Variability Modeling in the Automotive Domain

Past, Present and Future Approaches

Stefan Kuntz

P ES GE HSA SW

Introduction
The Key Note Speaker

Group Leader SW Architecture

Stefan Kuntz
Powertrain - Engine Systems
P ES GE HSA SWRBG

Phone: +49 (941) 790-70073
E-Mail: Stefan.Kuntz@continental-corporation.com

10-1987 — SGS Thomson Microelectronics
- SW/Production Engineer
- uC Design and Verification

07-1990 — FORCE Computers
- SW Engineer, SW Architect, Chief Architect
- Multiprocessor Computing Systems

04-1999 — Xpherix
- Chief Architect, Director Engineering
- Internet Enterprise Applications

01-2003 — Micron Electronic Devices
- Team Leader
- SQA and Methods, ASPICE

09-2004 — Continental Automotive
- Project Manager Advanced Development
- Public Funded Projects, AUTOSAR, …

01-2014
- Group Leader SW Architecture

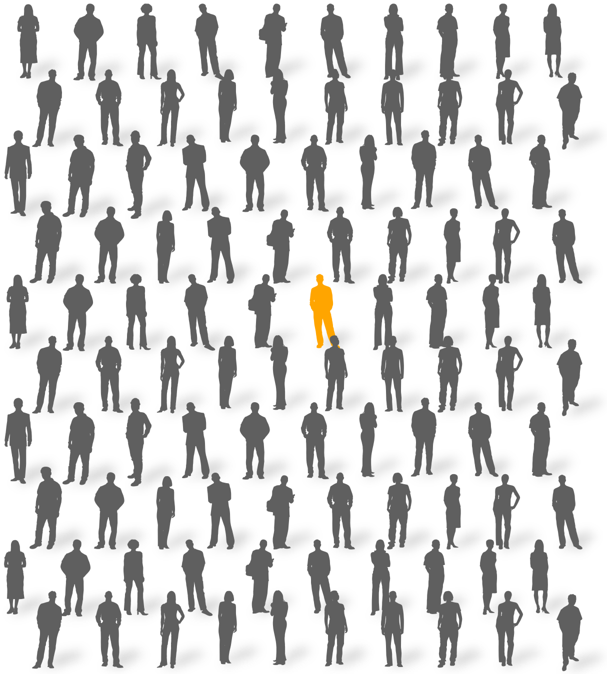
Agenda

1	Automotive Domain
2	Past
3	Present
4	Future
5	Conclusion

Automotive Domain
Significant Driving Forces

- Cost sensitive
- Resource constraints
- Short time-to-market

- Increasing functionality
- Larger scope of distribution of functionality
- Tighter collaboration among organizations
- High degree of division of labor



Continental Corporation
Five Strong Divisions

Chassis & Safety	Powertrain	Interior	Tires	ContiTech
Vehicle Dynamics	Engine Systems	Instrumentation & Driver HMI	PLT, Original Equipment	Air Spring Systems
Hydraulic Brake Systems	Transmission	Infotainment & Connectivity	PLT, Repl. Business, EMEA	Benecke-Kaliko Group
Passive Safety & Sensorics	Hybrid Electric Vehicle	Intelligent Transportation Systems	PLT, Repl. Business, The Americas	Compounding Technology
Advanced Driver Assistance Systems (ADAS)	Sensors & Actuators	Body & Security	PLT, Repl. Business, Asia Pacific	Conveyor Belt Group
	Fuel & Exhaust Management	Commercial Vehicles & Aftermarket	Commercial Vehicle Tires	Elastomer Coatings
			Two Wheel Tires	Fluid Technology
				Power Transmission Group
				Vibration Control

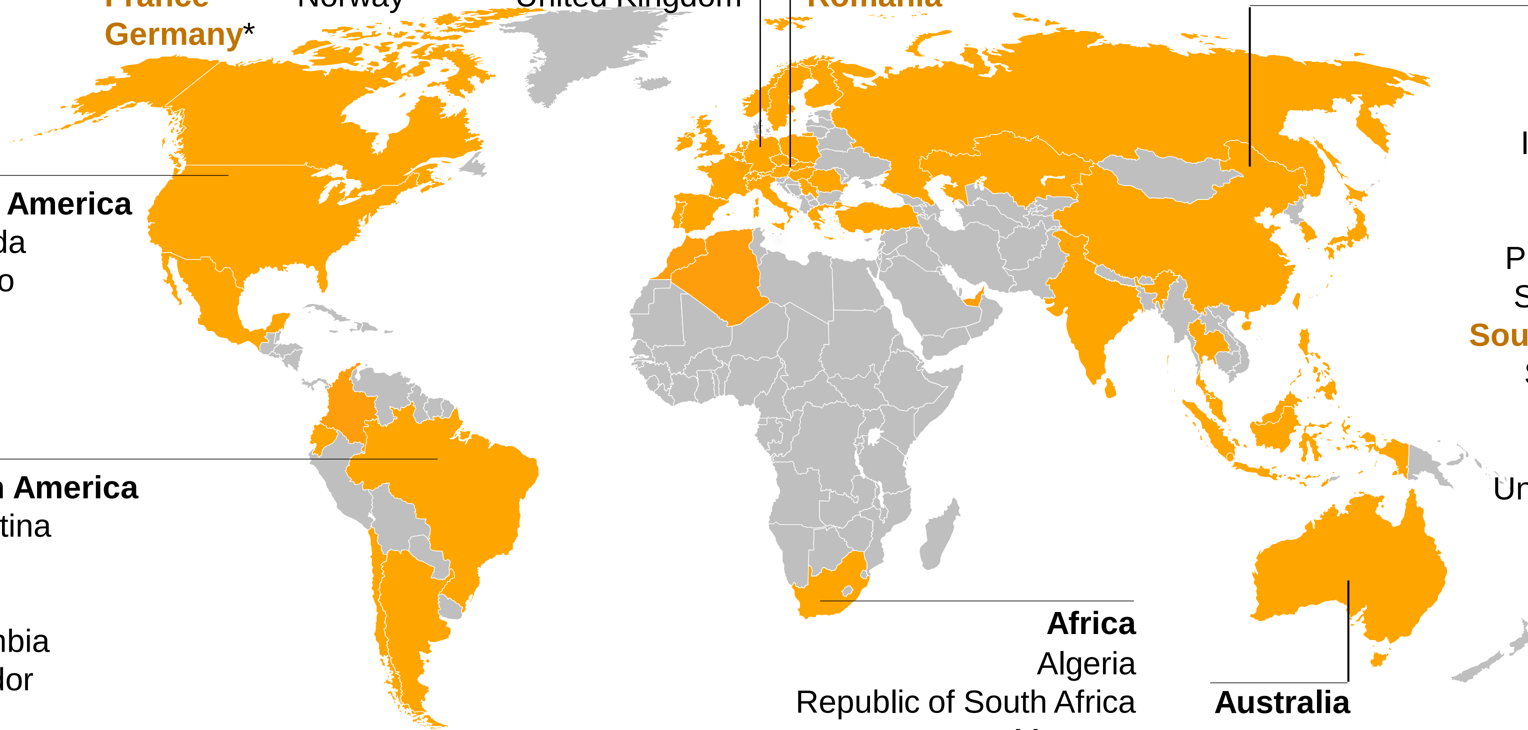
Automotive

PLT – Passenger and Light Truck Tires

Continental Corporation
300 Locations in 49 Countries

Europe

Austria
Belgium
Denmark
Finland
France
Germany*
Greece
Ireland
Italy
Netherlands
Norway
Portugal
Spain
Sweden
Switzerland
United Kingdom
Czech Republic
Hungary
Kazakhstan
Poland
Romania
Russia
Serbia
Slovakia
Turkey



North America

Canada
Mexico
USA

South America

Argentina
Brazil
Chile
Columbia
Ecuador

Asia

China
India
Indonesia
Japan
Malaysia
Philippines
Singapore
South Korea
Sri Lanka
Taiwan
Thailand
United Arab Emirates

Africa

Algeria
Republic of South Africa
Morocco

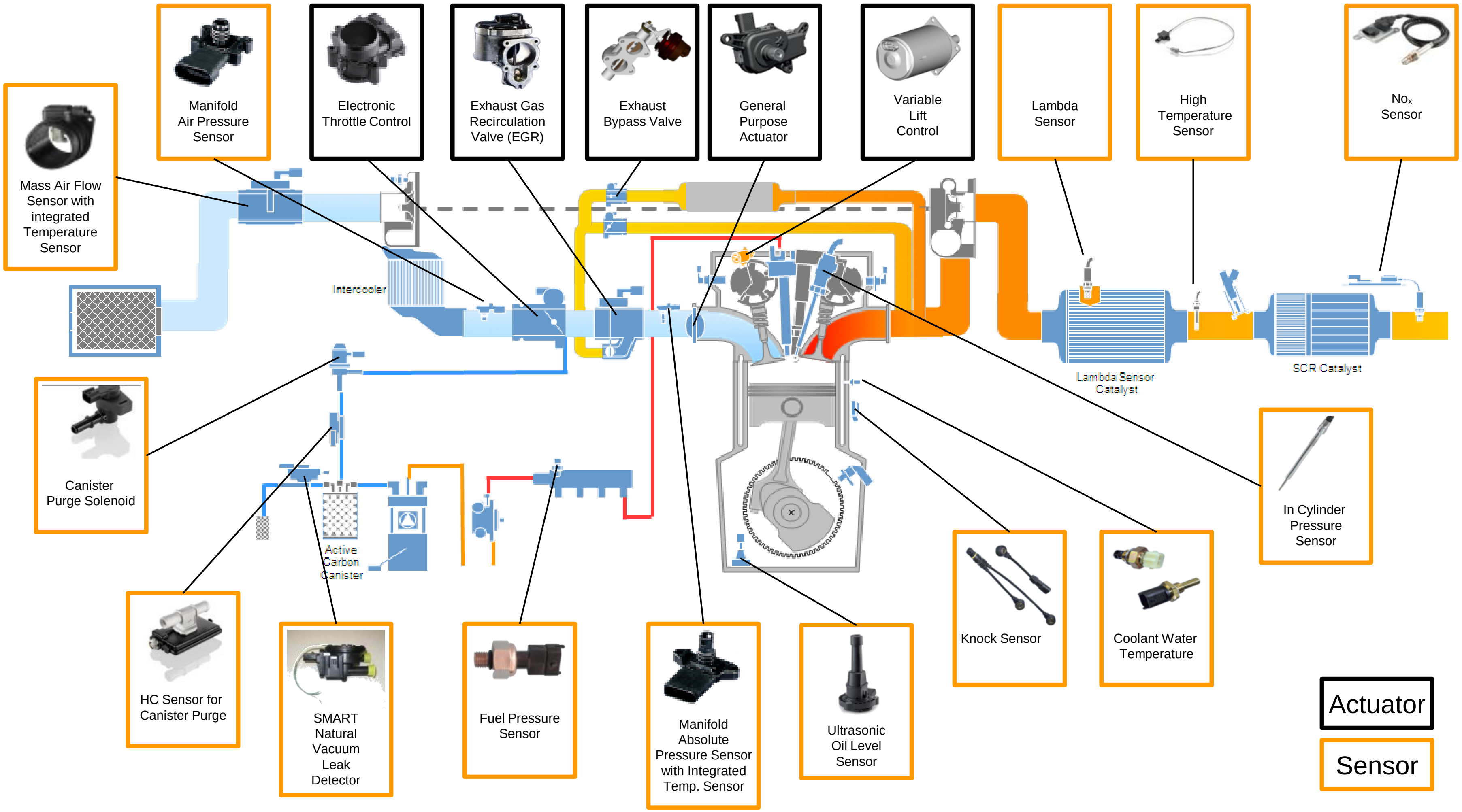
Australia

*Headquarters in Hanover

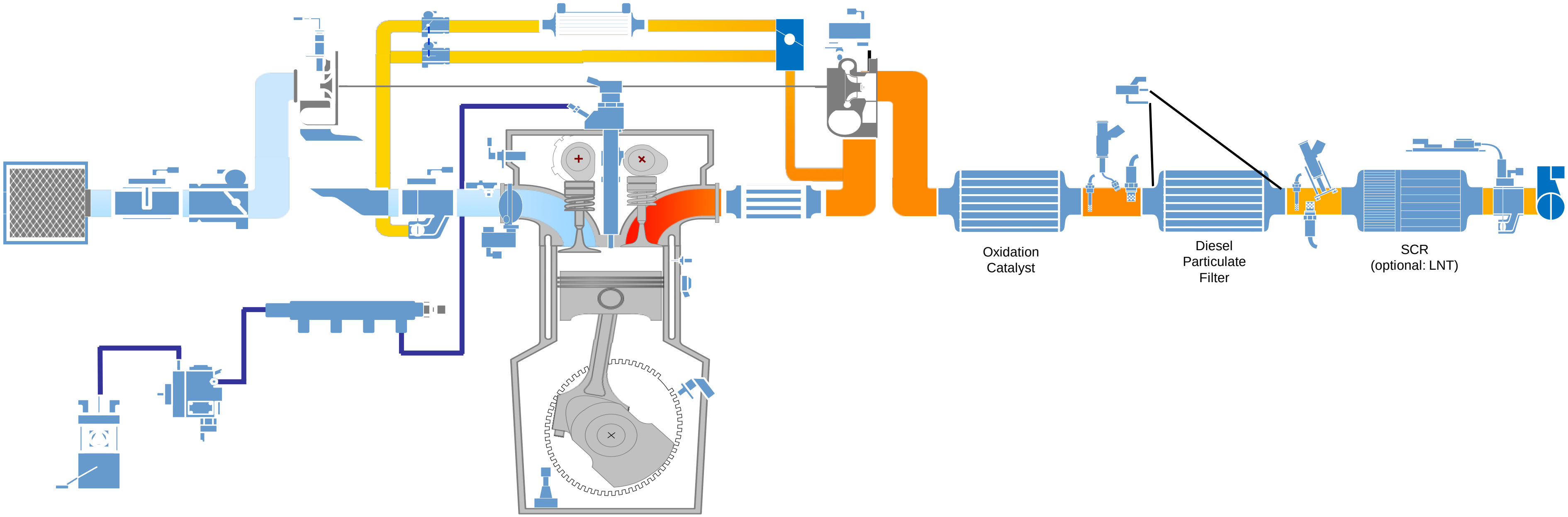
Status: January 2014

Automotive Domain

Powertrain Engine Systems: Gasoline DI and SCR



Automotive Domain
Powertrain Engine Systems: Diesel DI

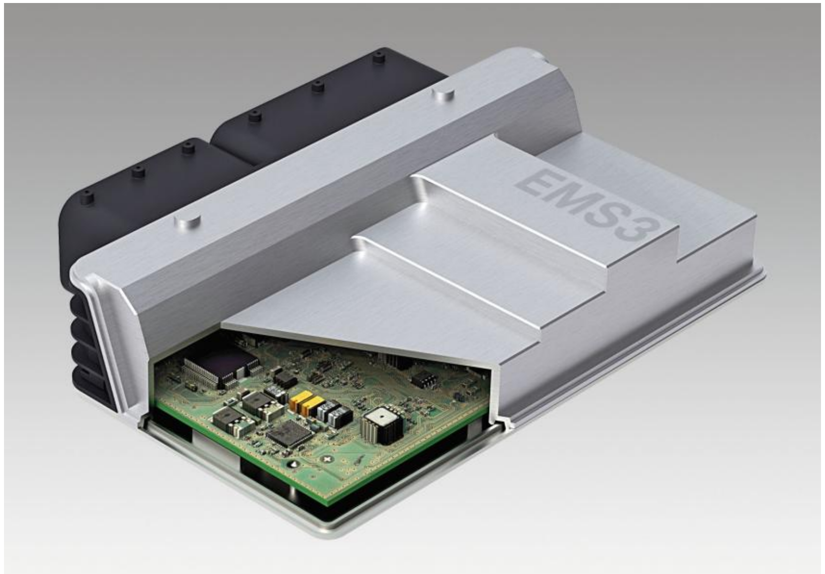


Automotive Domain
Example: Combustion Engine Management System

- Roughly over 6.000 system/engine variants
- Software:
  - 70 up to 140 functionalities
  - 4.000 to over 8.000 executable units
  - 20.000 to over 42.000 data/variables
  - 1.5 to 2.2 MByte of ROM
  - 0.75 to 1.5 MByte of RAM
  - Calibration data

Hardware/ECU:

  - Minimum 7.000 variants



Automotive Domain
Example: Smart NOX Sensor

- Roughly 200 Variants
  - Three different micro controller families
  - Different standard and proprietary communication protocols
  - Various functionality including third-party functionalities
  - Diagnostics
  - Number of NOX sensors in a vehicle - NOX Sensor Network
  - Passenger and commercial vehicles

Automotive Domain
Conclusion

Taking all elements of a system that are varying into account the challenge is to manage several thousands of variants

!

Variations shall be managed

- at different levels of abstraction, and
- at various activities in the development and maintenance process

It is an architectural topic: Requirement - Structure - Variability

Agenda

Past
Specification of Variation

- Manual approach
- Document based supported by proprietary tool chain - documents and tables
- Verification and validation by reviews - Four-Eyes Principle
- Detection of inconsistencies at build/compile time respectively specific testing/inspections

Specification of variation - Definition of Variants

1.4 Version and Configuration

Versions:

Data Name	Values	Physical Meaning	S* (Y/N)
NC_ENSD_VERS	01	With Camshaft Sensor	Y
	02	Without Camshaft Sensor	N
	03	For HED engines	N

S = Supported*

Legend:
HED Hybrid Electric Drives
NC Non Configurable

Programming Language Artifact

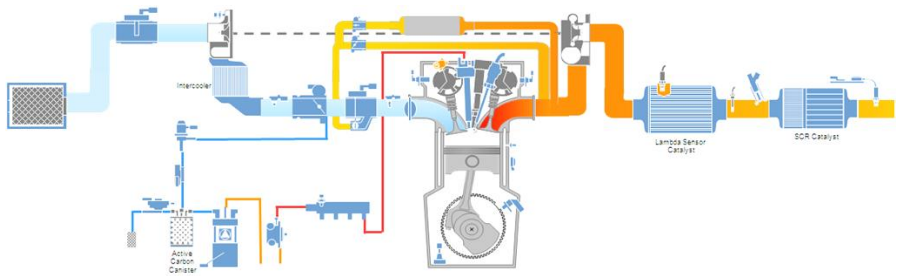
```
int function ( ... ) {

  ... <statements> ...
#if NC_ENSD_VERS == 1
  ... <statements> ...
#endif
  ... <statements> ...
  return( OK );
}
```

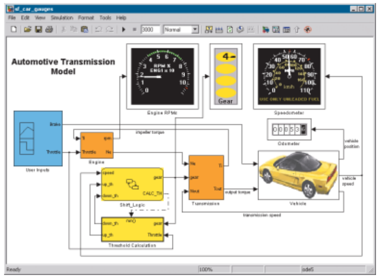
Agenda

Present

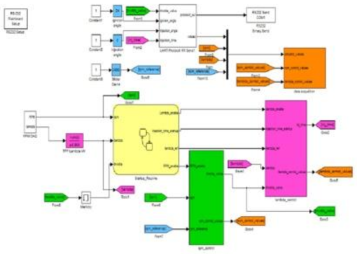
Levels of Abstraction

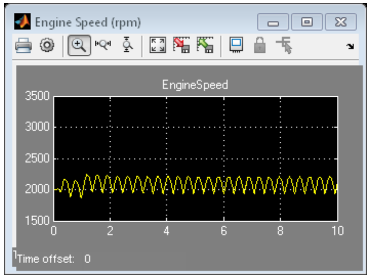
System

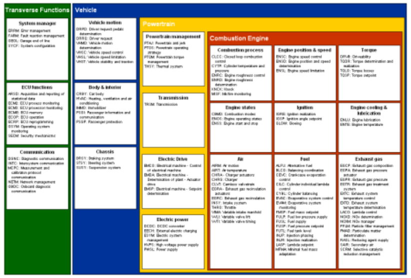




Function



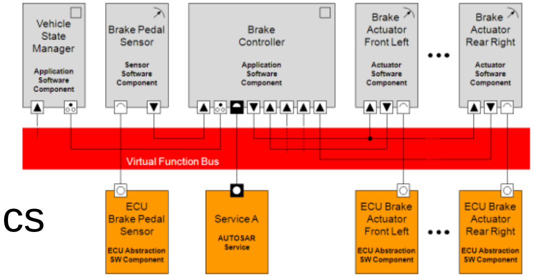


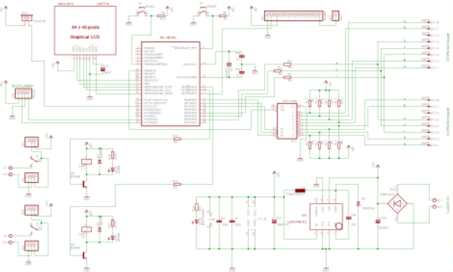


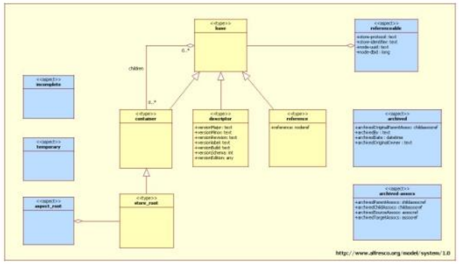
Software

Hardware

Mechatronics





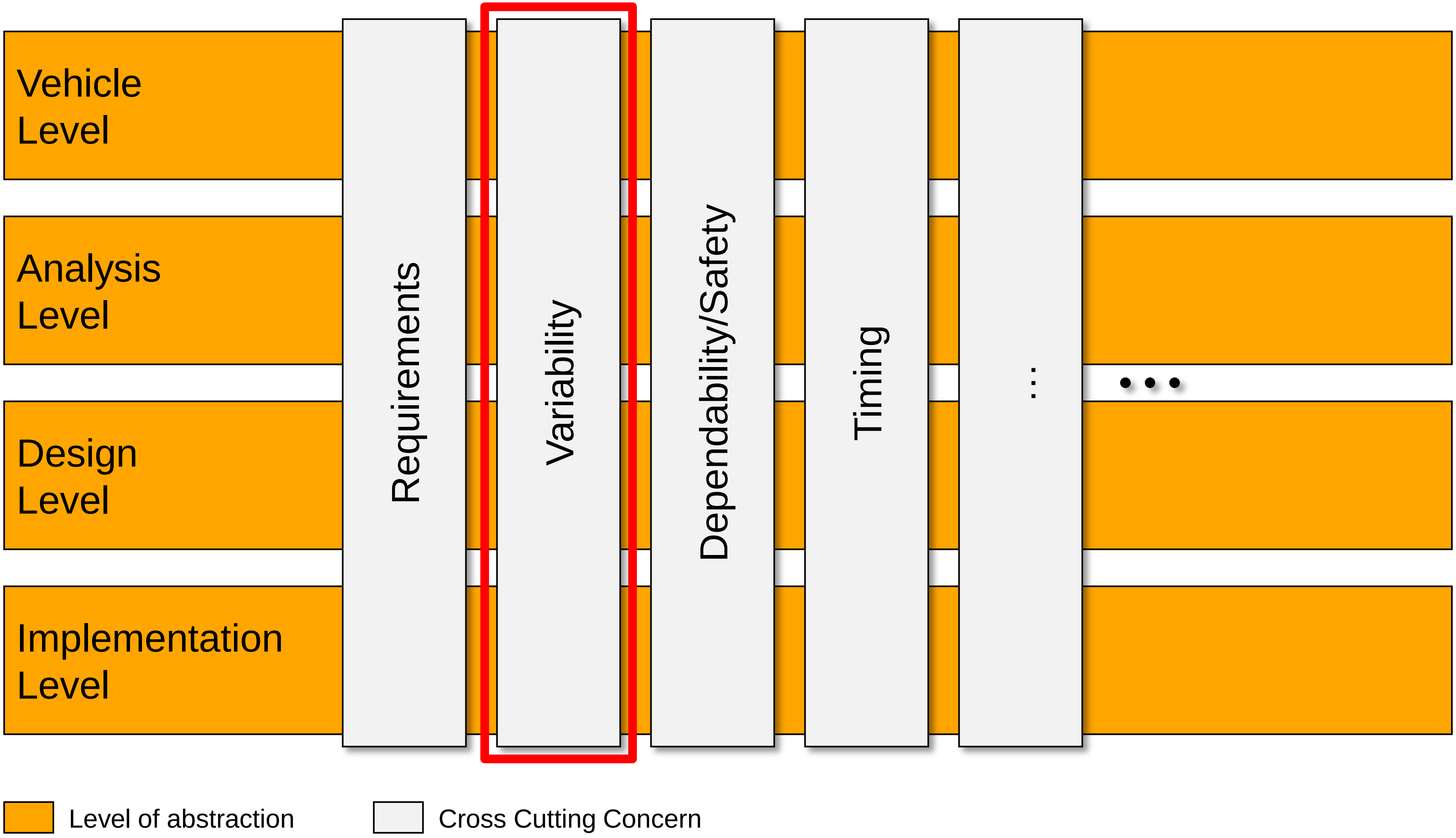


Present
EAST-ADL - Purpose of Abstraction Levels

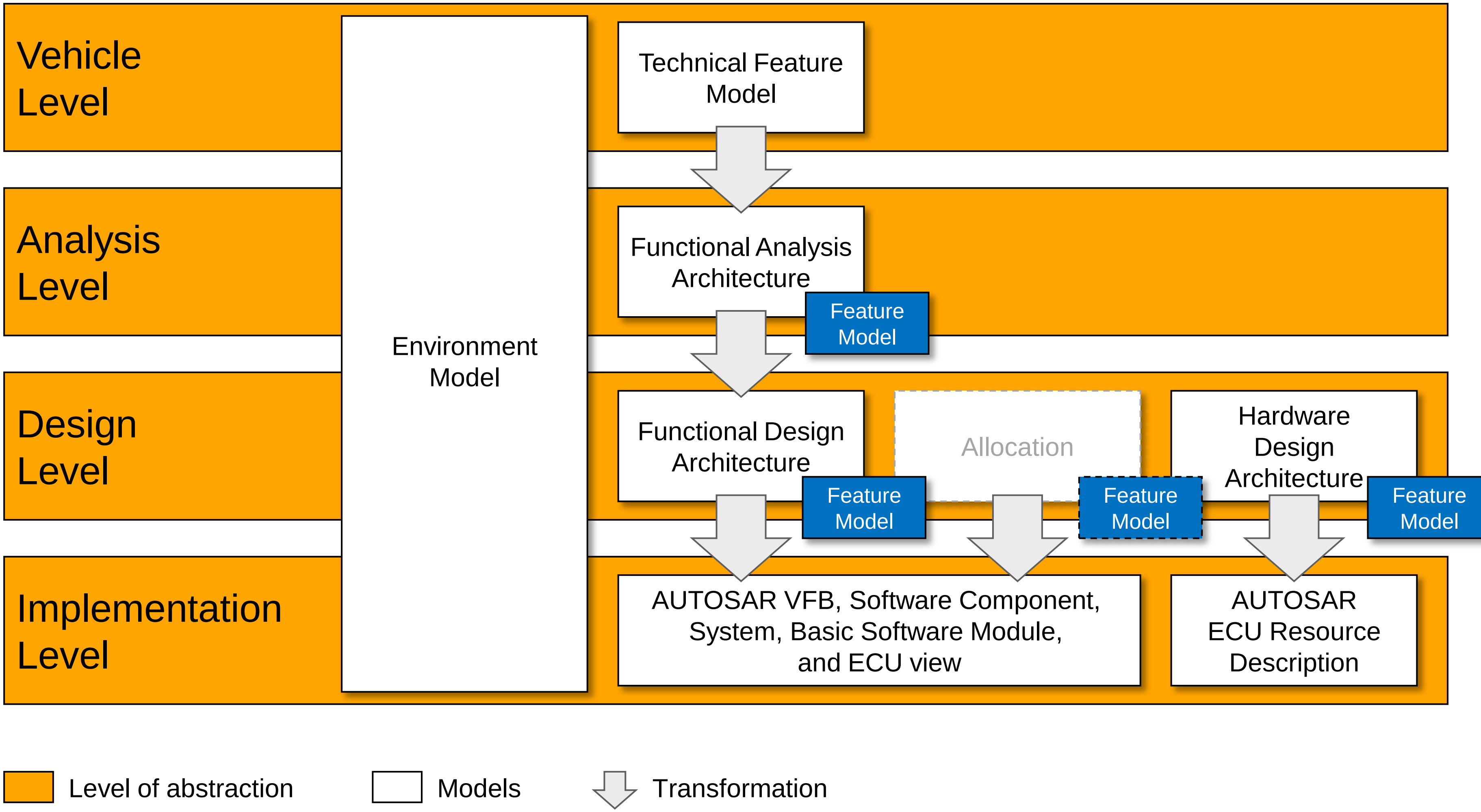
Vehicle Level	This level describes the features visible to the stakeholder “driver” such as windscreen wipers, window lifter, cruise control, etc, as well as the dependencies between these features.
Analysis Level	This level captures the external visible behavior and algorithms of the functionality, as well as the inter-dependencies between these functionalities. “What the system shall do?”
Design Level	This level represents the realization of each functionality analyzed on the Analysis Level. It represents the logical architecture. “How the system is doing what it shall do?”
Implementation Level	This level describes the implementation of the functionality described on the Analysis and Design Level. It represents the technical architecture, and consists of software, hardware and mechanics.

Level of abstraction

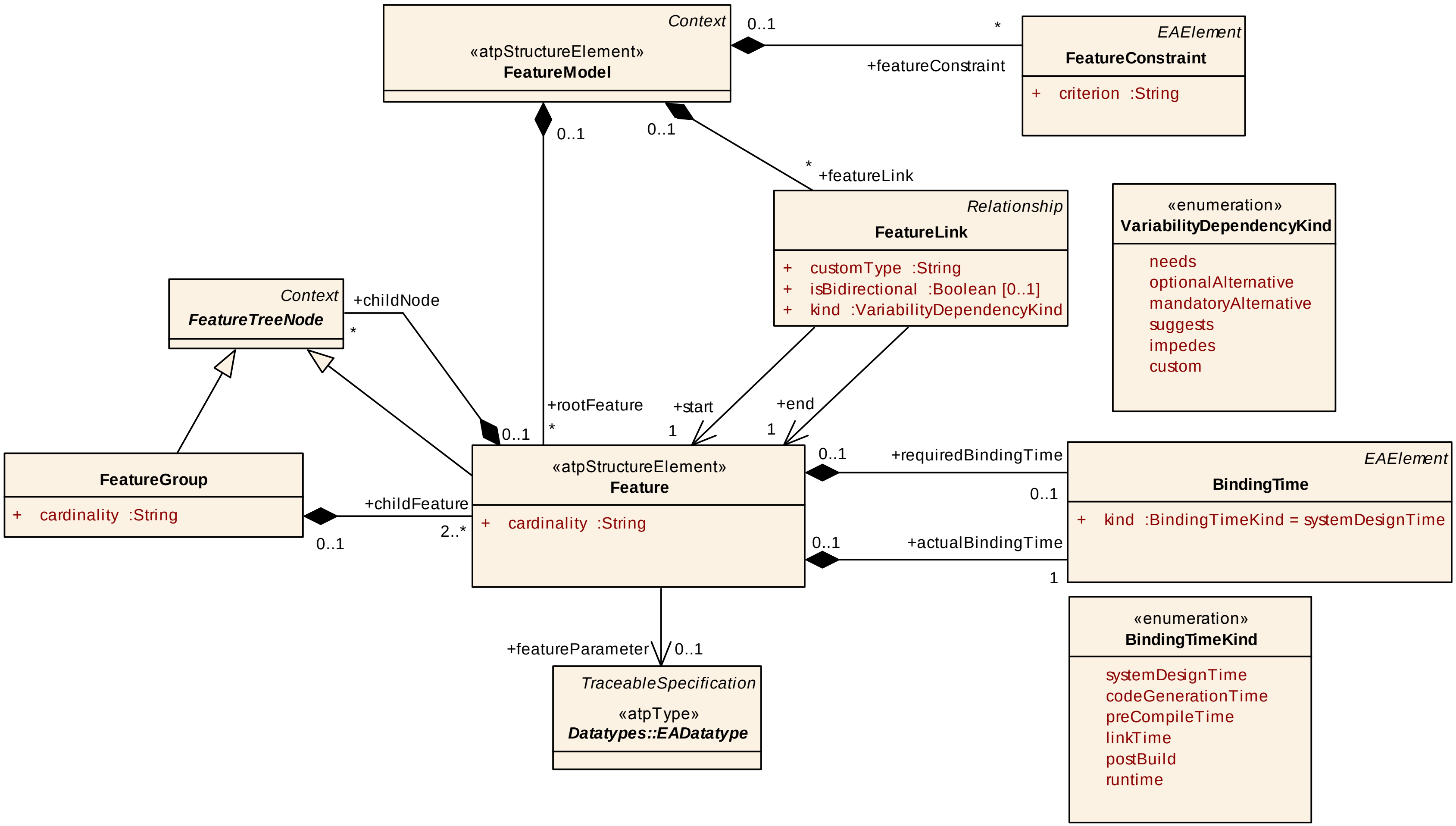
Present
EAST-ADL - Cross-Cutting Concerns



Present
EAST-ADL: Feature Modeling



Present
EAST-ADL: Feature Modeling Meta Model



Present
Brief Introduction into AUTOSAR - Objectives

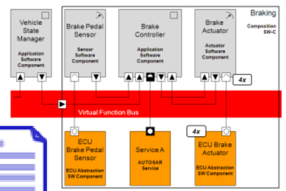
- Standardization of basic software functionality of automotive ECUs
- Scalability to different vehicle and platform variants
- Transferability of software
- Support of different functional domains
- Definition of an open architecture
- Collaboration between various partners
- Development of highly dependable systems
- Support of applicable automotive international standards and state-of-the-art technologies

Present

Brief Introduction into AUTOSAR - Methodology

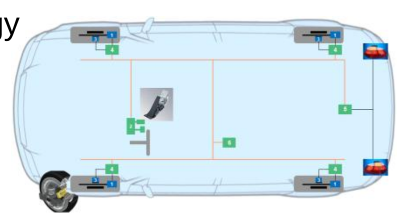
System - OEM

Views:
- Virtual Function Bus (VFB)
- Software Component (SW-C)
- System Topology



SW-C Descriptions
VFB, SW-C

ECU Resource Descriptions
System Topology

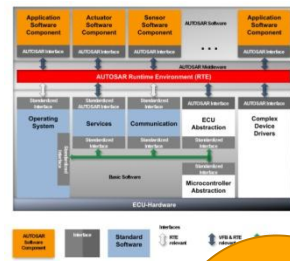


System Configuration Generator

ECU Extract of System Configuration

Electronic Control Unit (ECU) - Supplier

Views:
- ECU
- Basic Software (BSW) Module



ECU Configuration Generator

Runtime Environment (RTE)

Operating System (OS)
Communications (COM)

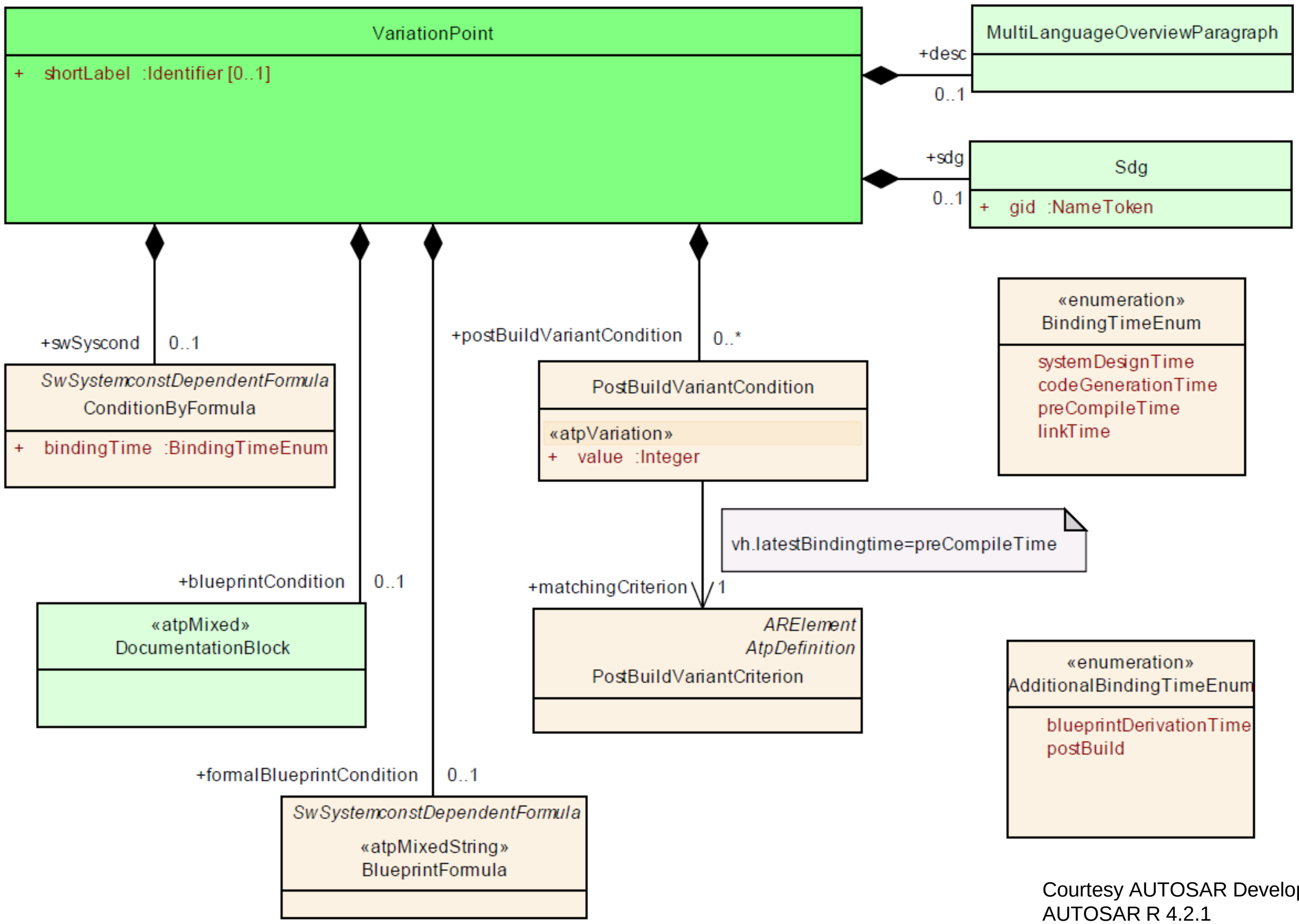
BSW Modules (BSWM)
µC Abstraction Layer (MCAL)

Software Components (SW-C)

Present
AUTOSAR Support for Variability

- Variation Point
  - AUTOSAR meta model: stereotype «atpVariation»
  - Controlled by System Constants
  - Aggregation
  - Association
  - Attribute Value
  - Property Set
- Feature Model
  - Specific AUTOSAR template
  - Feature Selection and Feature Map

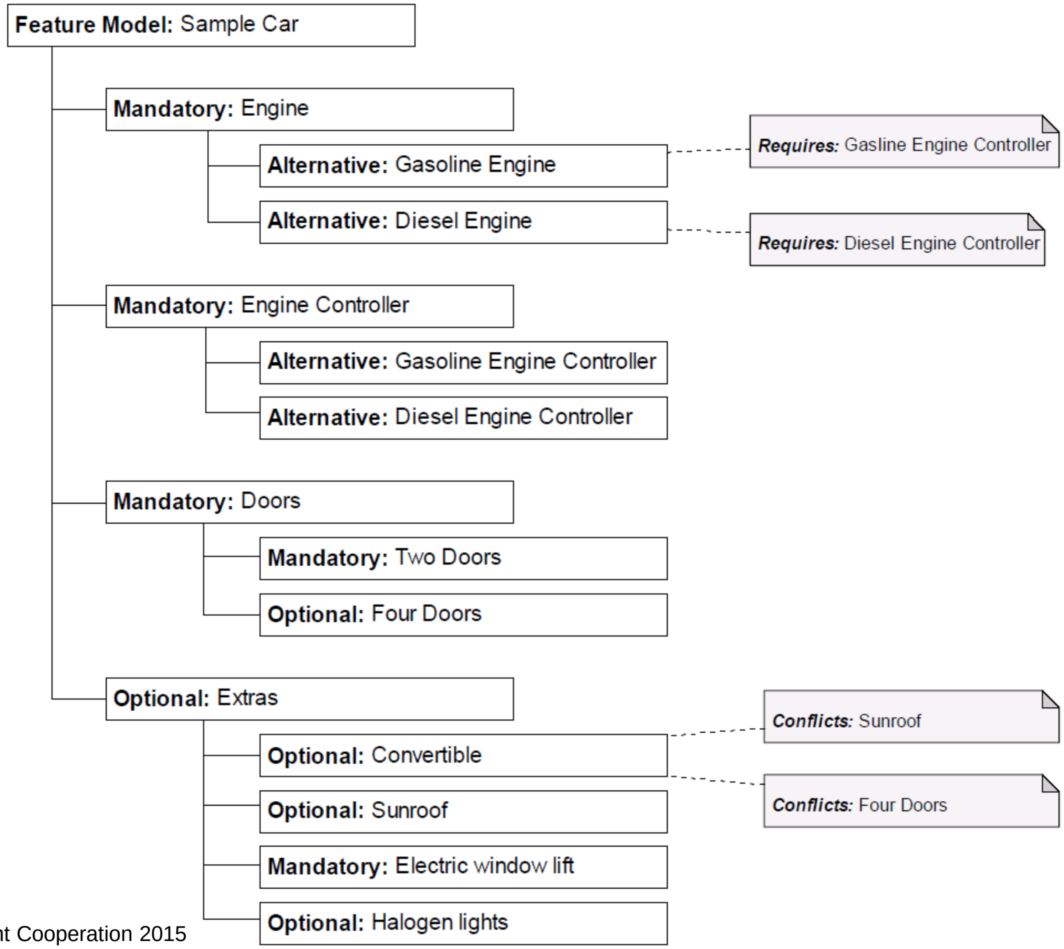
Present
AUTOSAR Variation Point



Courtesy AUTOSAR Development Cooperation 2015
AUTOSAR R 4.2.1

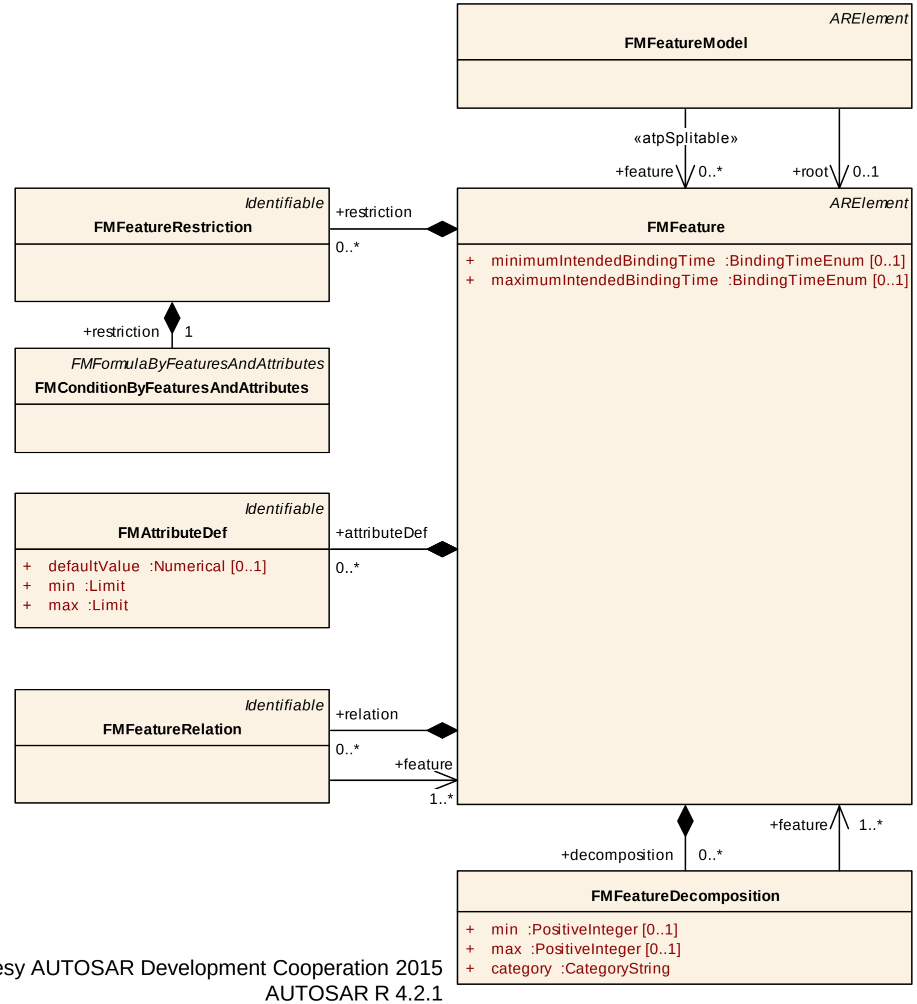
Present
AUTOSAR Feature Model

- Feature Model - Hierarchical model containing features and its dependencies
- Feature Selection - Decision model
- Feature Map - Mapping features and system constants



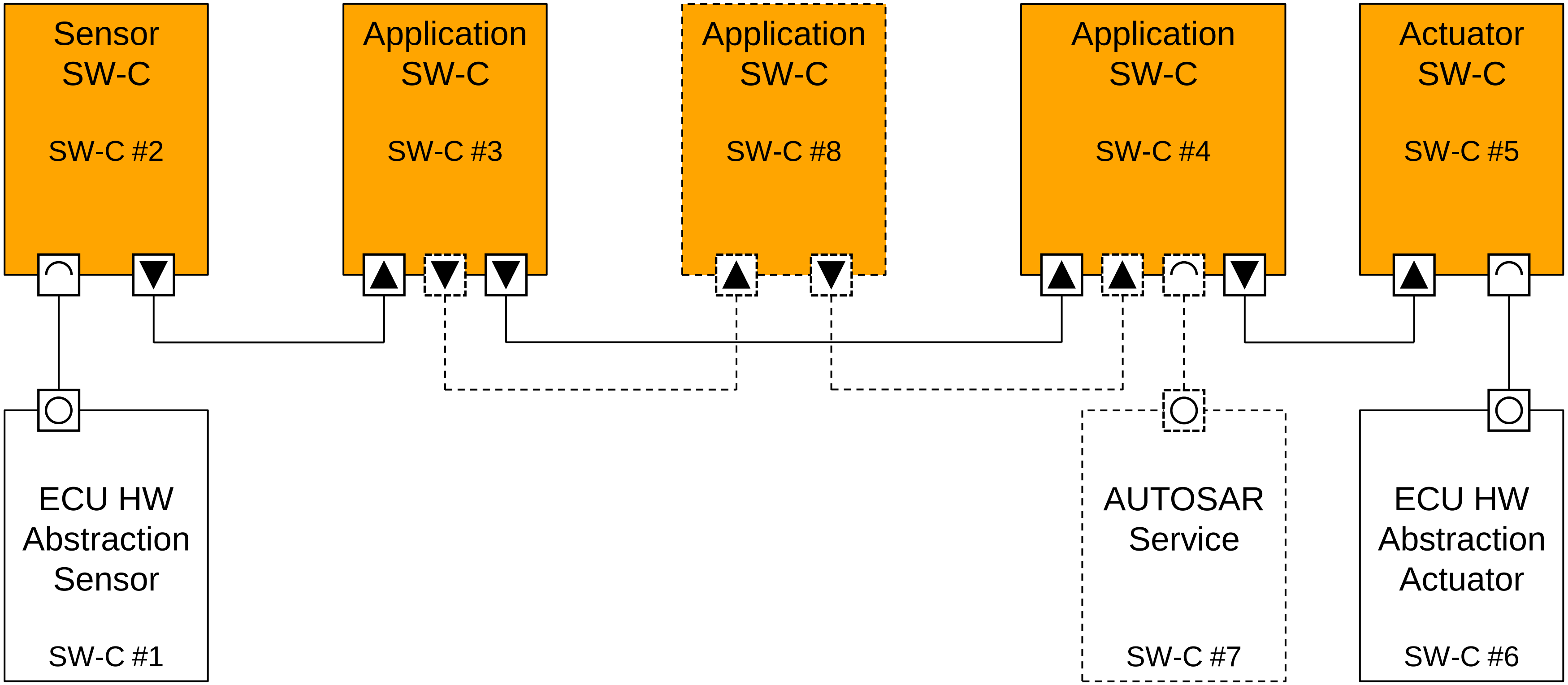
Courtesy AUTOSAR Development Cooperation 2015
AUTOSAR R 4.2.1

Present
AUTOSAR Feature Model - Meta Model

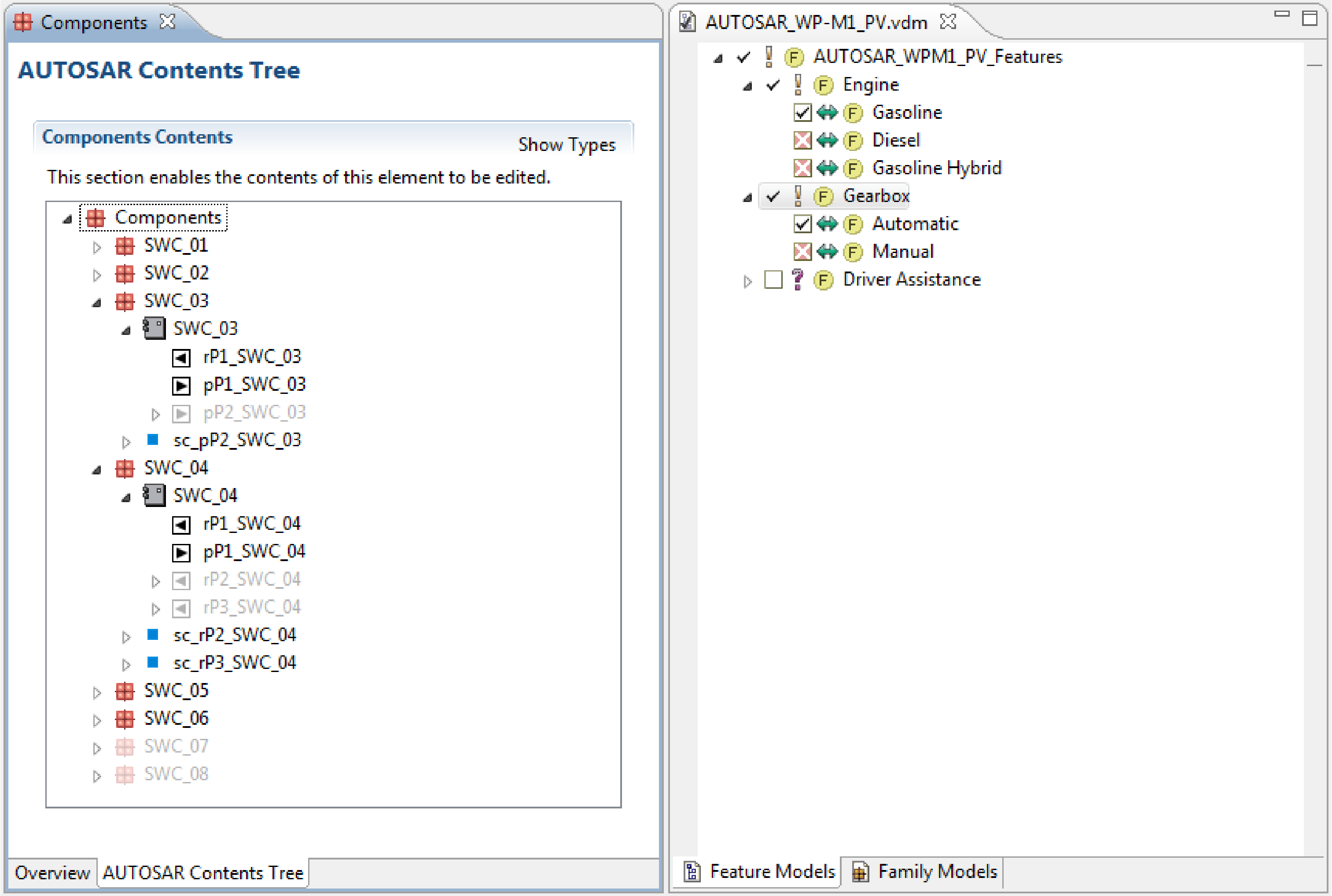


Courtesy AUTOSAR Development Cooperation 2015
AUTOSAR R 4.2.1

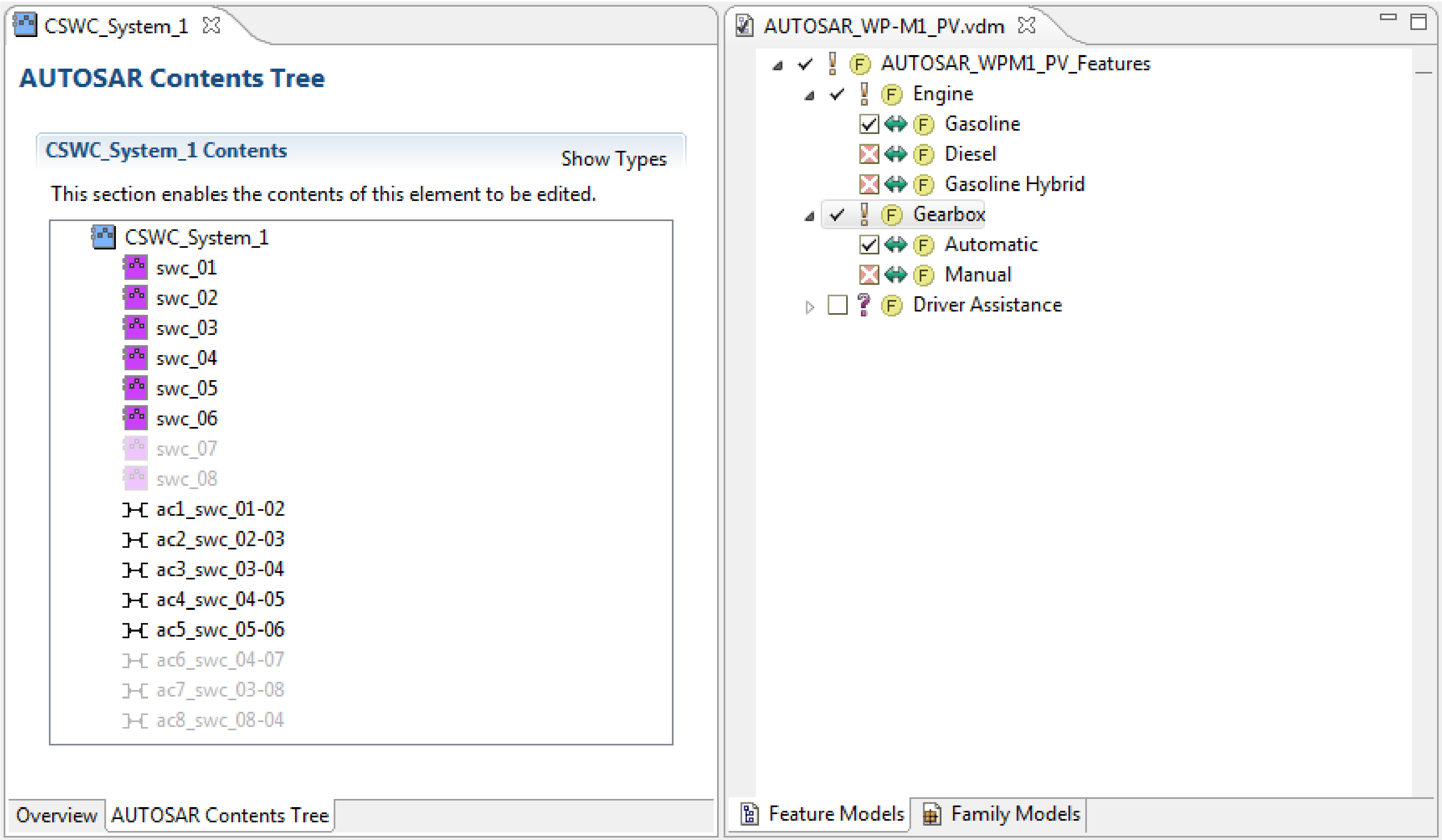
Present
AUTOSAR Example



Present
AUTOSAR Example … *continued*



Present
AUTOSAR Example … *continued*



Present
Conclusion

The automotive industry is making good progress and works towards a standard
!

http://www.autosar.org/

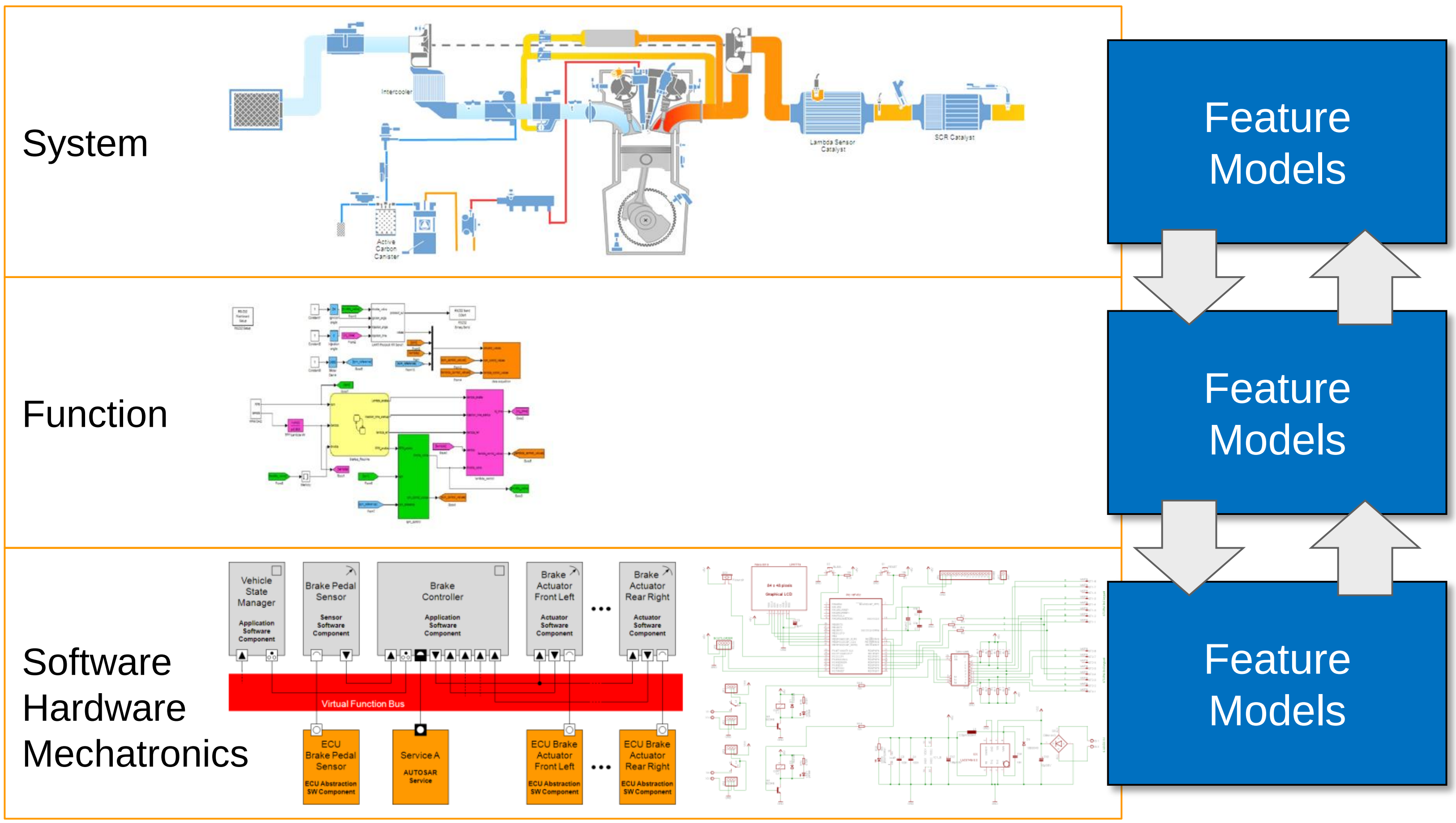
http://www.east-adl.info/

Agenda

Future
Challenges

- Heterogeneous solutions for methods and tools
- There is still no seamless integrated tool chain supporting variations across all levels of abstraction
- Variability is still not considered as integral part of architecture: Requirements - Structure - Variability
- Lack of tools to migrate existing assets: “Harvesting Variability”
- Lack of tools for analysis of variations and variability (visualization)
- Determination of reusable packages

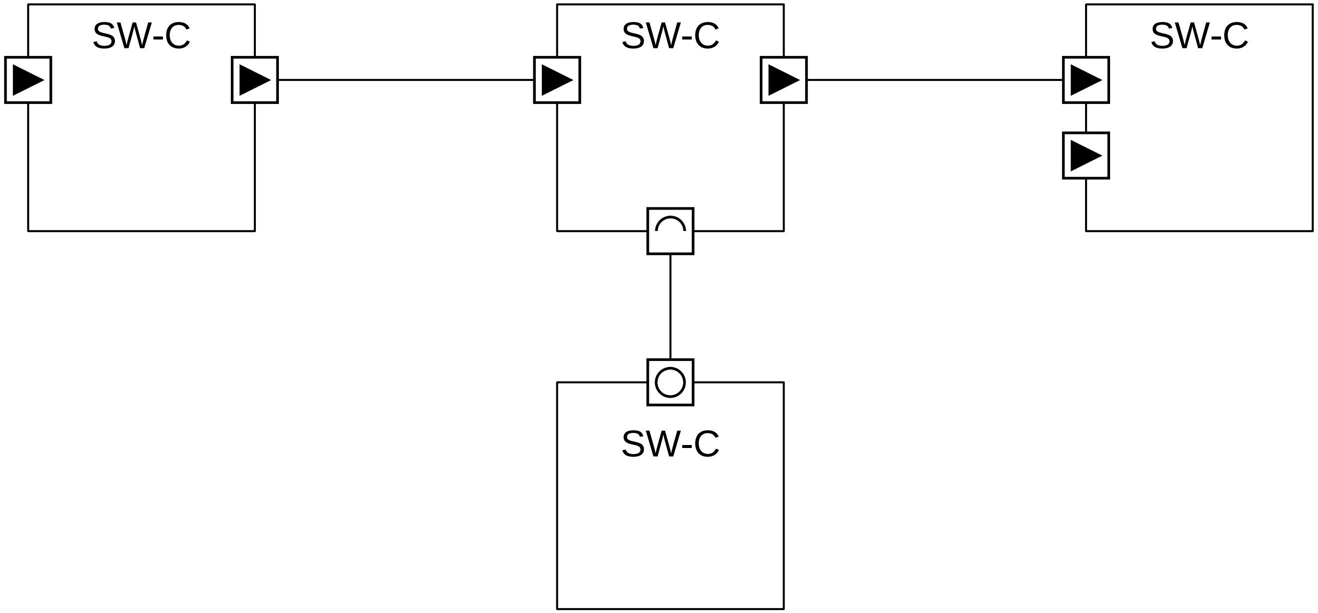
Future
Seamless Integrated Tool Chain



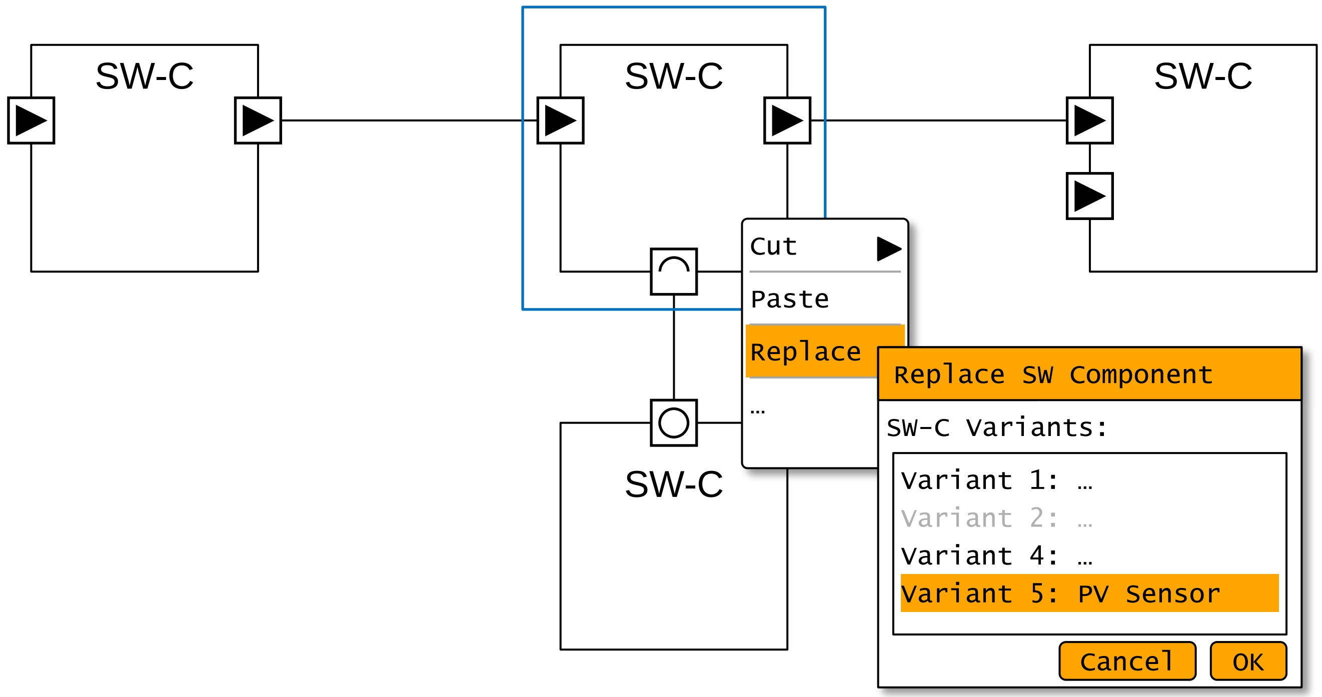
Future
Idea: Configuration Management System and Variability

- Checking out version and variant and vice versa

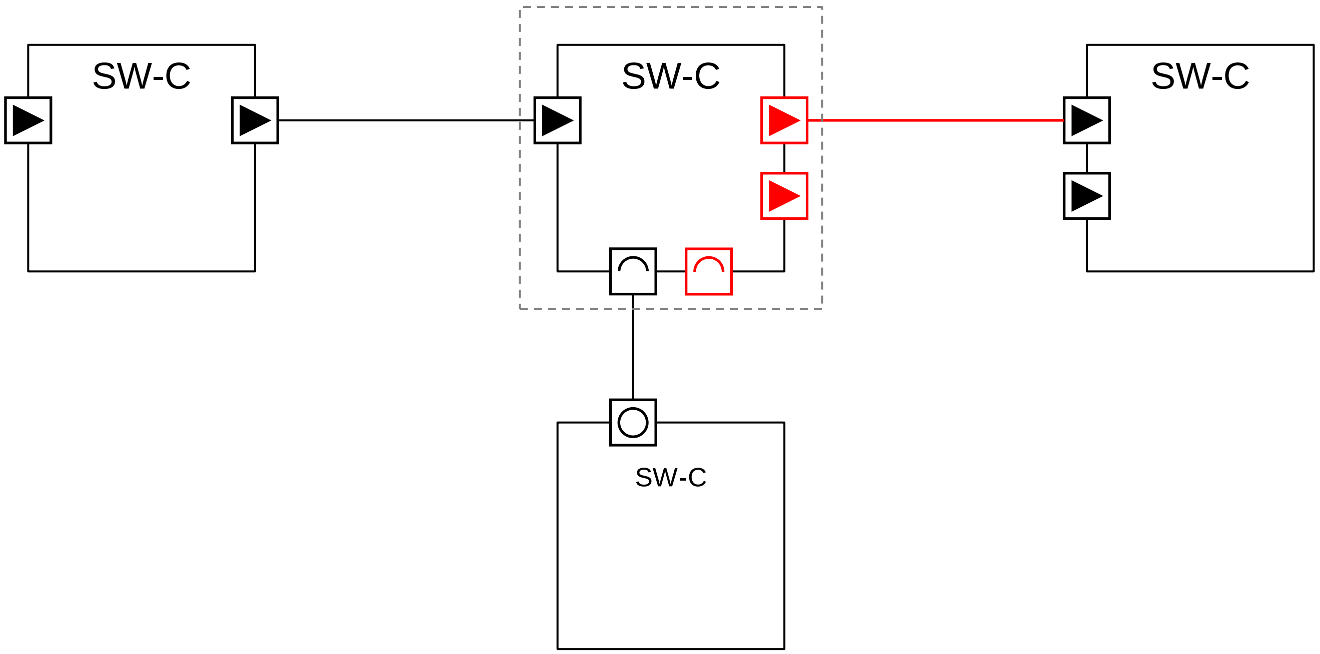
Future
Idea: Integration Tool



Future
Idea: Integration Tool … *continued*



Future
Idea: Integration Tool … *continued*



Agenda

Conclusion
Conclusion

- Theory of variations respectively variability is already understood in the industry
- Lack of integrated and seamless tool chain, for
  - Defining and specifying variations
  - Analyze variations
  - Process variations
  - Visualize variations depending on context

Discussion
Questions and Answers

Thank you very much for your attention!