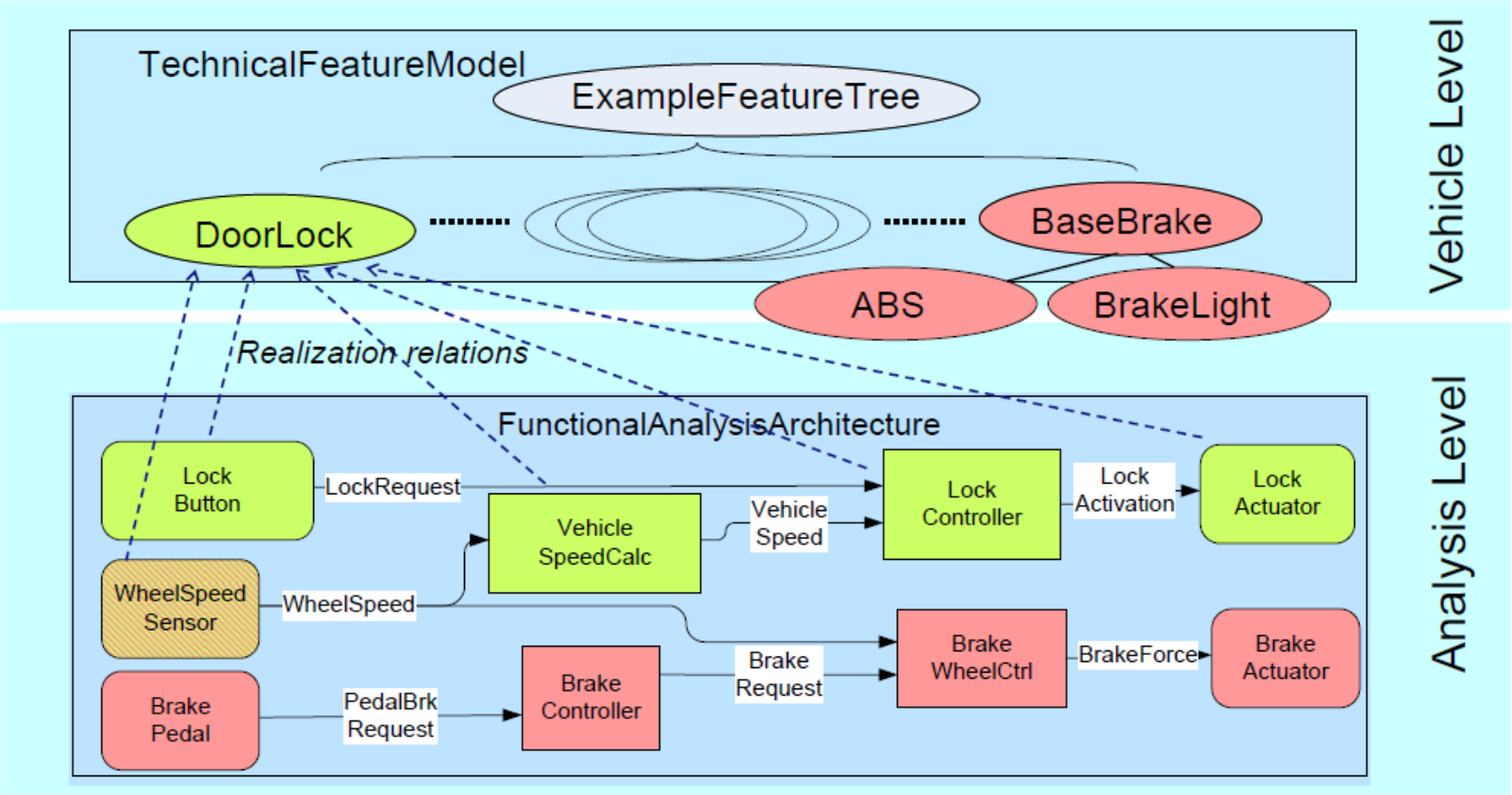
- EAST-ADL
  - Advancing Traffic Efficiency and Safety through Software Technology 2 (ATESST) project
  - Refined EAST-ADL2 language, profile, methodology, tools
  - It provides means to represent the embedded system in several abstraction levels.
  - Main source: http://www.east-adl.info/

EAST-ADL and AUTOSAR

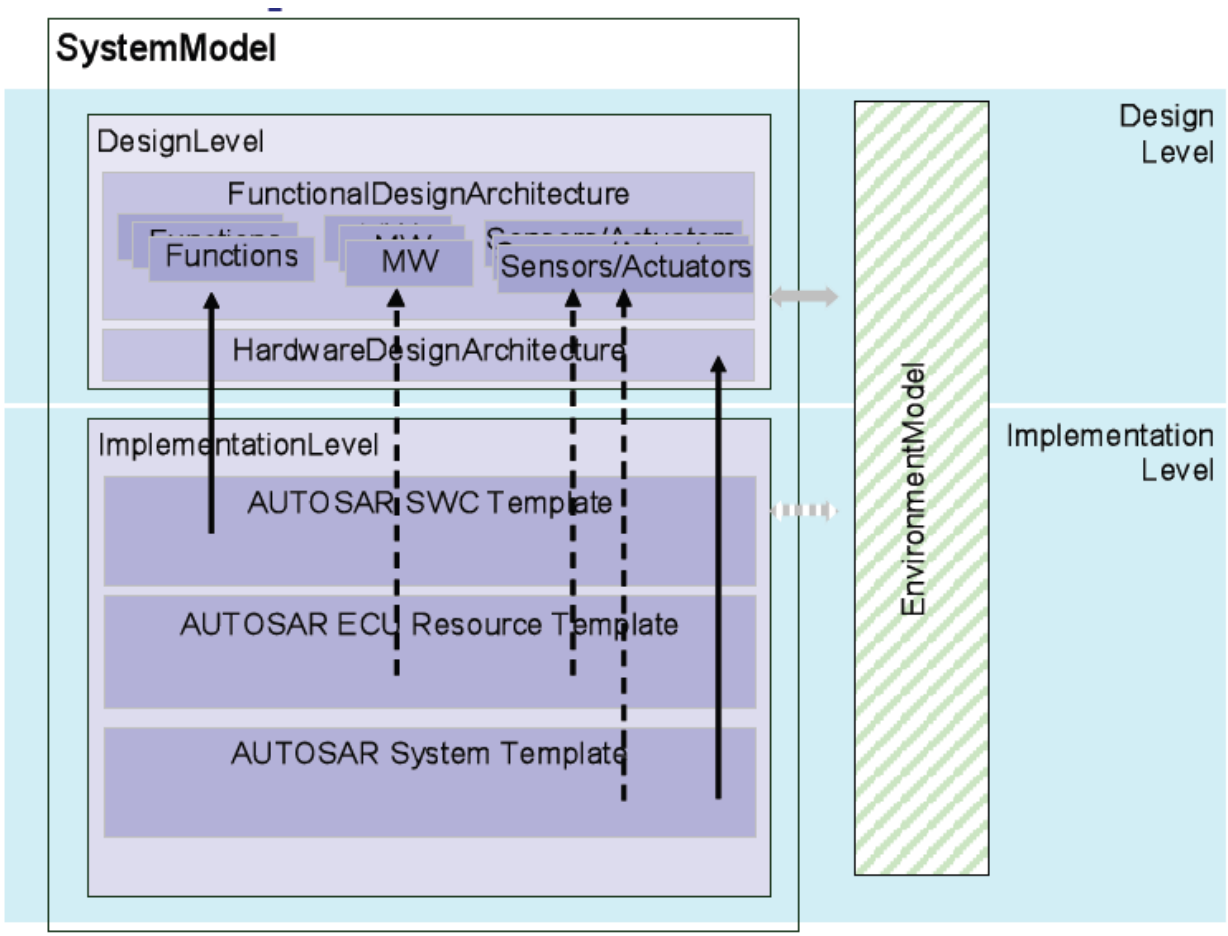


http://maenad.eu/

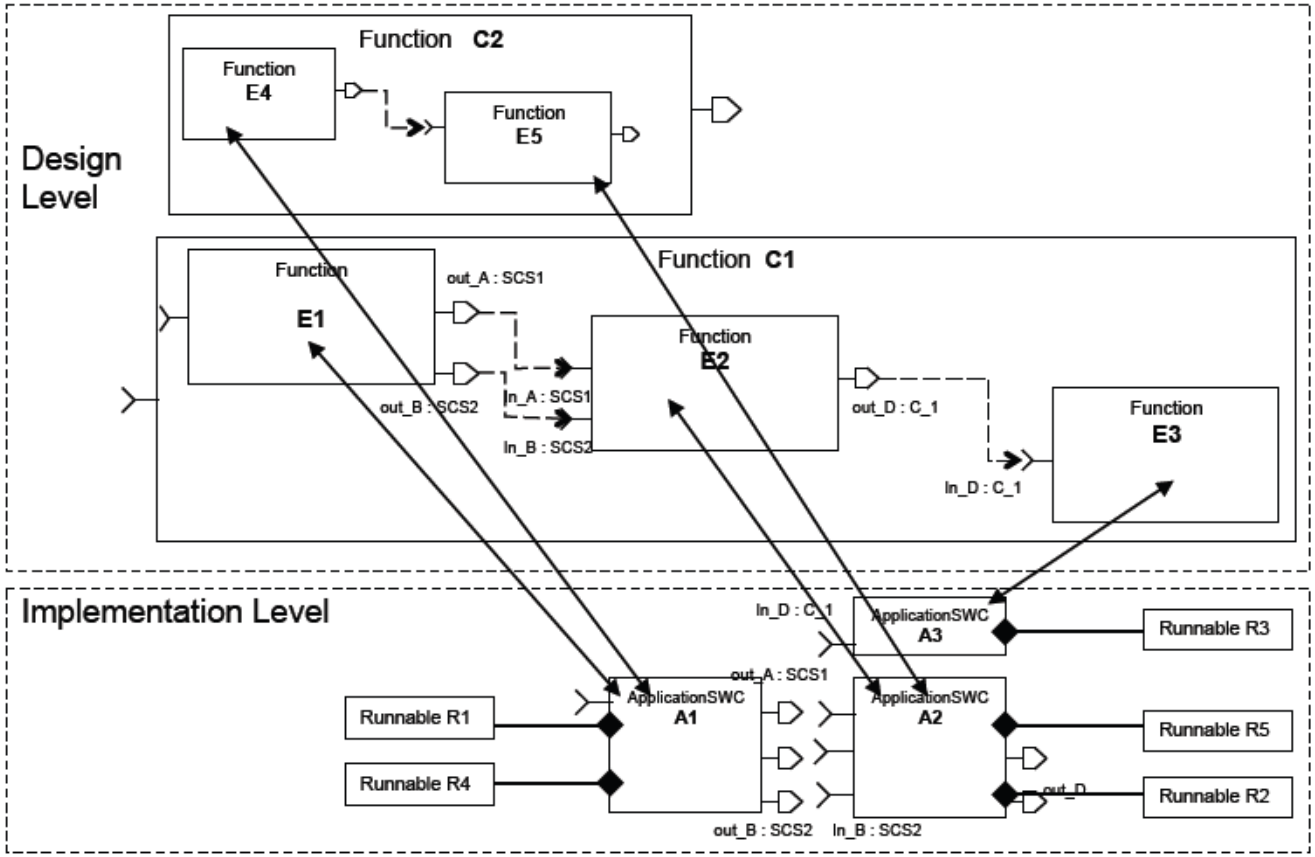
EAST-ADL Abstraction Levels



EAST-ADL Abstraction Levels

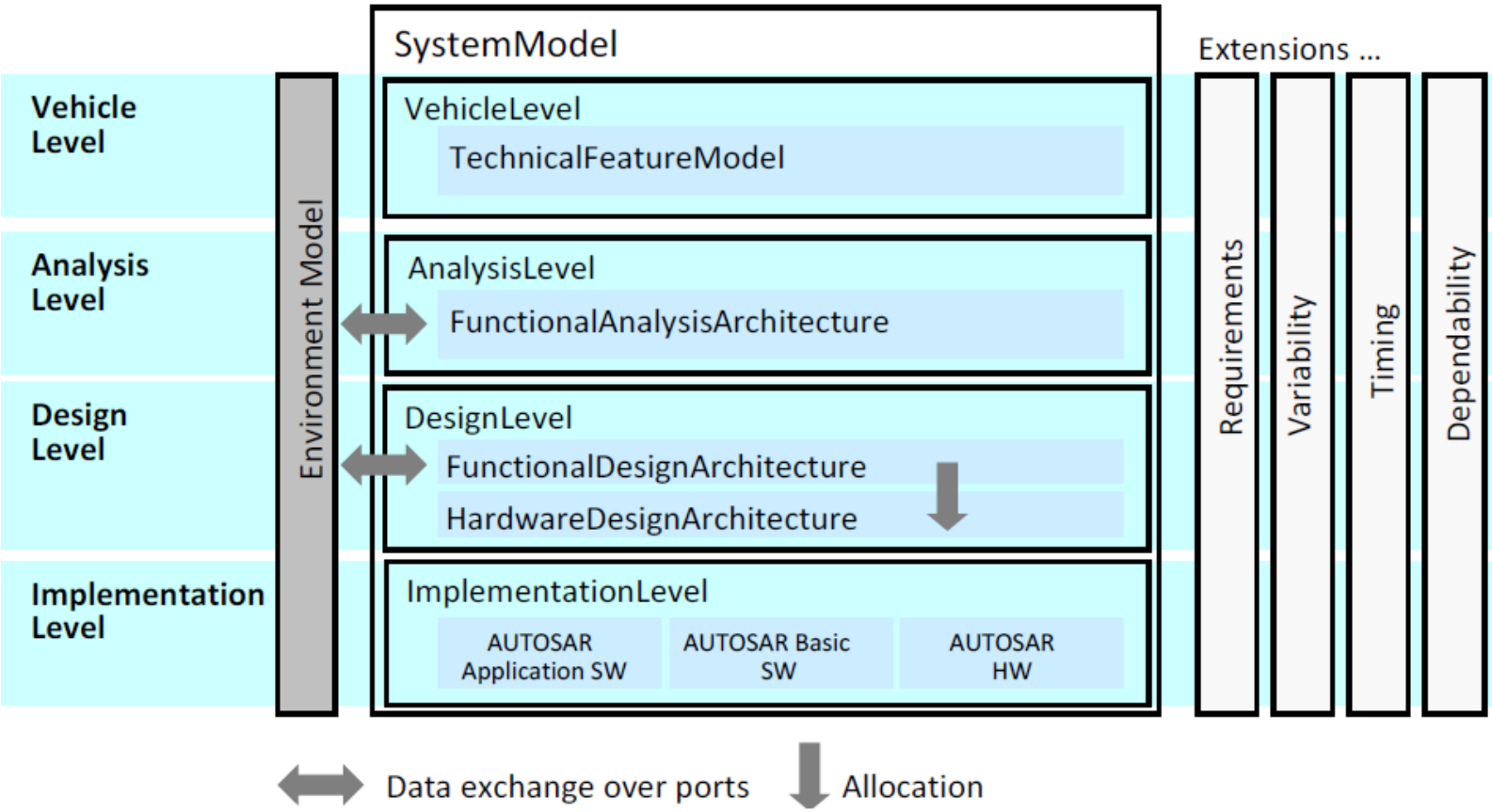


Example of function-to-component Mapping



n Function to 1 SW Component

EAST-ADL Metamodel Structure



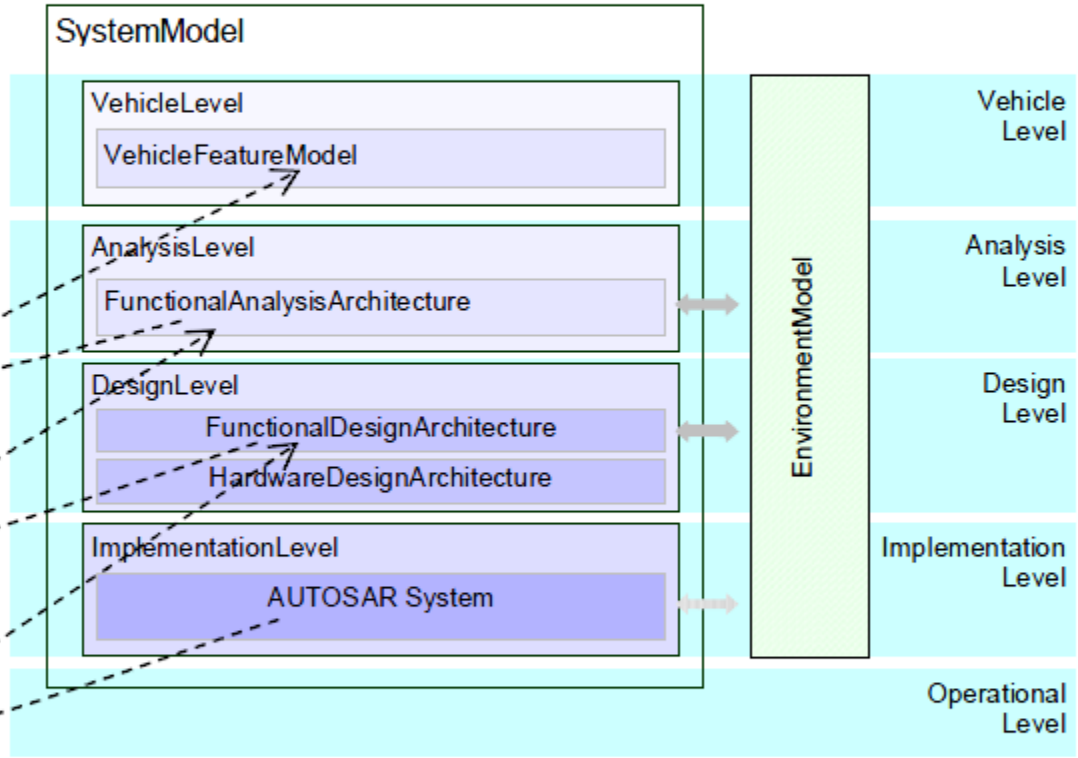
Traceability between abstraction levels

Realization relations identify which abstract element is realized by a more concrete entity

Functions on analysis level realizes features on vehicle level

Functions on design level realizes functions on analysis level

SW components or runnables on implementation level realizes functions on design level



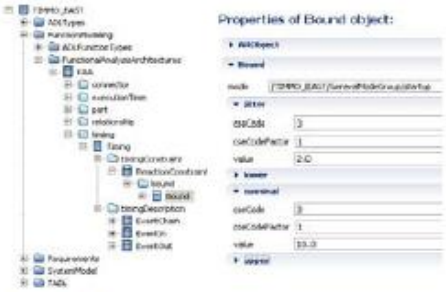
EAST-ADL Tooling

UML-based Tooling

- CEA Papyrus
  - Integrated Eclipse application with 5 ATESST plugins
- MagicDraw UML
- …

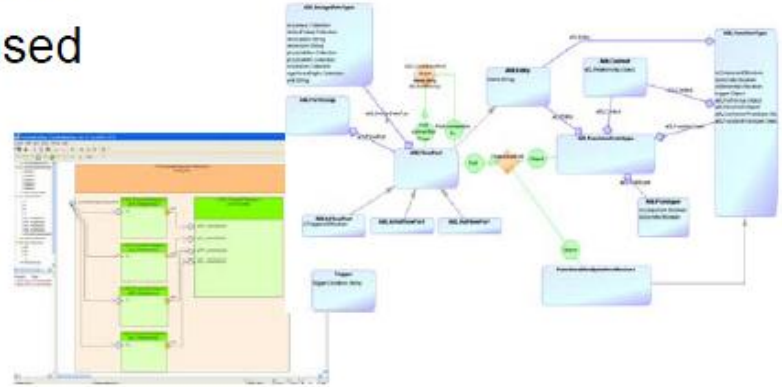
DSL AUTOSAR Tooling

- MentorGraphics VSA



DSL Generic Tooling

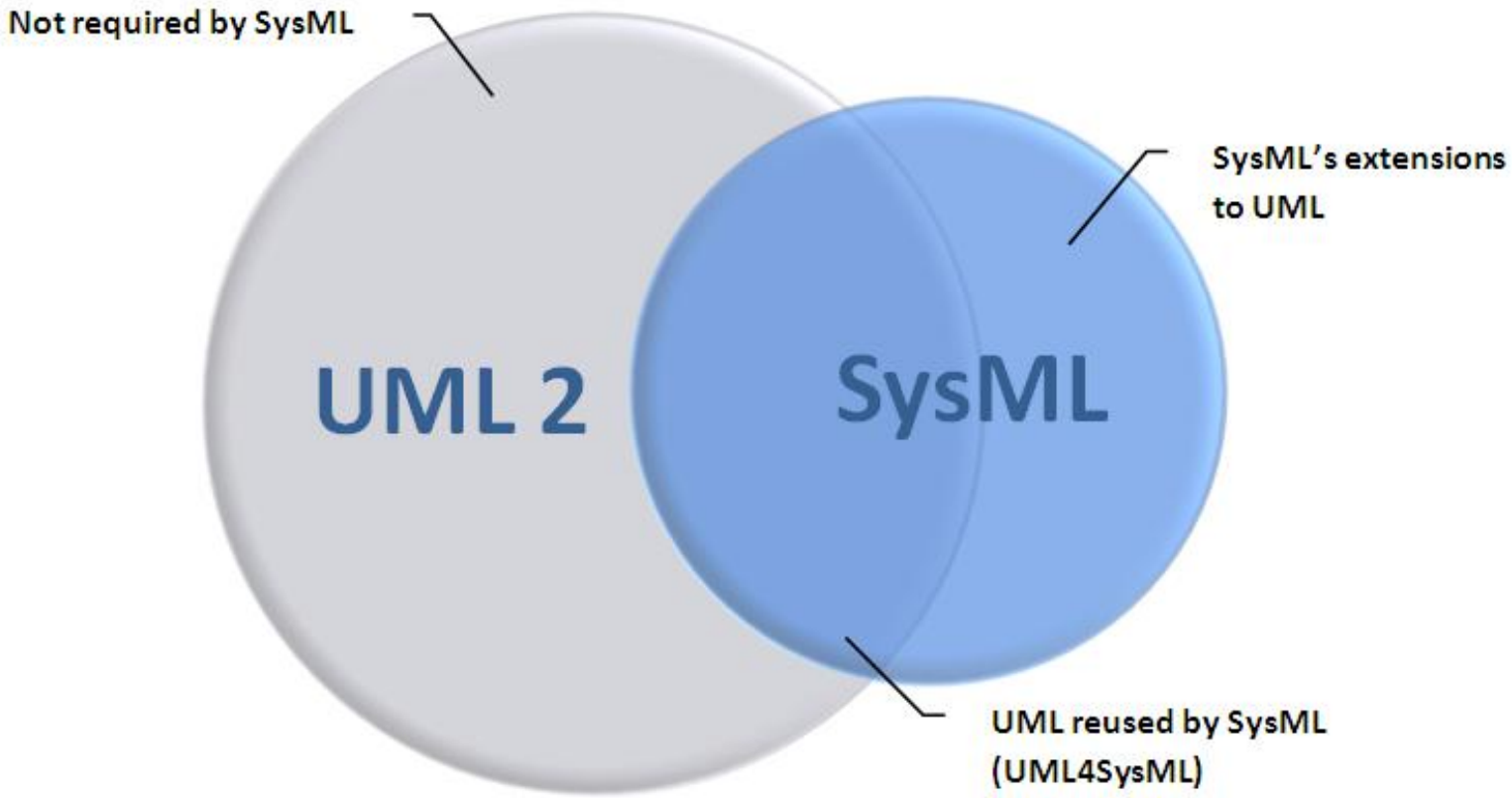
- MetaEdit+
- TopCased



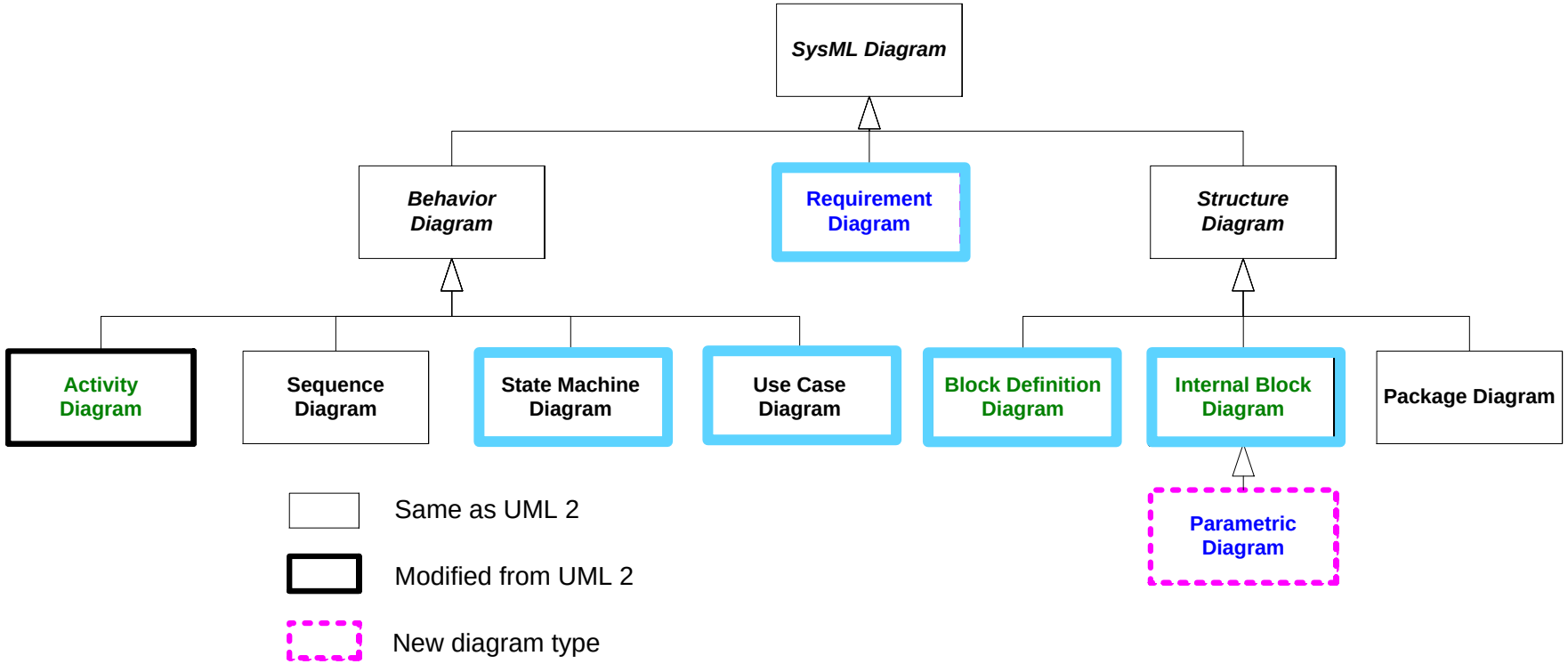
EAST-ADL Summary

- Defines several abstraction levels and mapping between them
- Extensions to traditional ADLs:
  - Requirements
  - Variability
  - Timing
  - Dependability
  - Safety (alignment with ISO26262)
  - Environment modeling
- Not well applied yet in automotive industry

SysML and UML

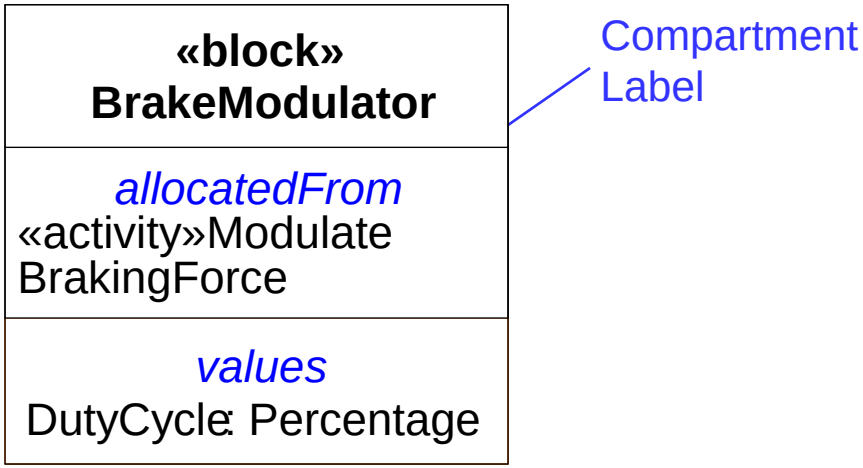


SysML Diagram Taxonomy



Blocks are Basic Structural Elements

- **Provides a unifying concept to describe the structure of an element or system**
  - **System**
  - **Hardware**
  - **Software**
  - **Data**
  - **Procedure**
  - **Facility**
  - **Person**



- **Multiple standard compartments can describe the block characteristics**
  - **Properties (parts, references, values, ports)**
  - **Operations**
  - **Constraints**
  - **Allocations from/to other model elements (e.g. activities)**
  - **Requirements the block satisfies**
  - **User defined compartments**

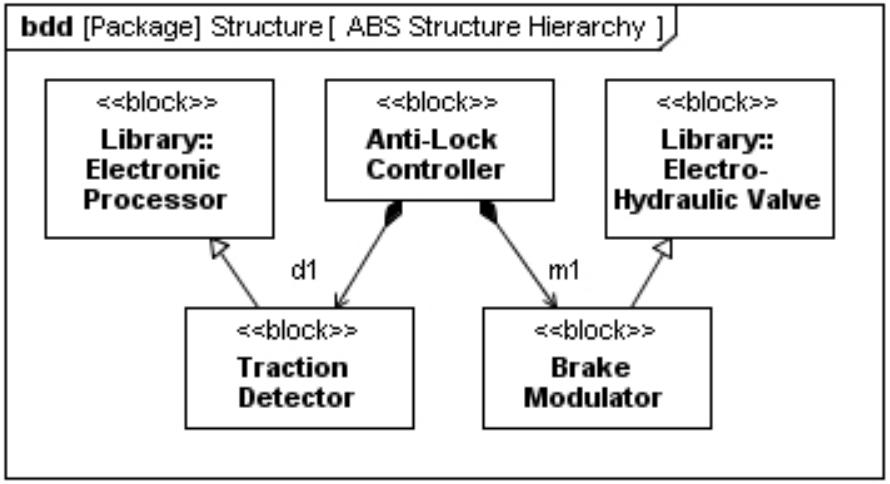
Using Blocks

- **Based on UML Class from UML Composite Structure**
  - **Supports unique features (e.g., flow ports, value properties)**
- **Block definition diagram describes the relationship among blocks (e.g., composition, association, specialization)**
- **Internal block diagram describes the internal structure of a block in terms of its properties and connectors**
- **Behavior can be allocated to blocks**

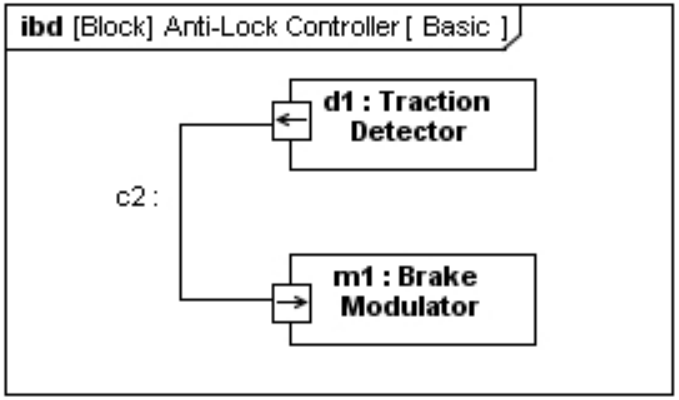
Blocks Used to Specify Hierarchies and Interconnection

Block Definition vs. Usage

Block Definition Diagram



Internal Block Diagram



Definition

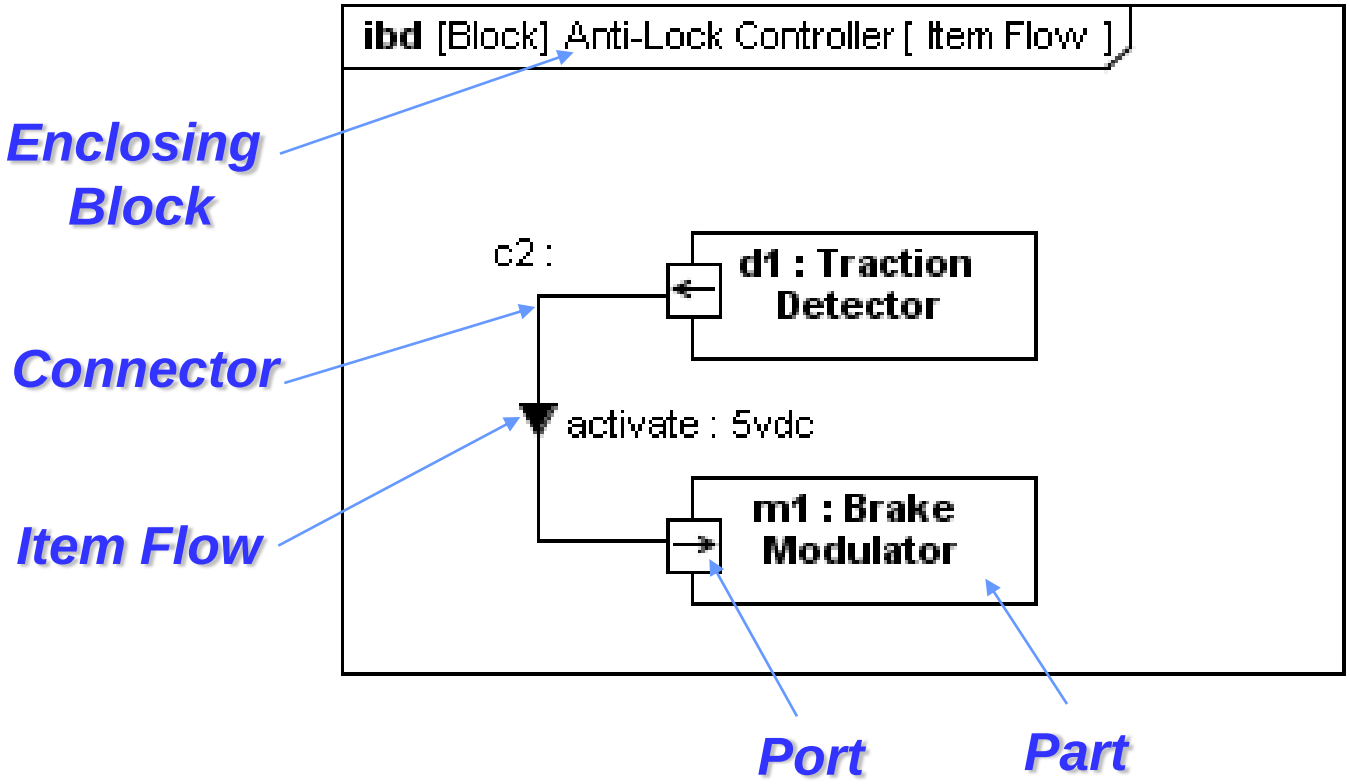
- **Block is a definition/type**
- **Captures properties, etc.**
- **Reused in multiple contexts**

Usage

- Part is the usage of a block in the context of a composing block
- Also known as a role

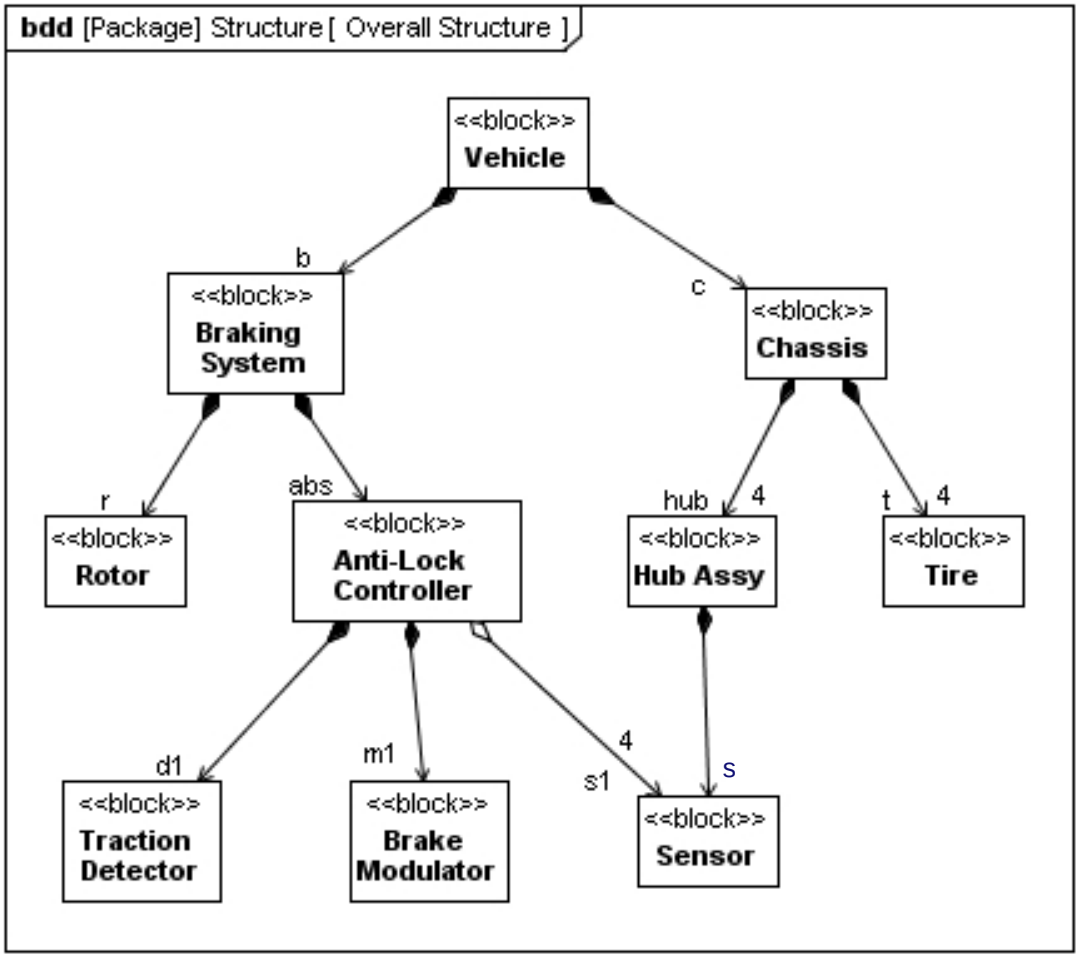
Internal Block Diagram (ibd)

Blocks, Parts, Ports, Connectors & Flows

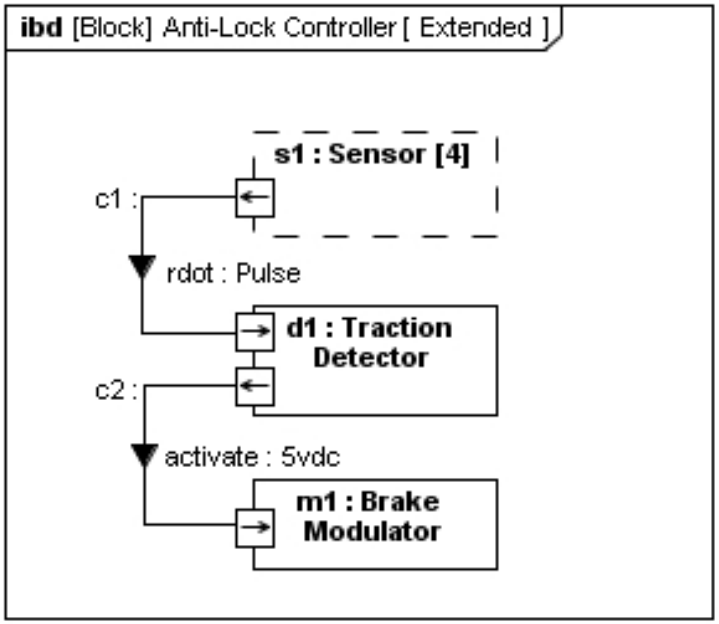


Internal Block Diagram Specifies Interconnection of Parts

Reference Property Explained



- **S1 is a reference part***
- **Shown in dashed outline box**



*Actual name is reference property

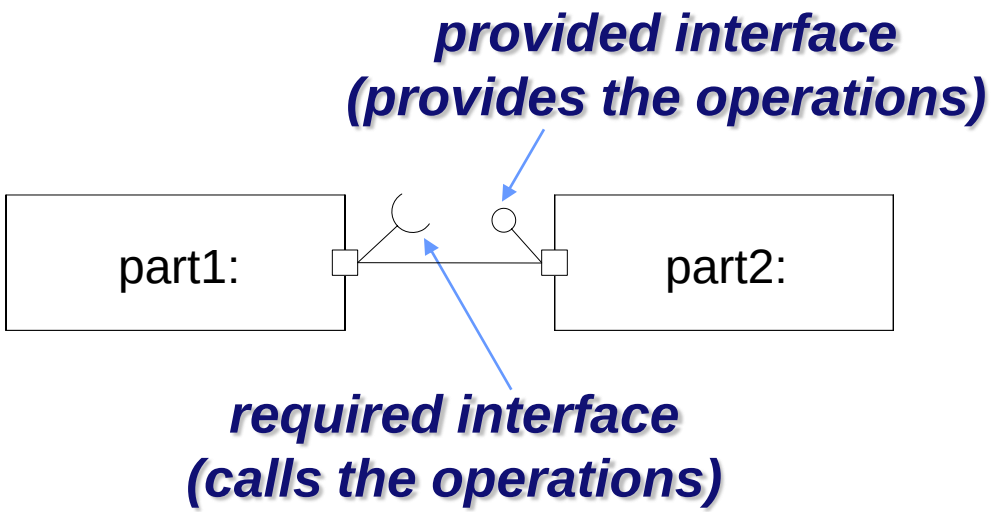
SysML Ports

- Specifies interaction points on blocks and parts
  - Integrates behavior with structure
  - portName:TypeName
- Kinds of ports
  - Standard (UML) Port
    - Specifies a set of required or provided operations and/or signals
    - Typed by a UML interface
  - Flow Port
    - Specifies what can flow in or out of block/part
    - Typed by a block, value type, or flow specification
    - Atomic, non-atomic, and conjugate variations

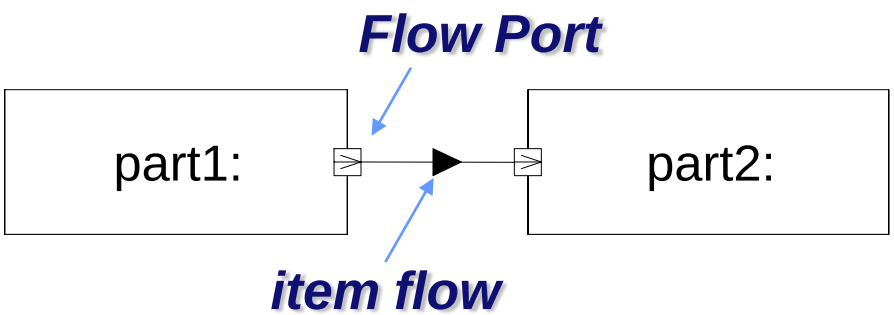
Standard Port and Flow Port
Support Different Interface Concepts

Port Notation

Standard Port



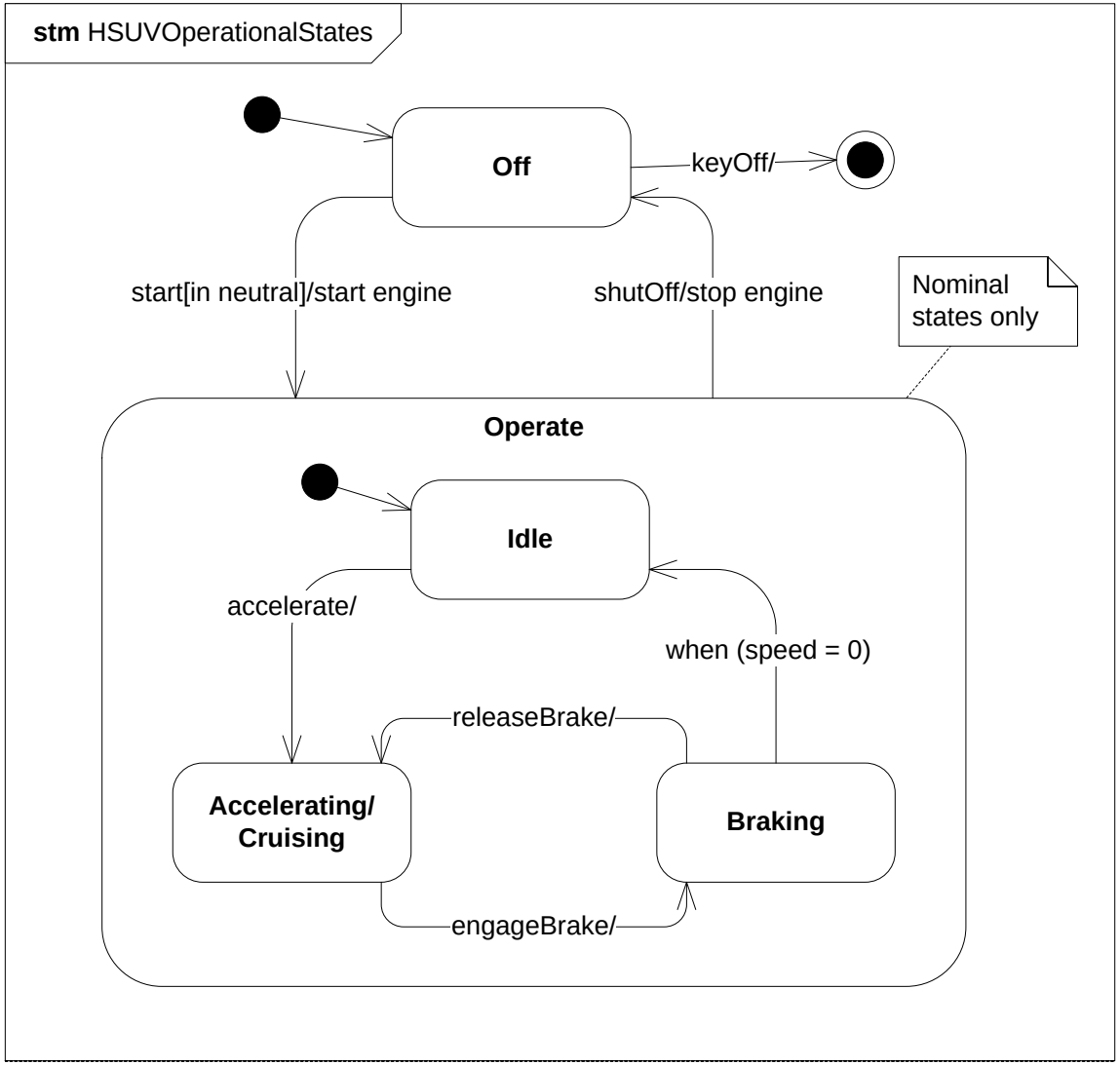
Flow Port



State Machines

- Typically used to represent the life cycle of a block
- Support event-based behavior (generally asynchronous)
  - Transition with trigger, guard, action
  - State with entry, exit, and do-activity
  - Can include nested sequential or concurrent states
  - Can send/receive signals to communicate between blocks during state transitions, etc.
- Event types
  - Change event
  - Time event
  - Signal event

Operational States (Drive)



Transition notation:
trigger[guard]/action

DEMO


PAUSE

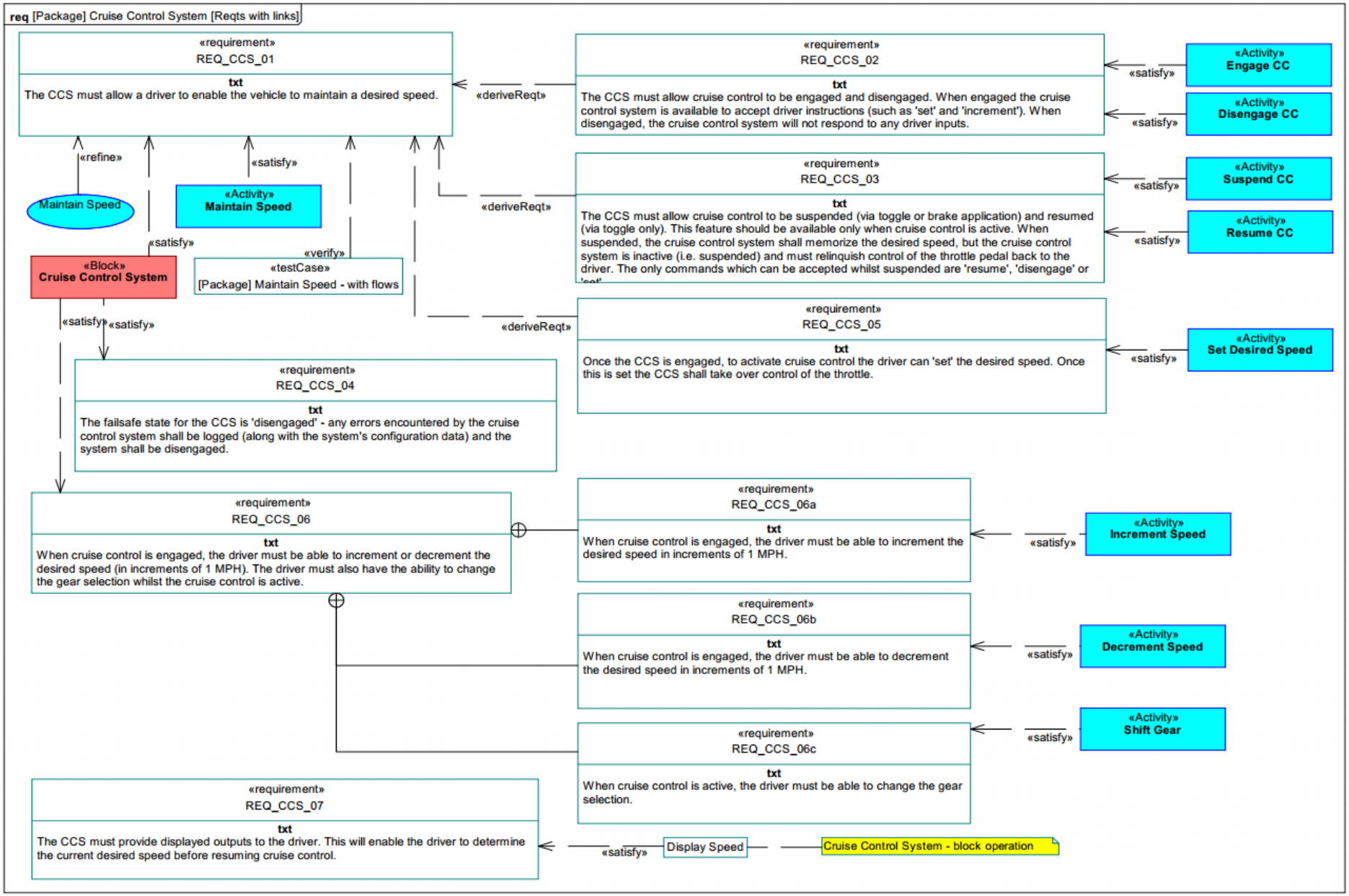
Adaptive Cruise Control (ACC) in SysML

Image: http://www.extremetech.com/

Modeling the ACC system for an E-truck with a top-down approach in SysML

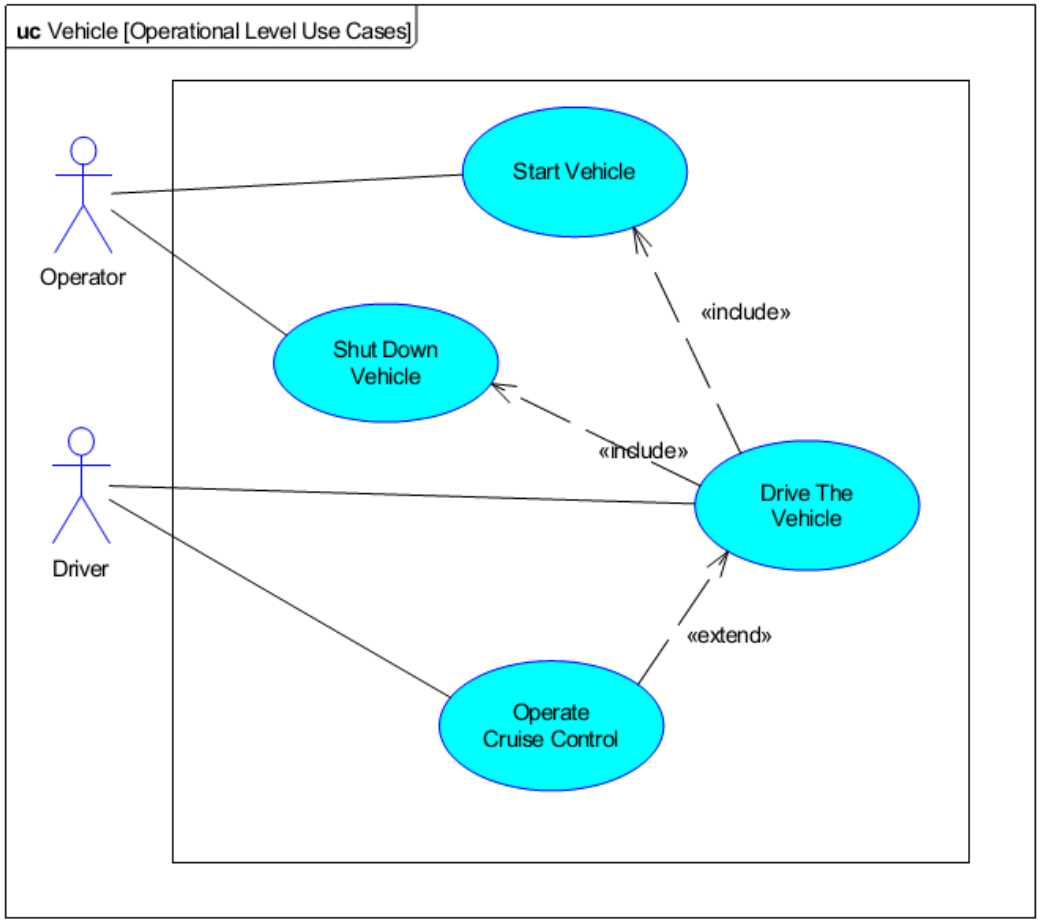
Technische Universiteit Eindhoven University of Technology

Requirements Diagram



Source: Artisan Software Tools

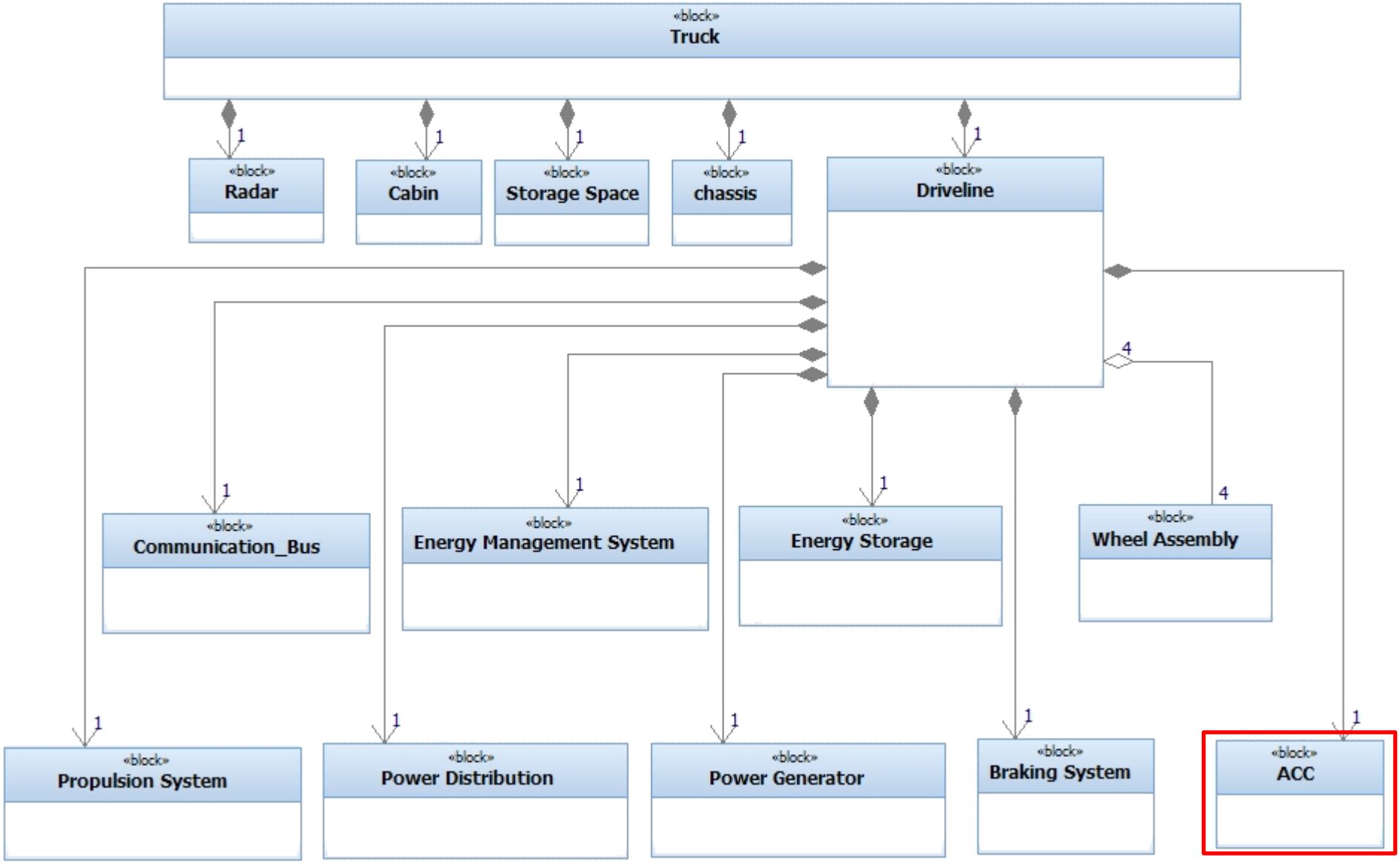
Use Case diagram



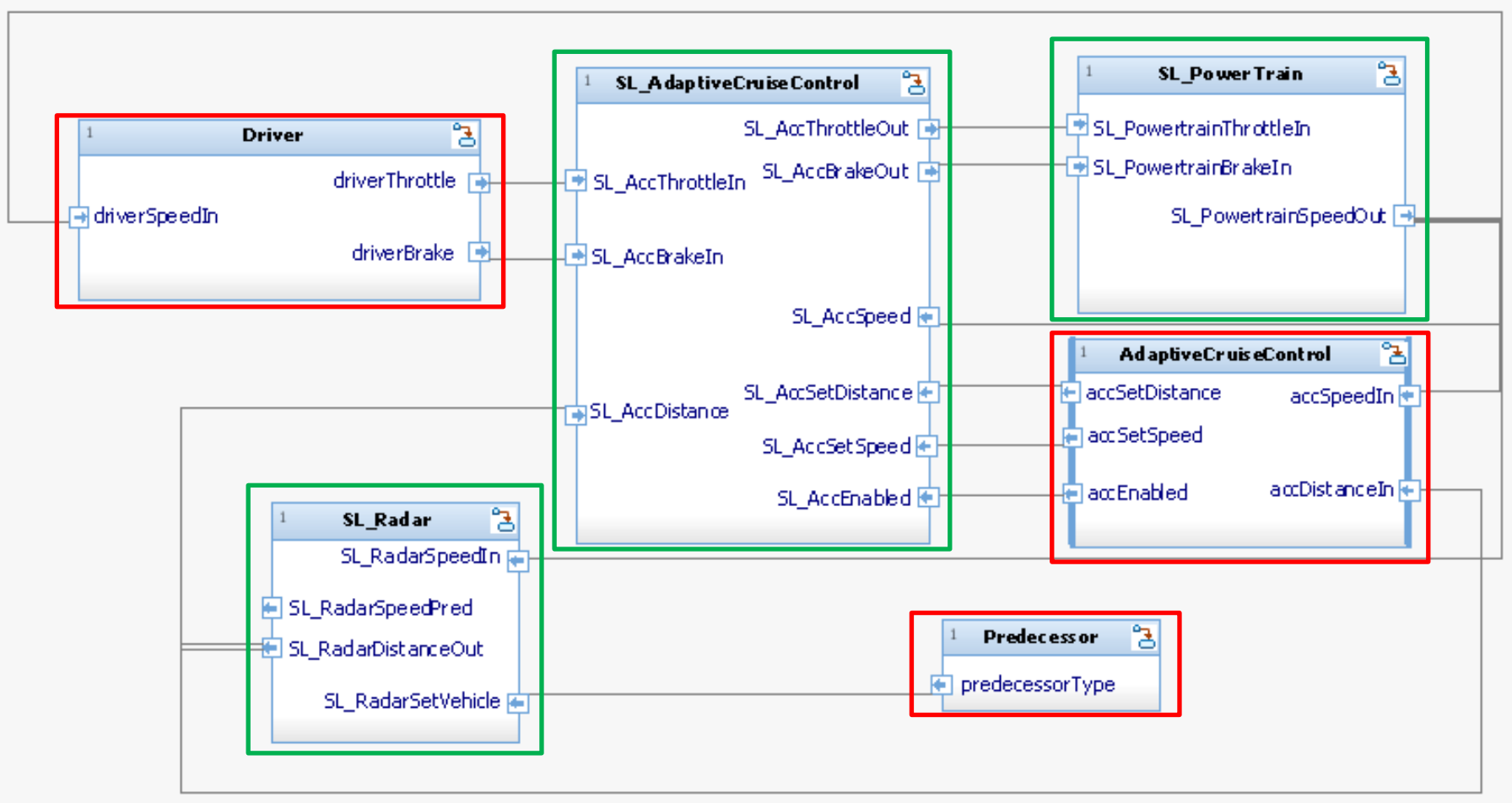
Source: Artisan Software Tools

- **Provides means for describing basic functionality in terms of usages of system by actors**
- **Generally elaborated via other behavioral representations to describe detailed scenarios**

System architecture

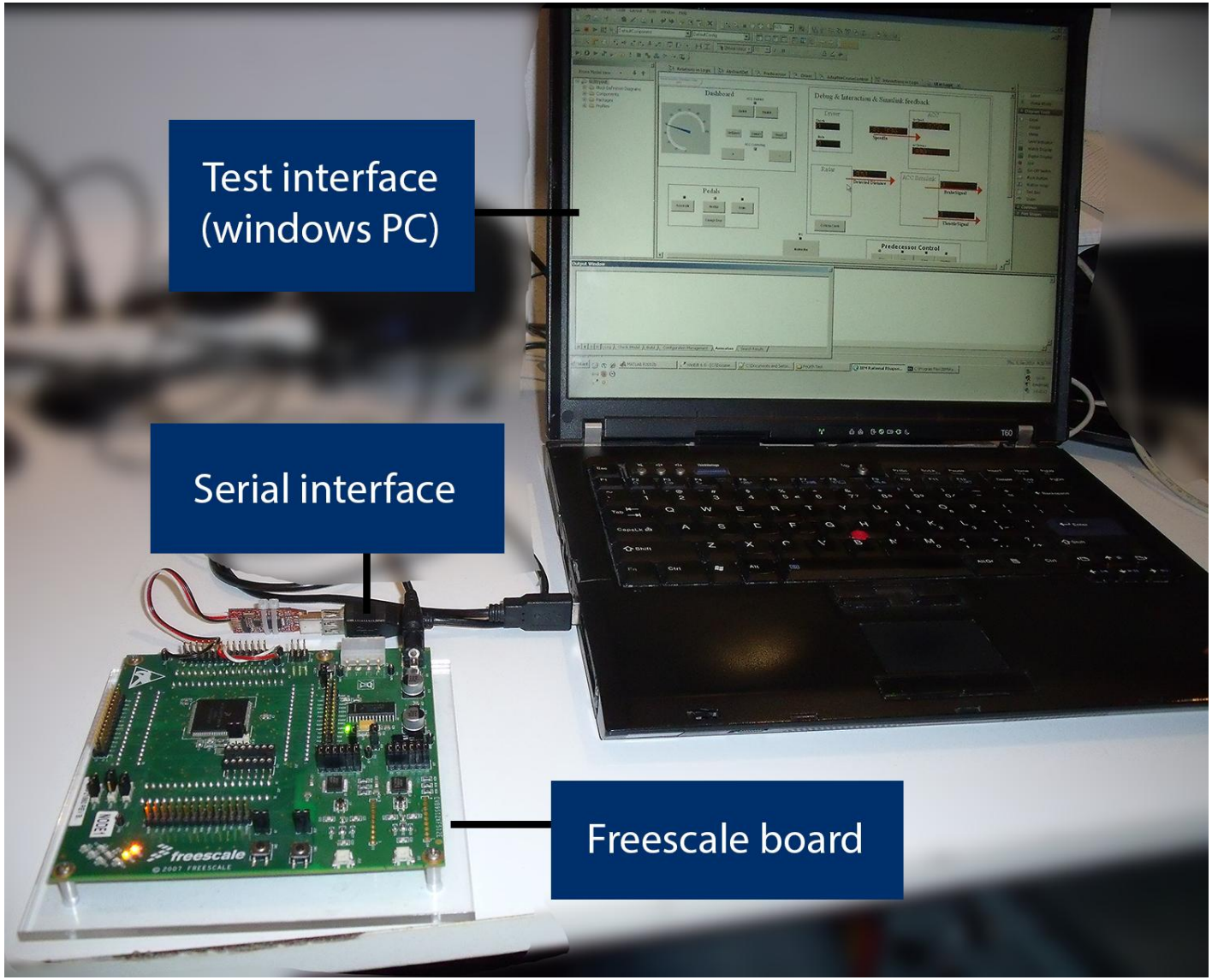


System integration



- **Software**
- **Hardware**

Running ACC_UI on Freescale board



SysML summary

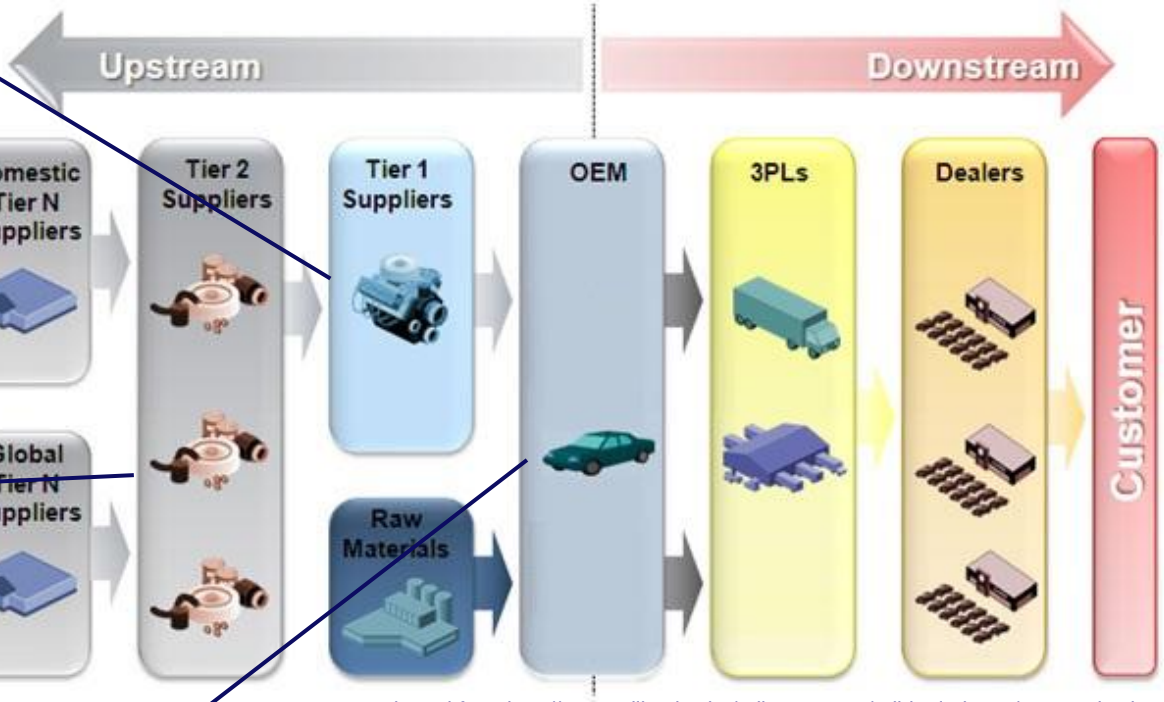
- SysML provides a general purpose modeling language to support specification, analysis, design and verification of complex systems
  - Subset of UML 2 with extensions
  - 4 Pillars of SysML include modeling of requirements, behavior, structure, and parametrics
- Intended to improve communications, tool interoperability, and design quality
- Multiple tools available
  - IBM –Rhapsody
  - Sparx Systems -Enterprise Architect
  - Atego –Artisan Studio etc.

Automotive supply chain software integration

Electronic Control Unit (ECU)

ECU

ECU

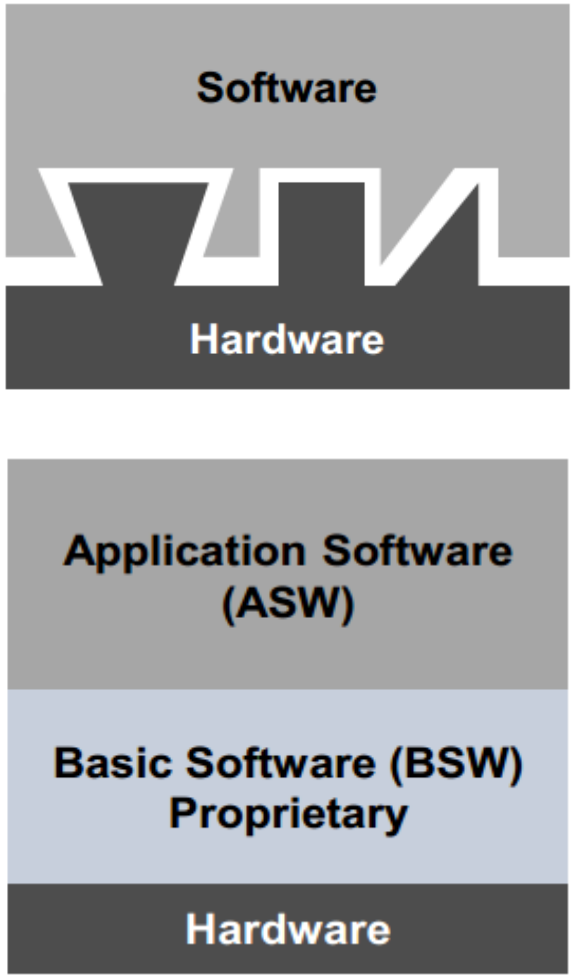


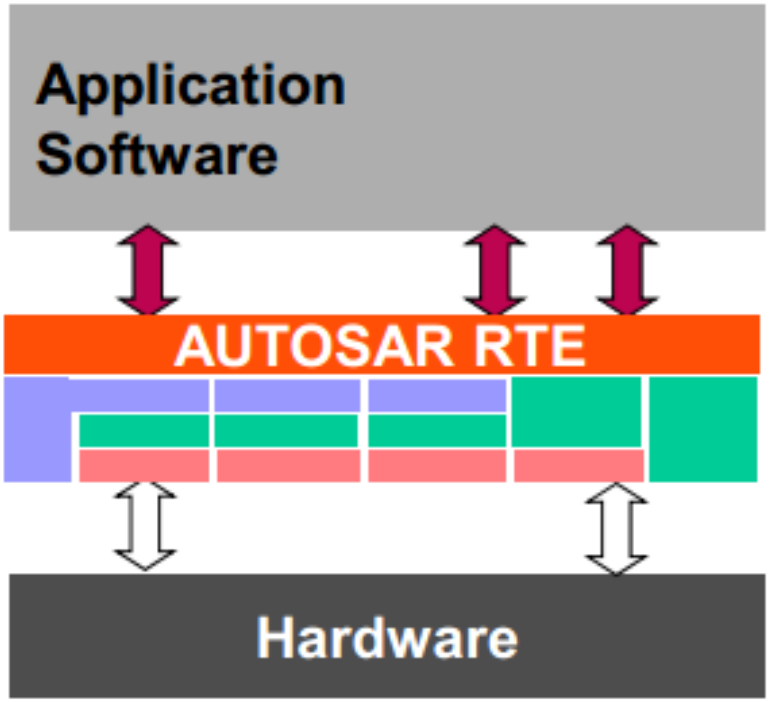
Adapted from http://www.edibasics.hu/edi-resources/edi-by-industry/automotive.htm

- Hardware dependent SW
- Not efficient software reuse and exchange
- Costly integration

AUTOSAR (AUTomotive Open System Architecture)

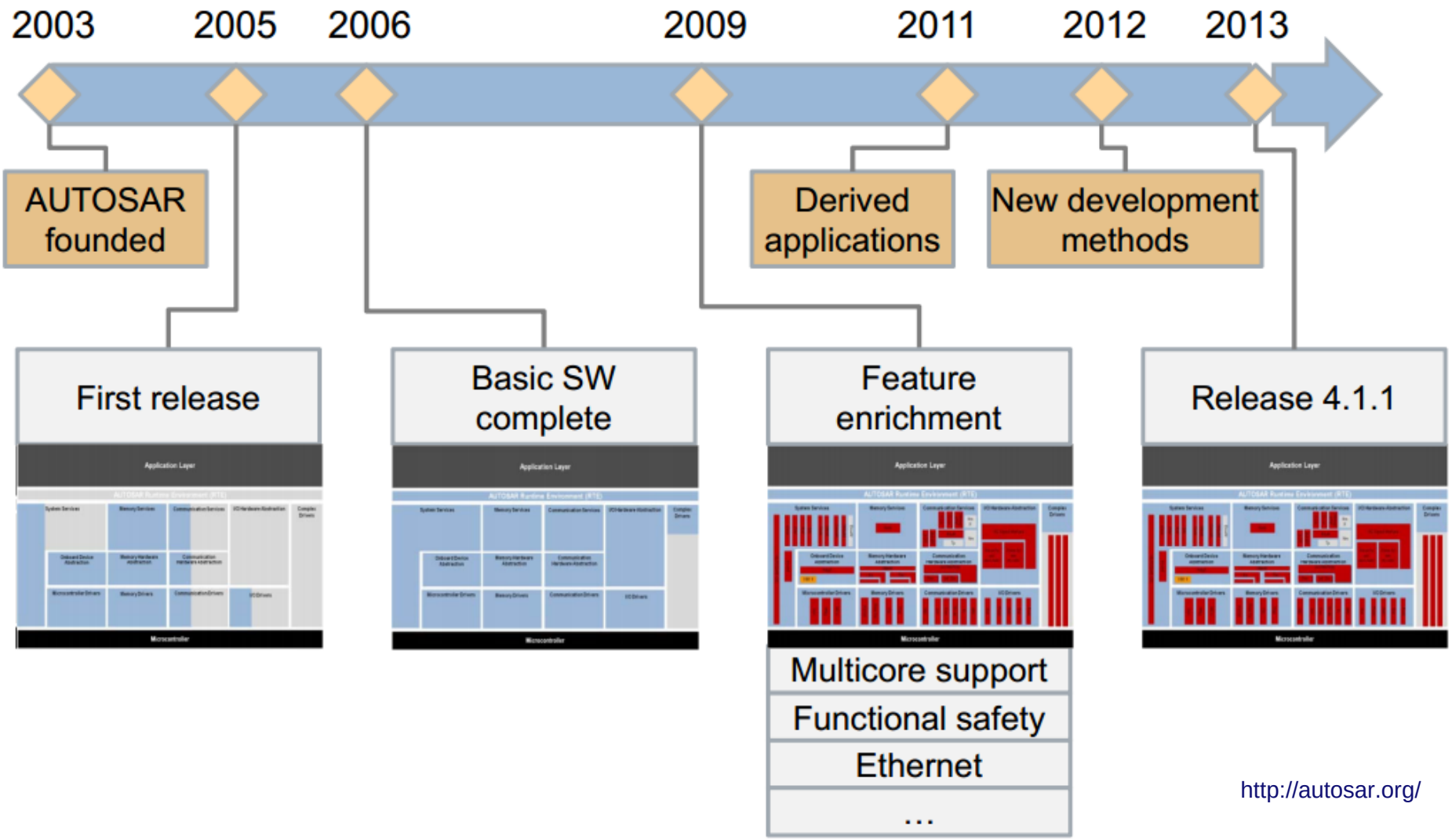
- An **open** and **standardized** automotive software architecture





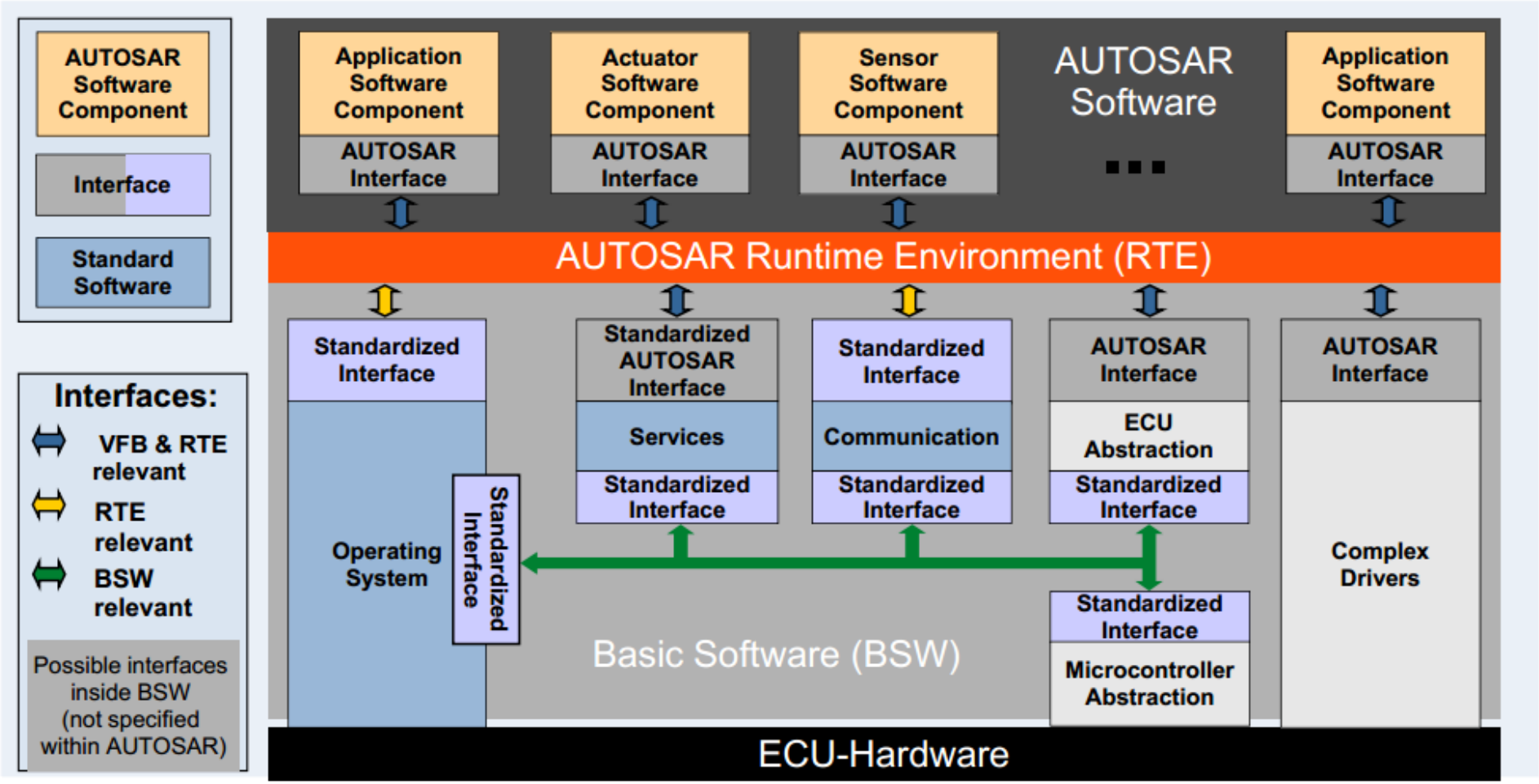
- **Architecture**
- **Methodology**
- **Application Interfaces**

AUTOSAR Milestones



http://autosar.org/

AUTOSAR Layered Architecture



http://autosar.org/

AUTOSAR Methodology

Virtual Integration

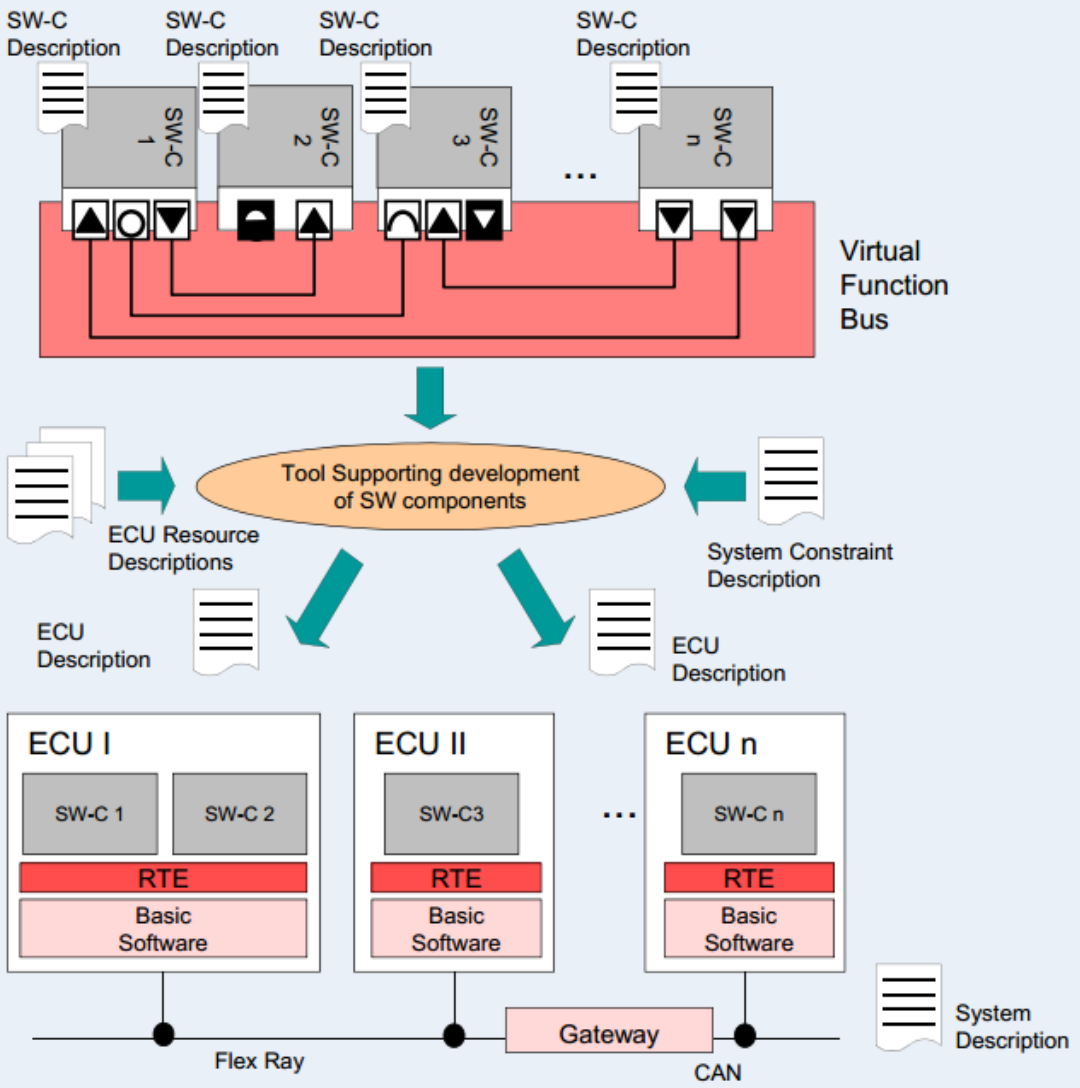
Virtual Functional Bus - Independent of hardware

Introduction of HW Attributes

Holistic view of the entire system, both software and hardware

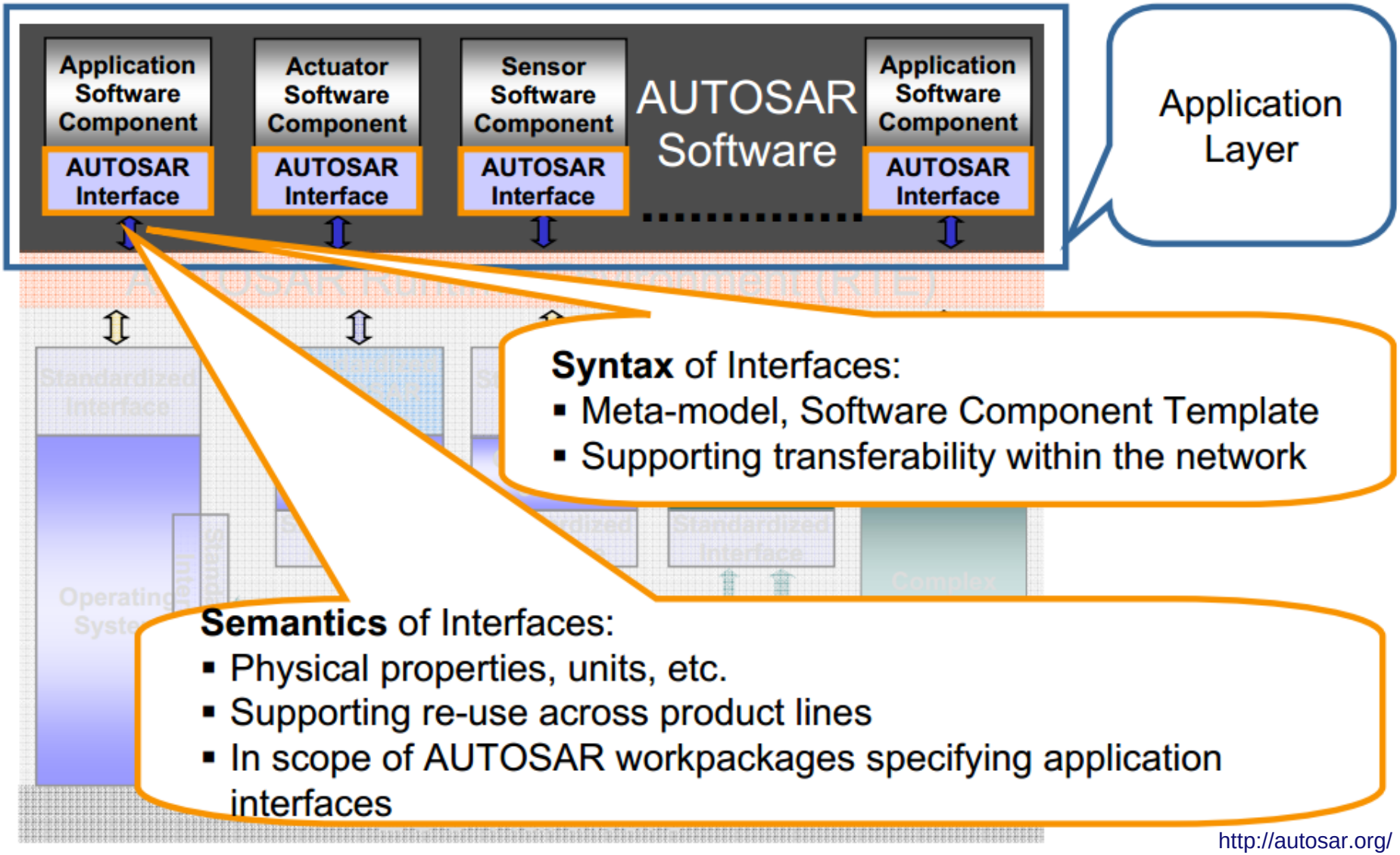
ECU Configuration

Separation of system into its ECUs with a common SW platform (infrastructure).



http://autosar.org/

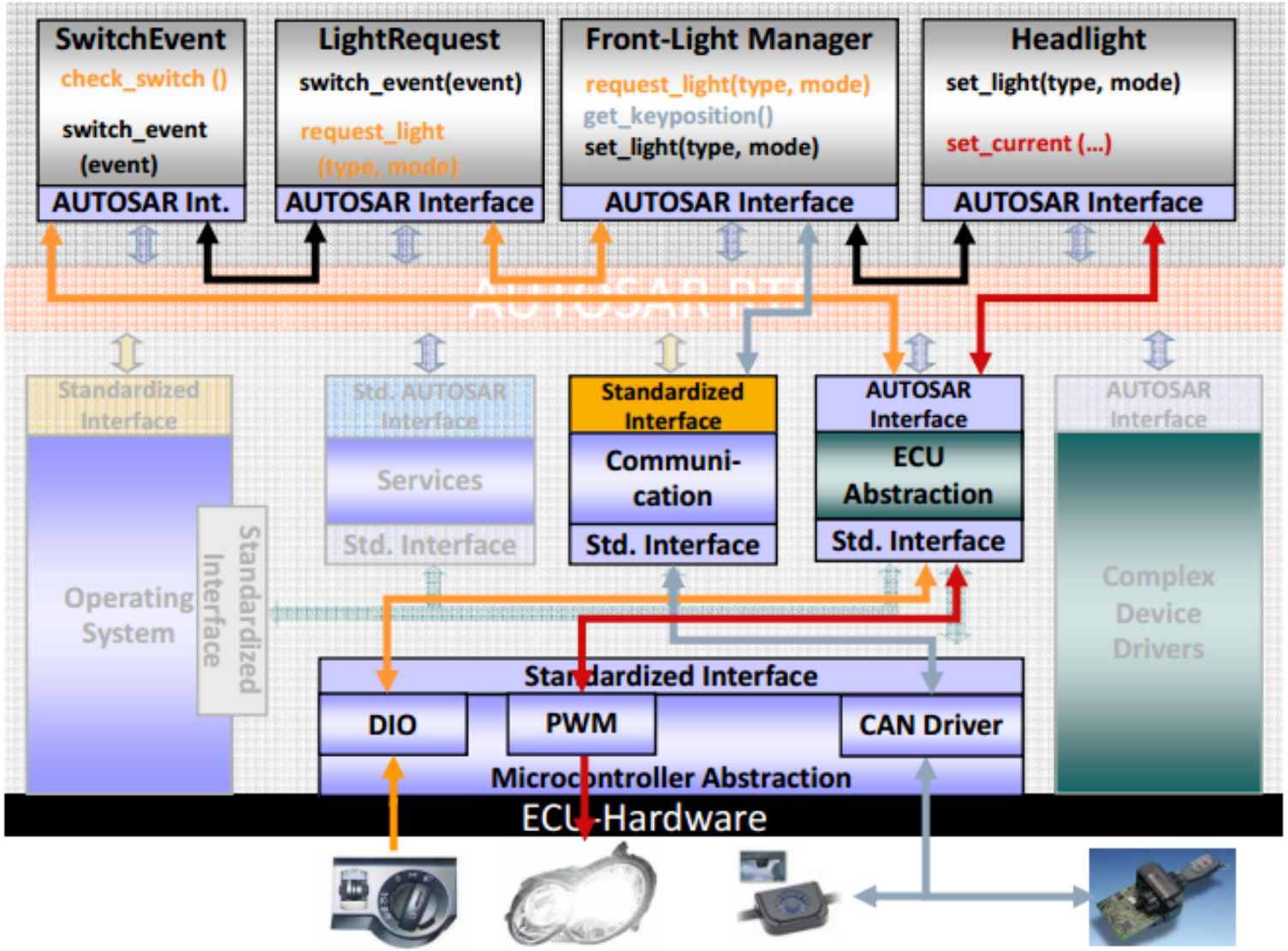
AUTOSAR Application Interface



http://autosar.org/

AUTOSAR Use Case

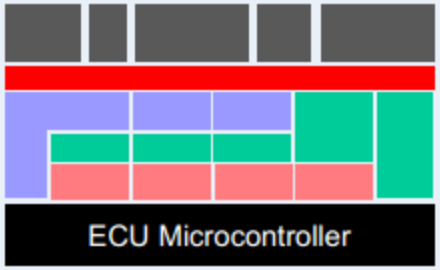
Use case 'Front-Light Management' in AUTOSAR



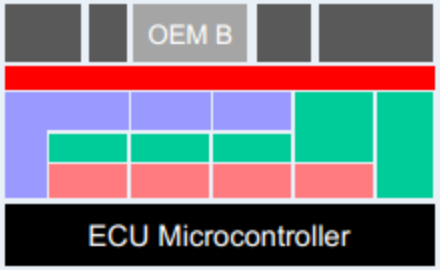
http://autosar.org/

AUTOSAR Benefits

Fully AUTOSAR compliant ECU

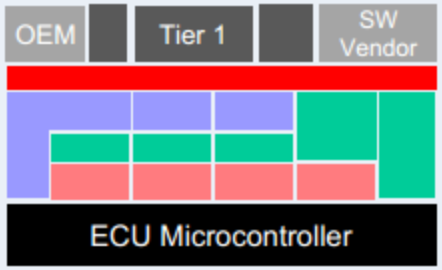


Scenario A



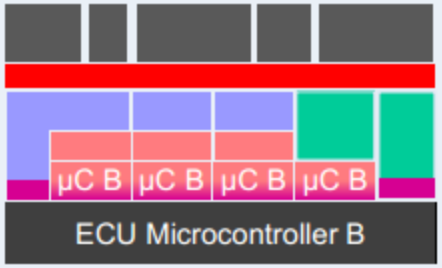
The supplier provides the ECU to a different OEM.

Scenario B



Integration of features, delivered from different sources.

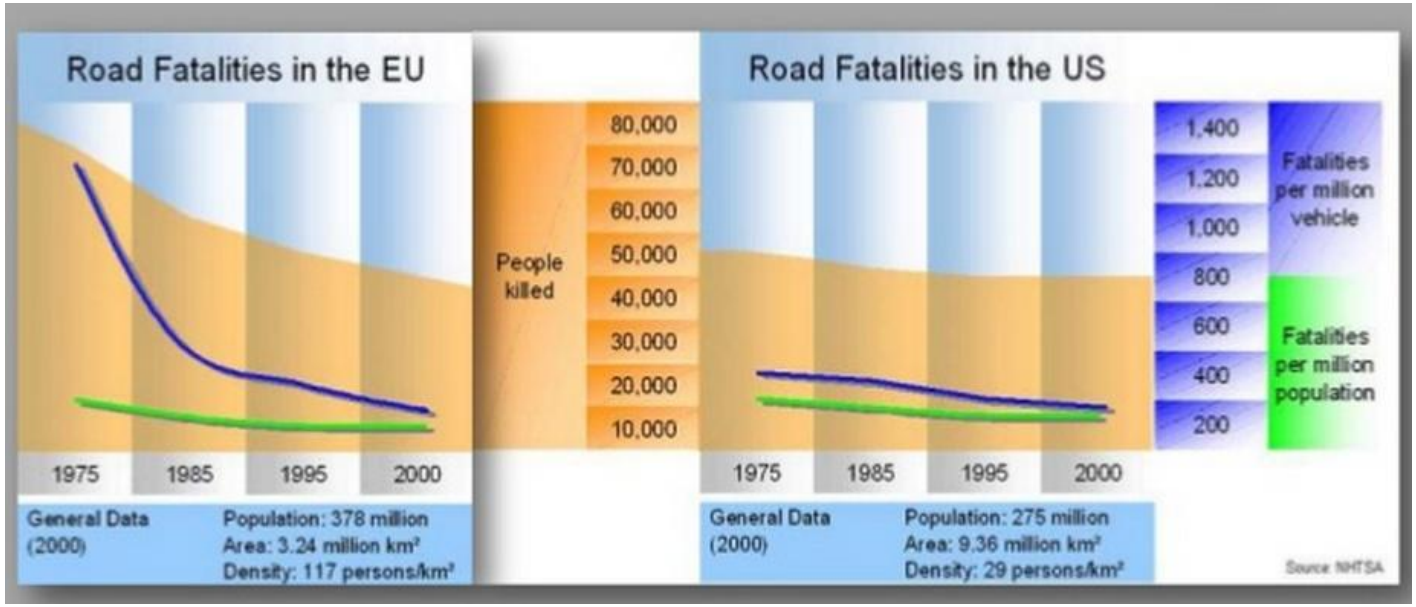
Scenario C



The hardware changes.

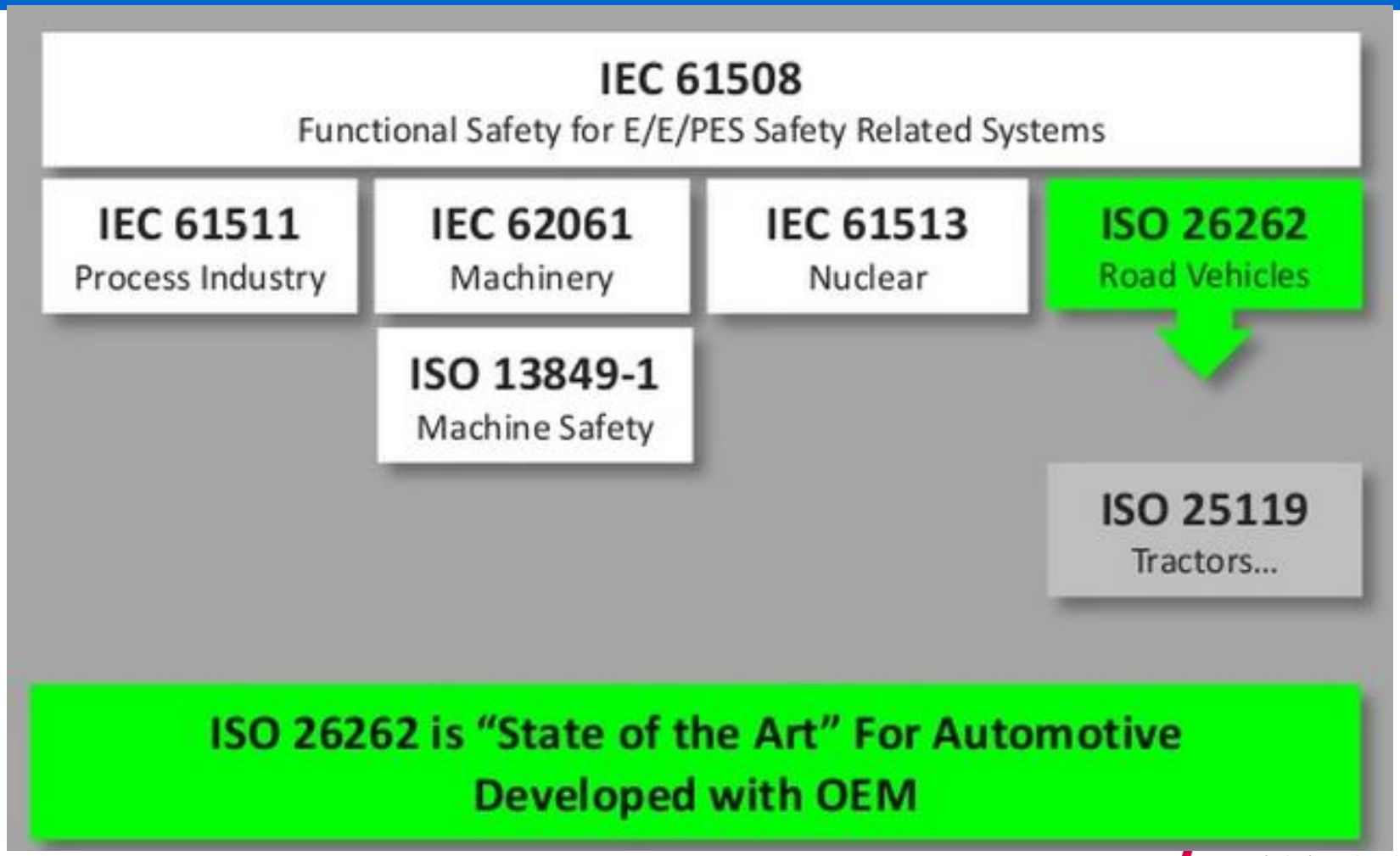
http://autosar.org/

Automotive Standards



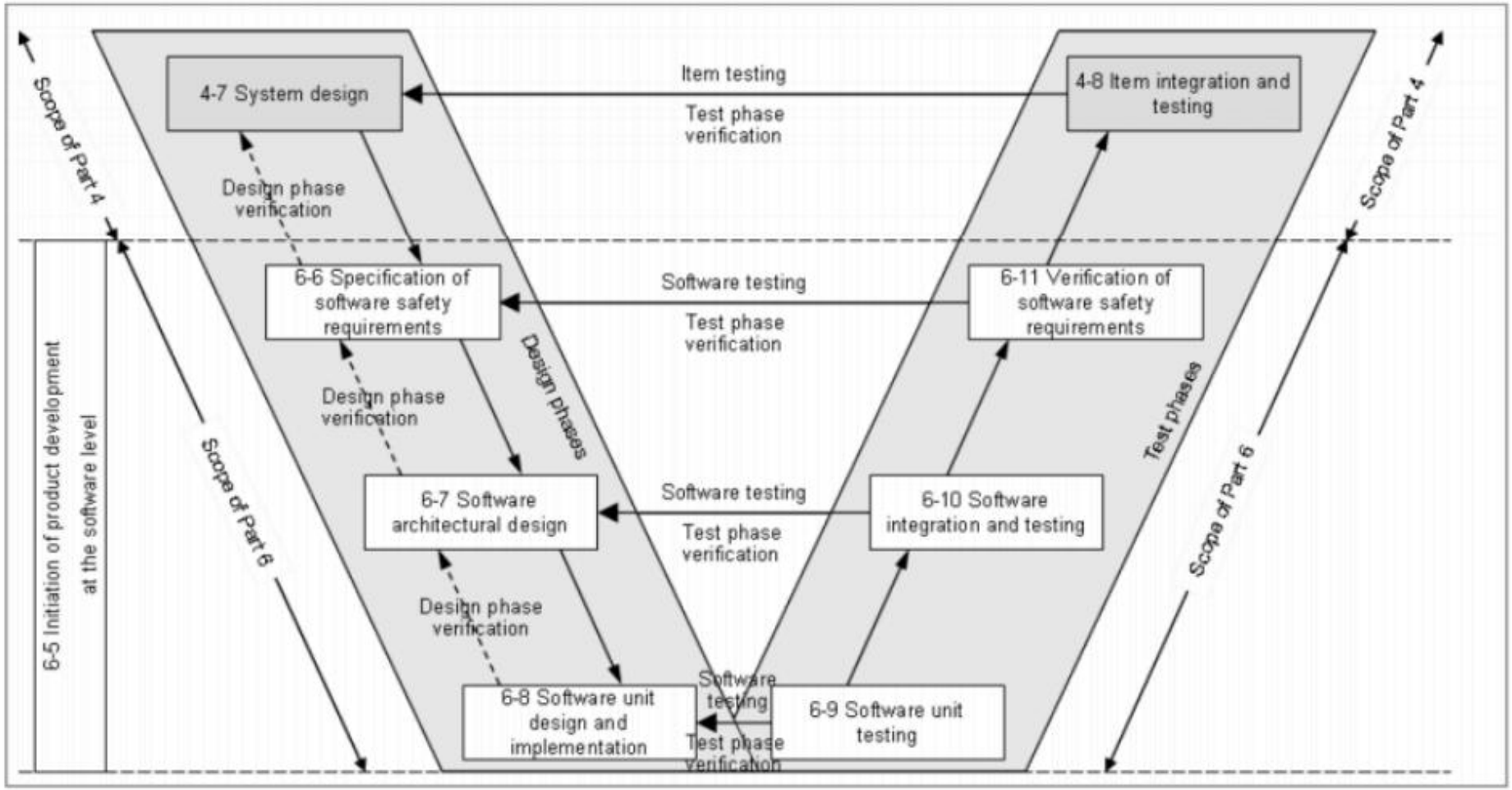
- **ISO 26262:**
  - **Absence of unreasonable risk due to hazards caused by malfunctioning behavior of E/E systems**
- **IEC 61508:**
  - **Part of the overall safety related to the equipment under control (EUC) that depends on the correct functioning of the safety-related system.**
- **MISRA C:**
  - **Software development standard**

ISO 26262

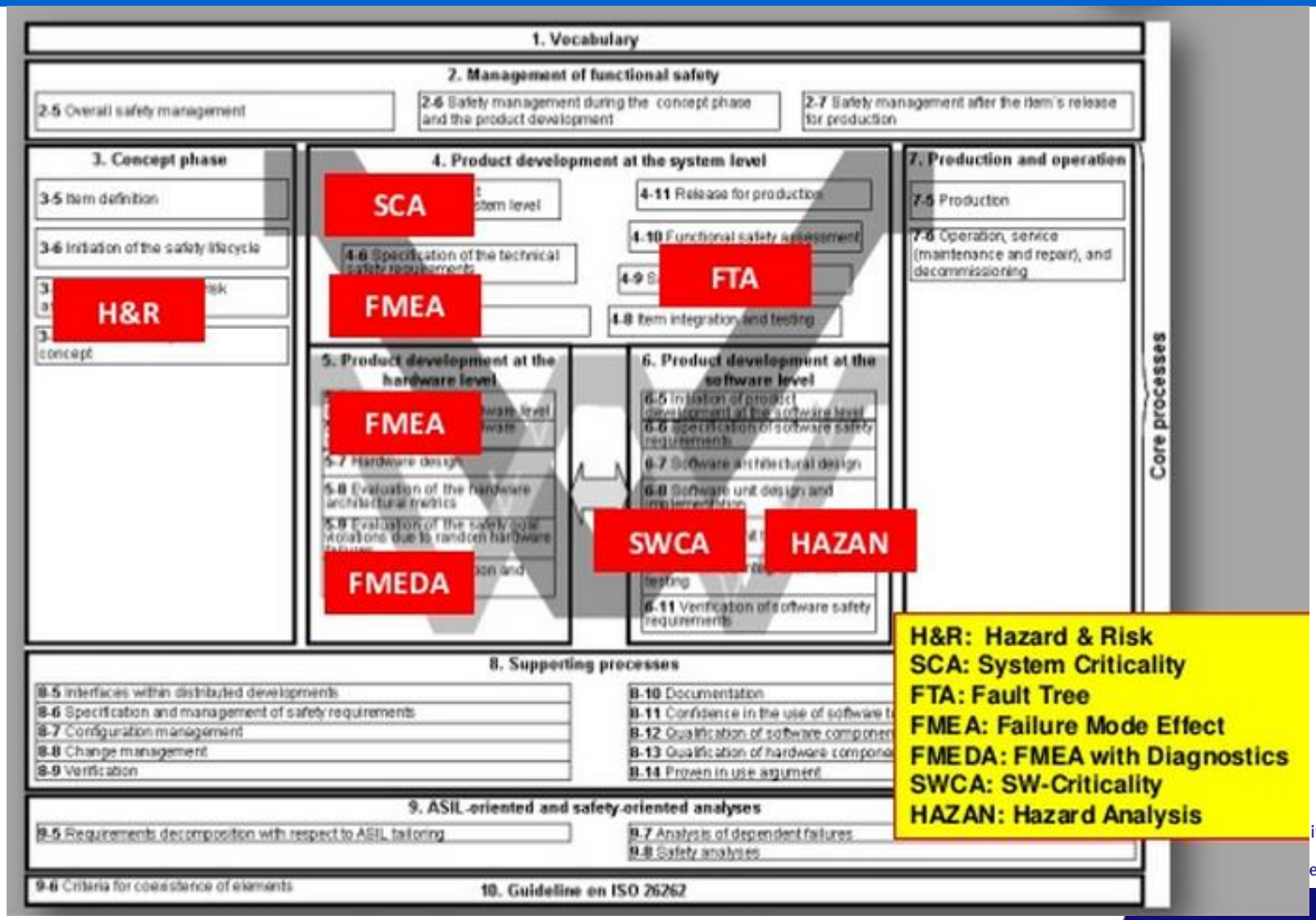


KoenLeekens, ISO-26262 introduction, 2012

Safety in V cycle



Safety Analysis in ISO 26262



KoenLeekens, ISO-26262 introduction, 2012

MISRA C

- **MISRA C is a software development standard for the C programming language developed by MISRA (Motor Industry Software Reliability Association).**
- **Its aims are to facilitate code safety, portability and reliability in the context of embedded systems, specifically those systems programmed in ISO C**
- **As with many standards the MISRA C guideline documents are not free to users or developers**

2014 record recall year

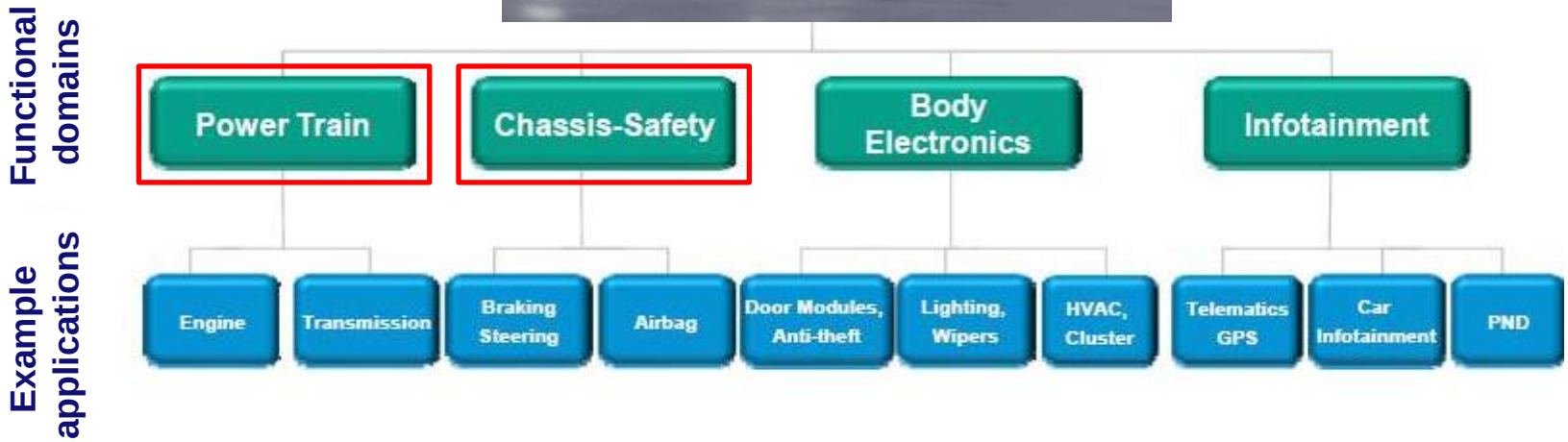
YEAR	TOTAL RECALLS ISSUED	TOTAL NO. OF VEHICLES AND EQUIPMENT RECALLED IN MILLIONS
1990	269	18.5
1991	282	14.4
1992	217	13.6
1993	264	11
1994	290	9.9
1995	348	19
1996	341	19.5
1997	312	16.7
1998	408	19.2
1999	440	55.6
2000	626	44.6
2001	527	22.4
2002	506	25.3
2003	600	22.9
2004	698	33
2005	645	20.4
2006	613	14.1
2007	713	20.6
2008	781	22.6
2009	571	18
2010	723	23
2011	657	17.5
2012	657	18.1
2013	714	27
2014 YTD	*500	**56

Source: National Highway Traffic Safety Administration

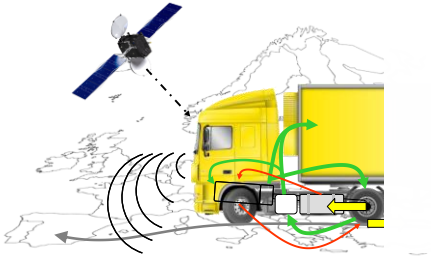
Software problem that could cause

- **the cars to <u>stop suddenly</u>**
- **accelerate <u>without warning</u>**
- **<u>overheats/damages</u> power electronics**
- **…**

'...And I'm saving even more on gas now that I'm afraid to drive it.'



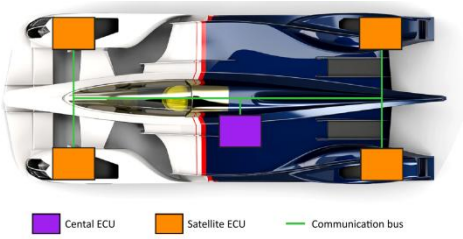
Hybrid Innovations for Trucks (HIT) project



Safety-Critical Domain Certification

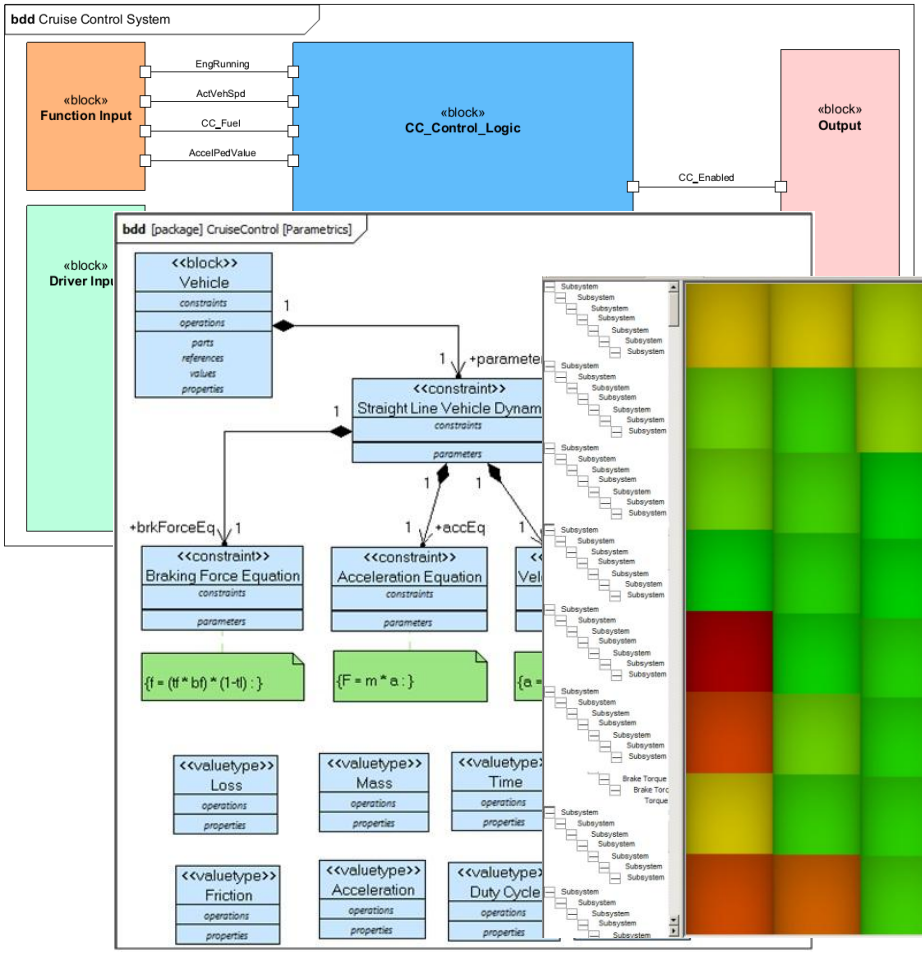
OPENCOSS

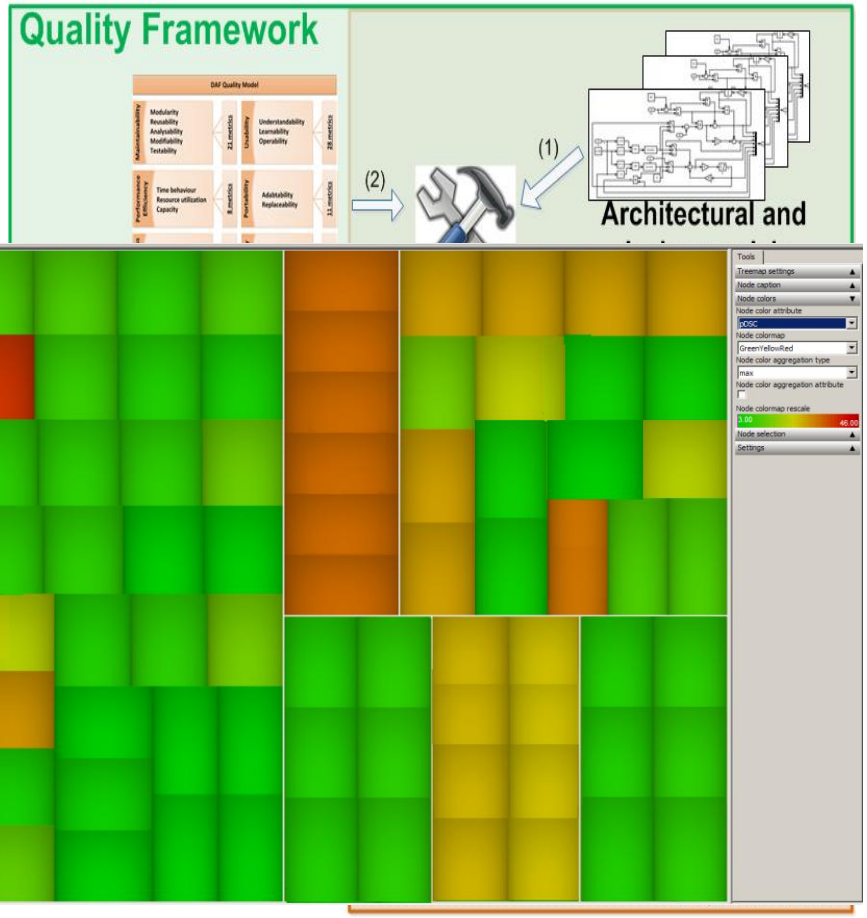
InMotion, Solar Team, “Cars in Context” TU/e projects



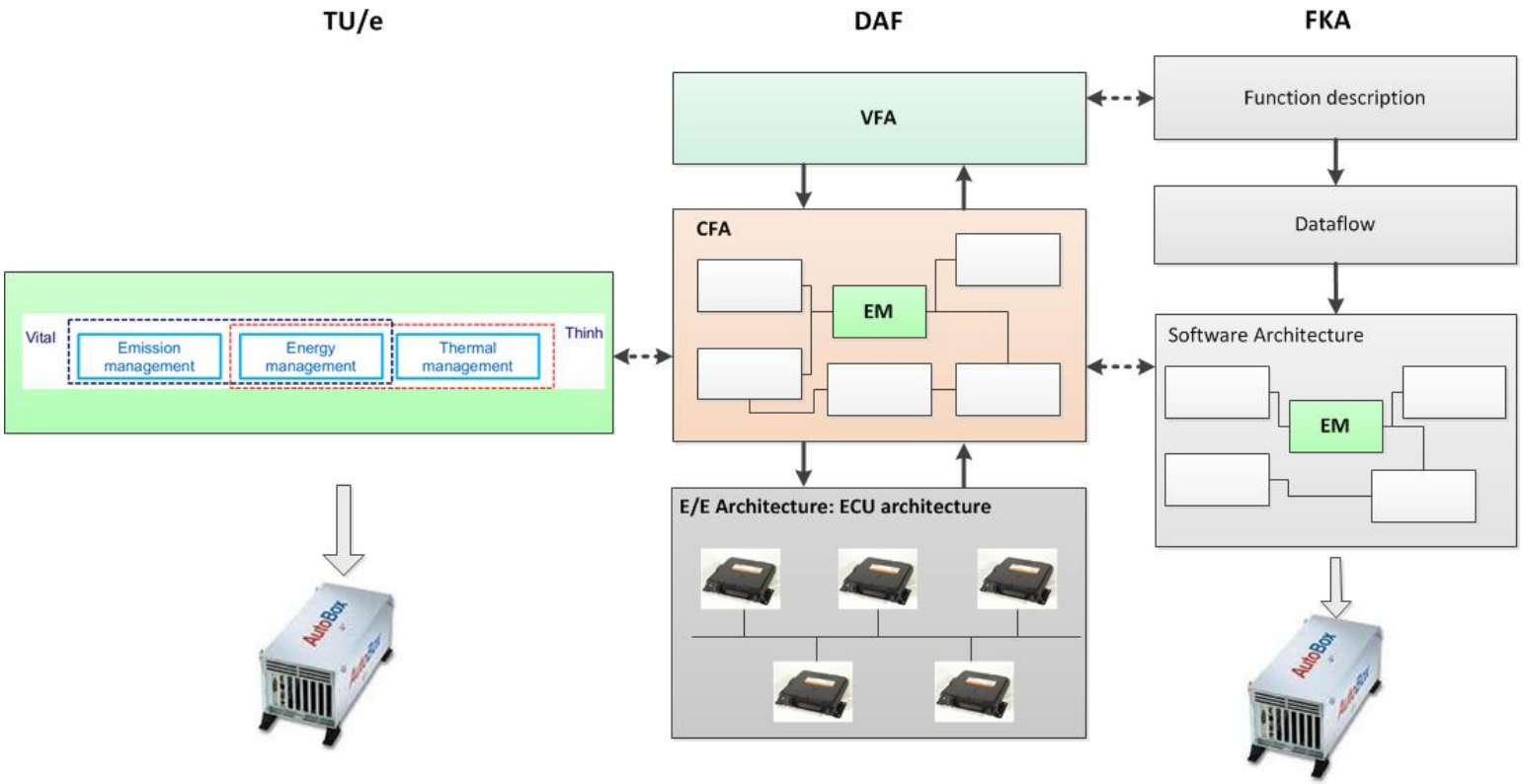
HIT Results

- **Architecture modeling approaches**
  - **ADL evaluation**
  - **Architecture framework**
- **Quality framework**
  - **Validated metrics for Simulink**
  - **Metrics tool**
  - **Visualization**



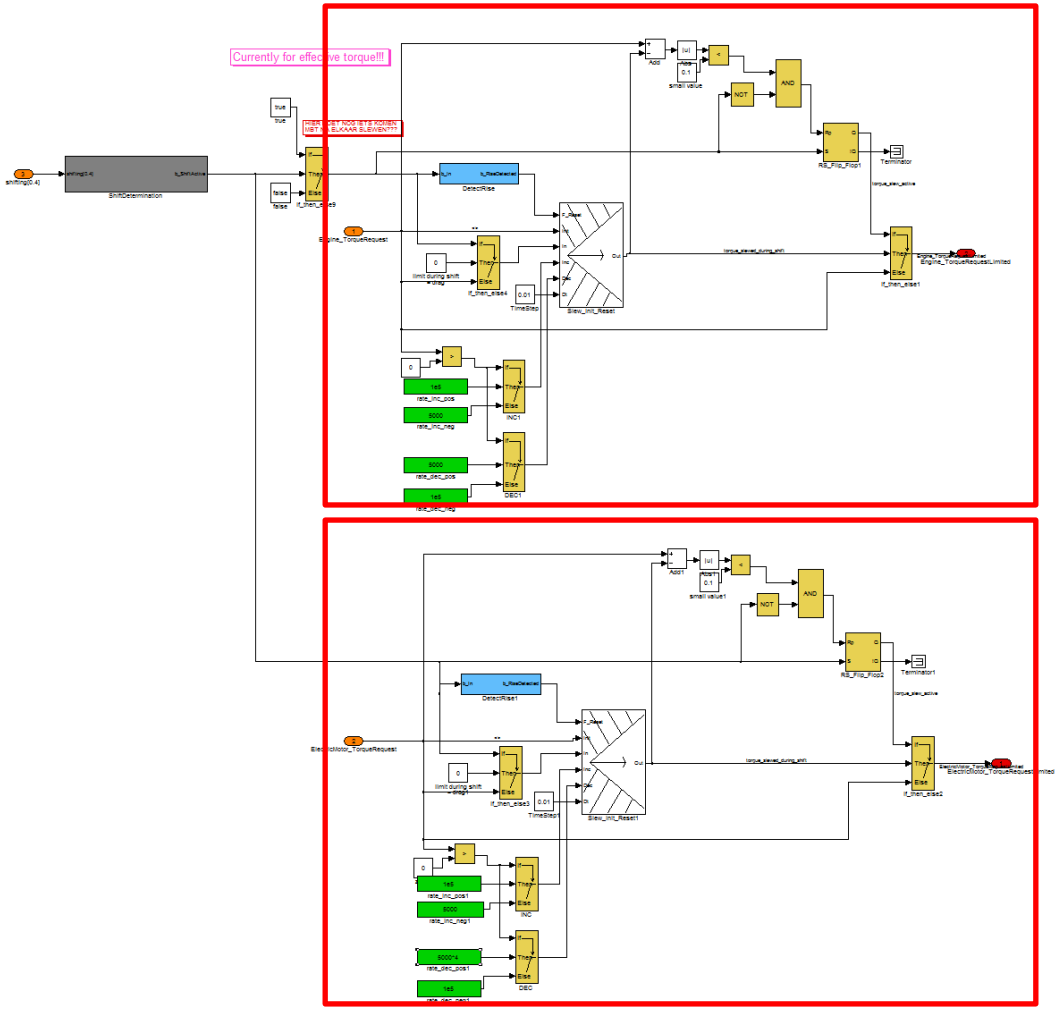


Energy Management (EM) Evaluation

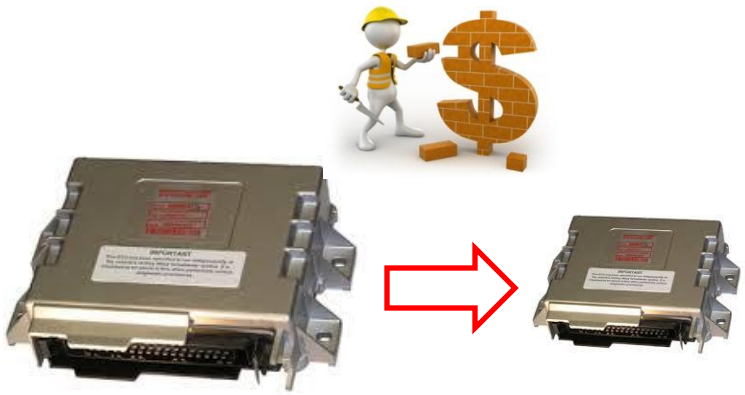


- **Integrated Energy and Thermal Management (IETM) model**
- **Integrated Energy and Emission Management (IEEM) model**

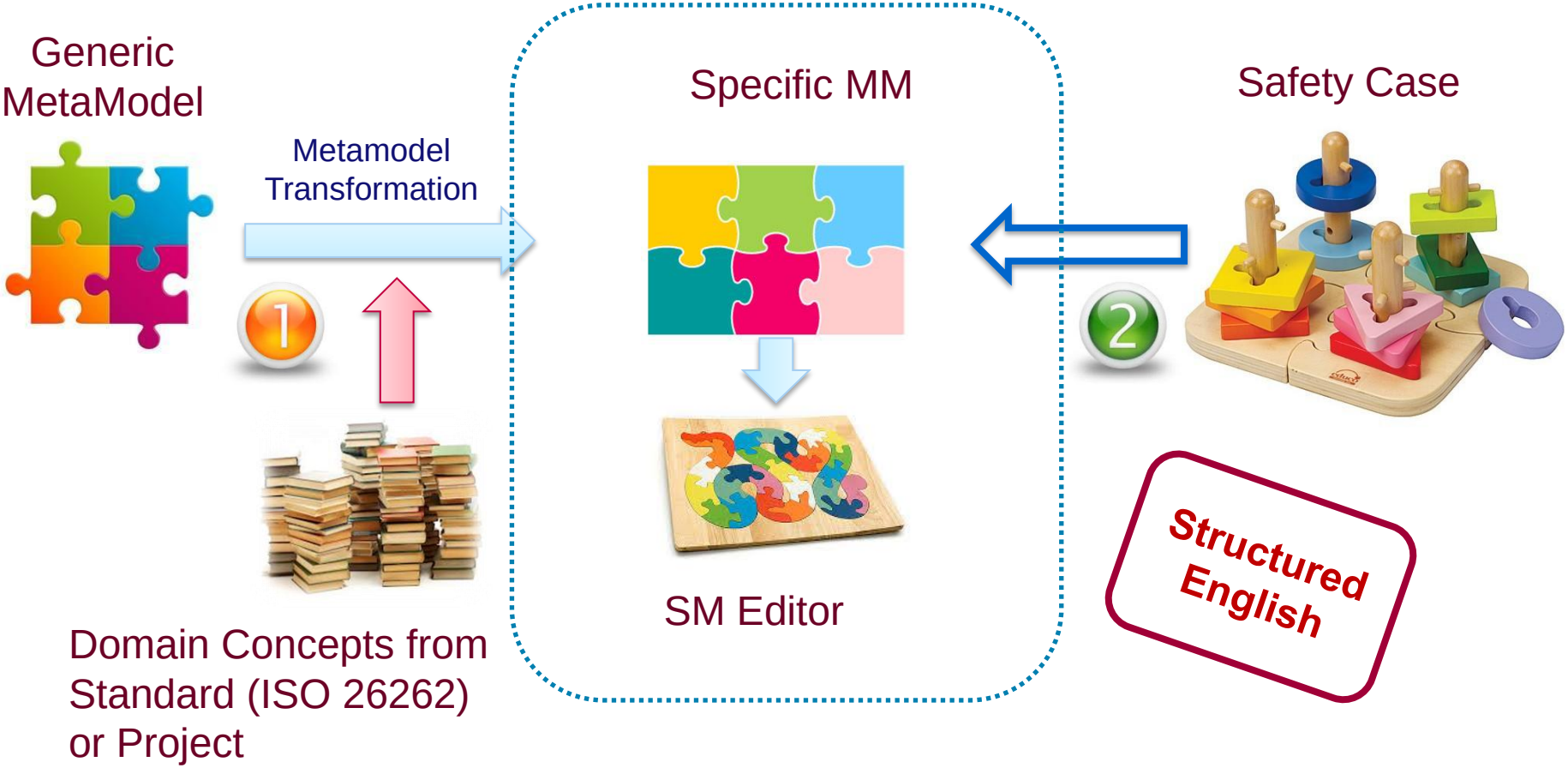
EM Evaluation



- Model clones may have the effect of increasing code size and duplication of errors.
- Number of duplicates affects modifiability as well.



Contribution to GMM based Editors



Add domain concepts

Automatic generated

Link to

Summary

- **In the automotive industry, more and more software and electronics system require system and software architecture methods.**

- **Automotive specific and generic purpose ADLs are being developed and applied.**

- **Many stakeholders, functionalities, safety and environment requirements require automotive specific standards.**


THANK
YOU

Contact for comments and questions:

Yanja Dajsuren
Tel: +31(0)402475052
Email: y.dajsuren@tue.nl
Address:
MF 6.085, Eindhoven University of Technology
5612 AZ Eindhoven, The Netherlands