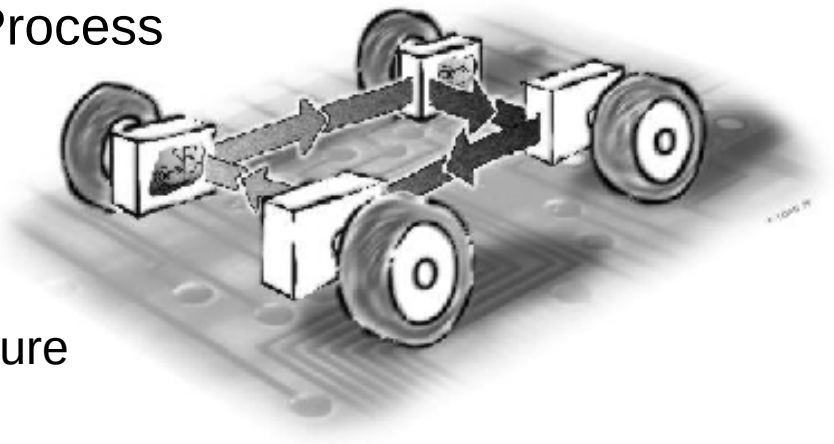
# The Challenge

Product Related Challenges

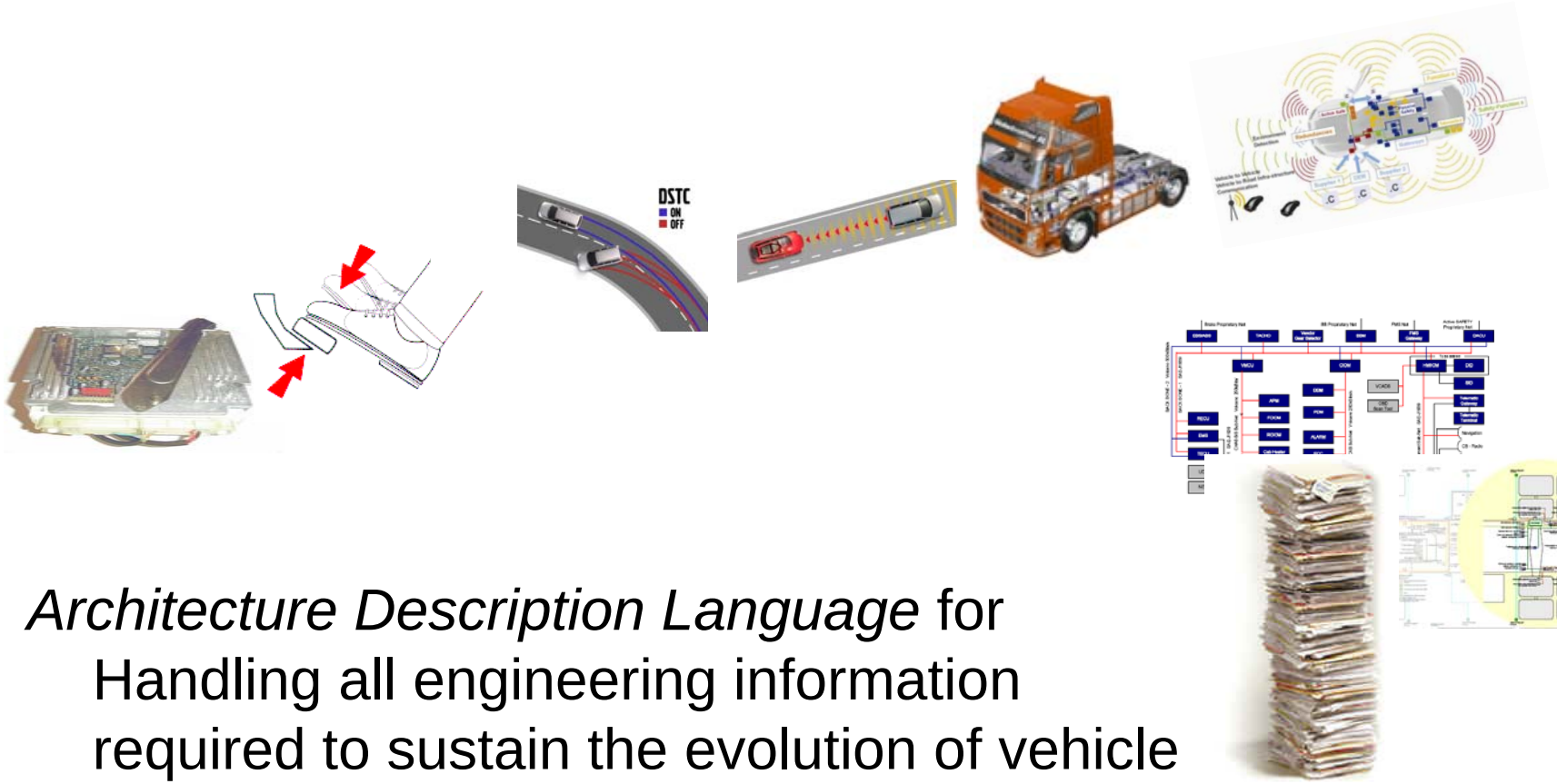
- Functionality increase
- Complexity increase
- Increased Safety-criticality
- Quality concerns

Challenges Related to Development Process

- Supplier-OEM relationship
- Multiple sites & departments
- Product families
- Componentization
- Separation of application from infrastructure
- Safety Requirements, ISO 26262



# The Response - EAST-ADL2



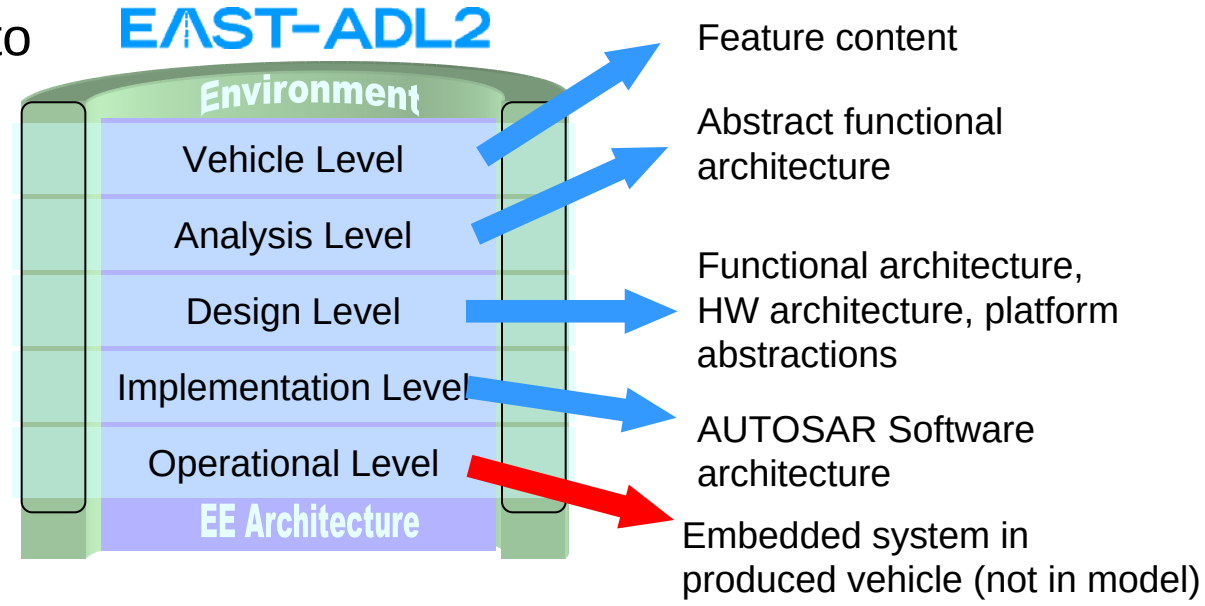

*Architecture Description Language* for

Handling all engineering information required to sustain the evolution of vehicle electronics

# EAST-ADL2

A System Modeling Approach/Architectural Framework that

- Is a template for how engineering information is organized and represented
- Provides separation of concerns
- Embrace the de-facto representation of automotive software – AUTOSAR



# Relation to other modeling languages and approaches?

Why Not UML?

- EAST-ADL2 is domain-specific but its UML2 profile gives access to UML2 tools.

Why not SysML?

- EAST-ADL takes up applicable SysML concepts but provides additional domain-specific support

Why not Autosar?

- EAST-ADL complements Autosar with respect to feature content, functional structure, safety properties, etc.

Why not AADL

- AADL represent the software implementation of a system while EAST-ADL2 starts on a more abstract level.

Why not proprietary tools (Simulink, Statemate, Modelica, ASCET, …)?

- EAST-ADL2 provides an information structure for the engineering data and integrates external tools

# Some Typical Engineering Scenarios

The Vehicle Manufacturer decides what to include in the next product

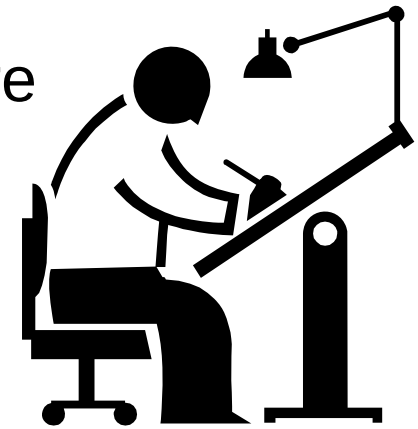
A Chassis engineer analyses a novel control algorithm

Application expert defines detailed design

Software engineer defines software architecture

Packaging and allocation, Integration on ECU

Early phase validation and verification

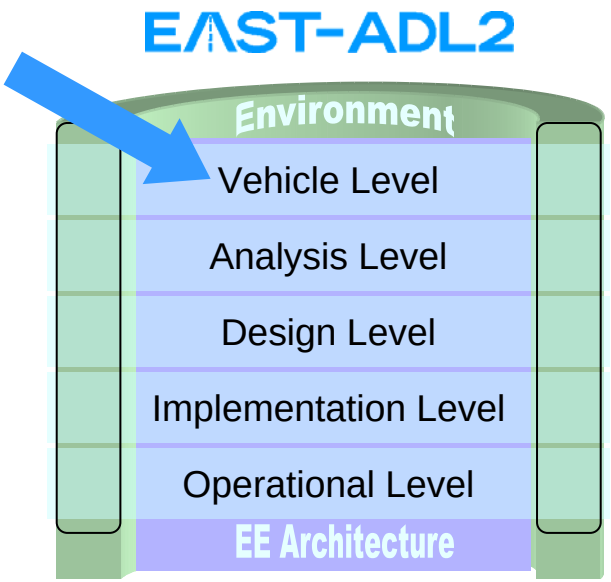


# Product Planners decide what to put in the next product

Features represent the properties/functionality/traits *(Brake, Wiper, CollisionWarning,... )*

Vehicle Feature Model organize Features for the vehicle

Variability mechanism supports the definition of rules for inclusion in different vehicles – Product Line Architecture



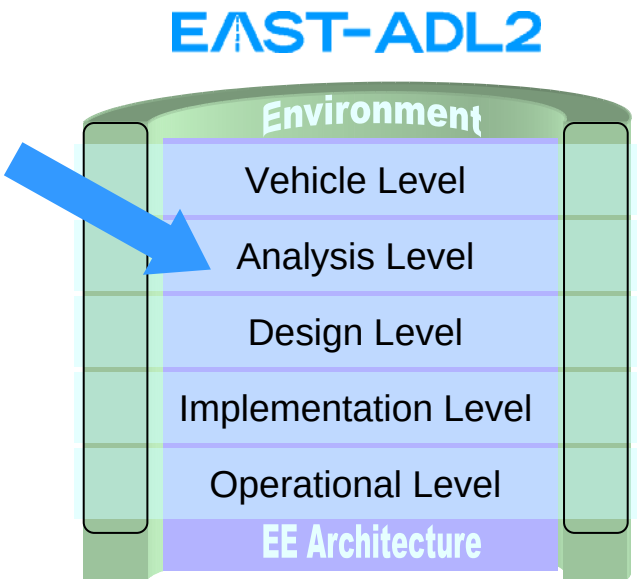
# A Chassis engineer analyses a novel control algorithm

Control algorithm is defined as a
    Function connected to a plant
    Function in the Environment model

EAST-ADL2 defines structure, legacy tools can be used for behavior definition, simulation, etc.

Realization details are omitted:

- Functional validation and verification can be done with respect to key aspects
- Understanding of key aspects is possible



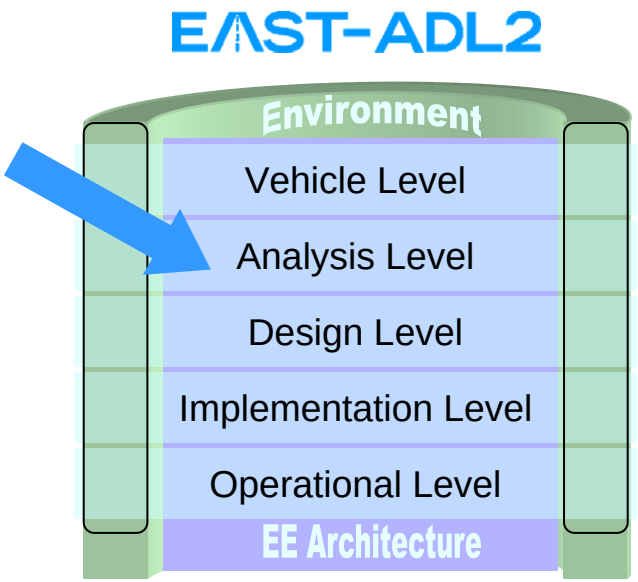
# An OEM and Supplier agree on specification

A model of the supplied system provides a clear and effective information exchange

Functions can be integrated and validated before SW and HW exists

Requirements are explicit and traceable to model elements

Interfaces and interaction clarified, avoiding common specification bugs



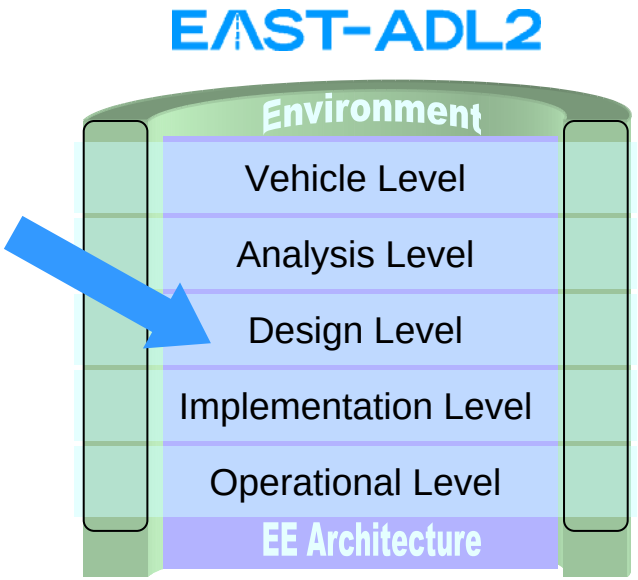
# Application expert defines detailed design

A detailed functional architecture is defined, addressing e.g.

- Hardware architecture
- Allocation
- Fault tolerance
- Implementation concerns
- Sensor, actuator constraints
- Interfaces to middleware

Focus on behavior and interaction of functions

Abstract system architecture is defined and assessed



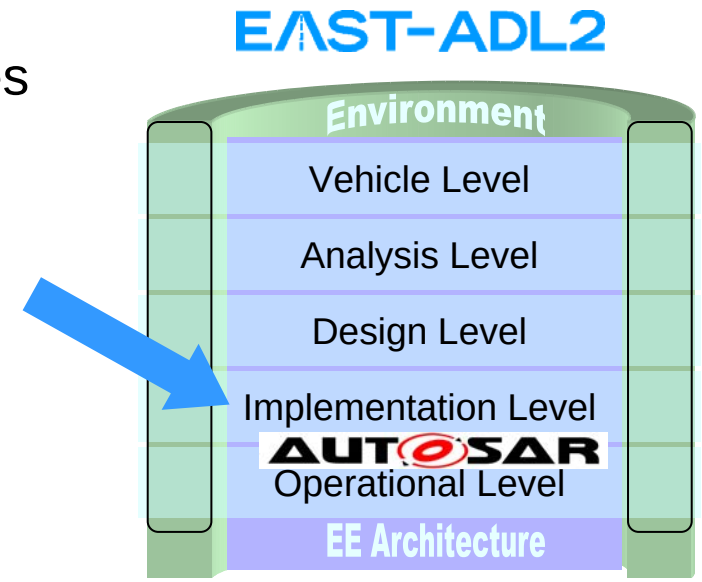
# Software engineer defines SW Architecture

AUTOSAR Application SW Components are defined

The set of SW components together realizes the Functional Architecture

Software organization and functional organization is decoupled and optimization of the SW architecture is possible.

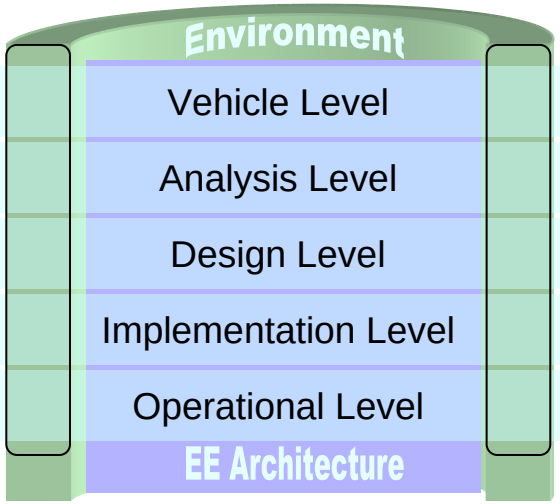
Legacy, sourcing, allocation, performance, verification, responsibility, re-use, etc. influence which functions are realized by each SW component



## Outline

- Example usage of EAST-ADL2
- **Model Structure**
- Example Model
- AUTOSAR Relation
- Areas covered by EAST-ADL2
- Conclusion

# How is an EAST-ADL2 model structured?



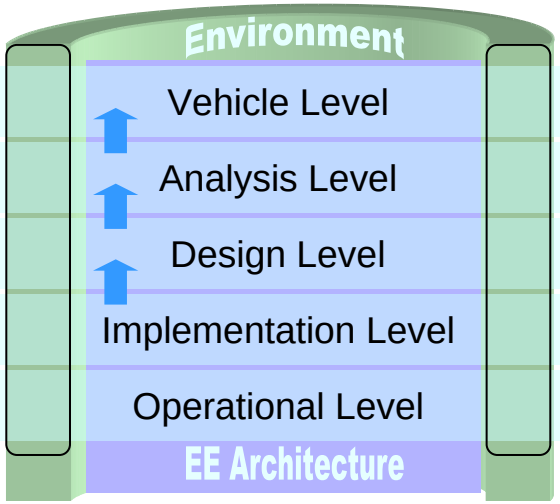

An EAST-ADL2 model is organized in several levels of abstraction, where the software and electronics based artifacts are modeled.

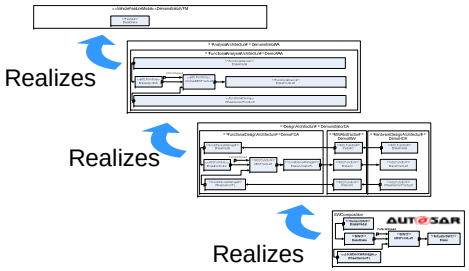
The abstraction levels are “views” on the model and a complete representation of the system.

The contents on an abstraction level forms a complete representation of the vehicle embedded system, with respect to the concerns of that abstraction level

The levels are refined top-down starting at the vehicle level.

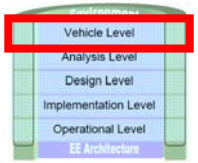
# How is an EAST-ADL2 model structured?



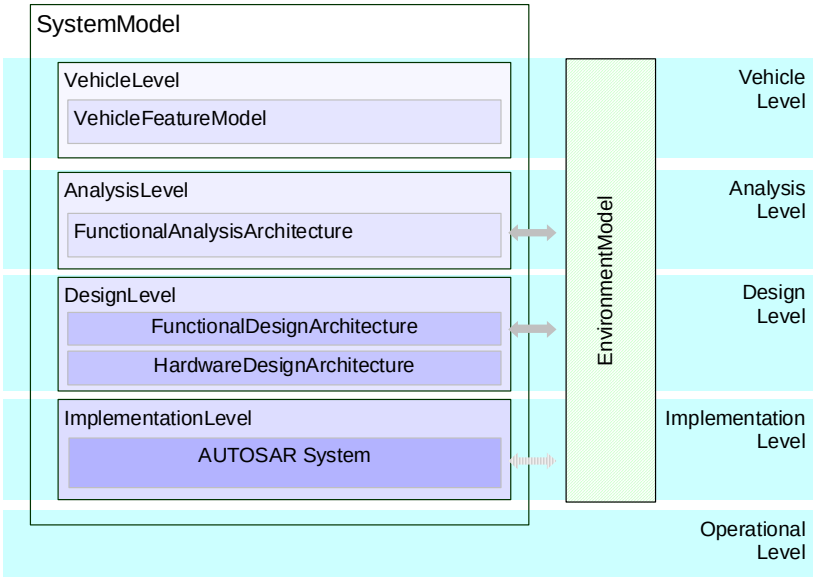




- On vehicle level the **features** of the vehicle
- On analysis level the **abstract functions**
- On design level the **hardware topology**, **concrete functions** and their **allocation** to nodes
- On Implementation level the **software architecture** as represented by AUTOSAR

# Vehicle Level



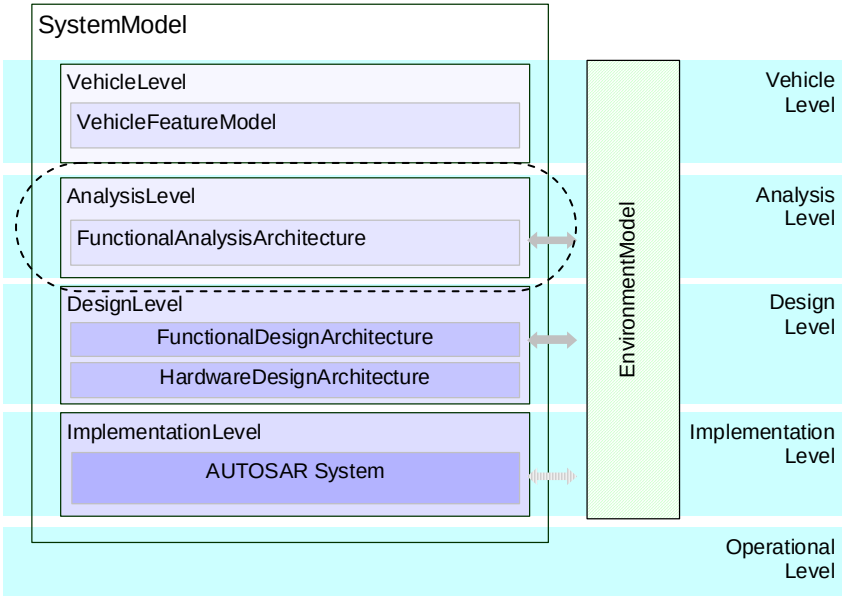

- A Vehicle is characterized by a set of Features
- Features are *stakeholder* requested functional or non-functional characteristics of a vehicle
- A Feature describes that "what", but shall not fix the "how"
- A Feature is specified by requirements and use cases
- From a top-down architecture approach the features are the configuration points to create a vehicle variant



# Analysis Level


Analysis Level is the abstract Functional description of the EE system

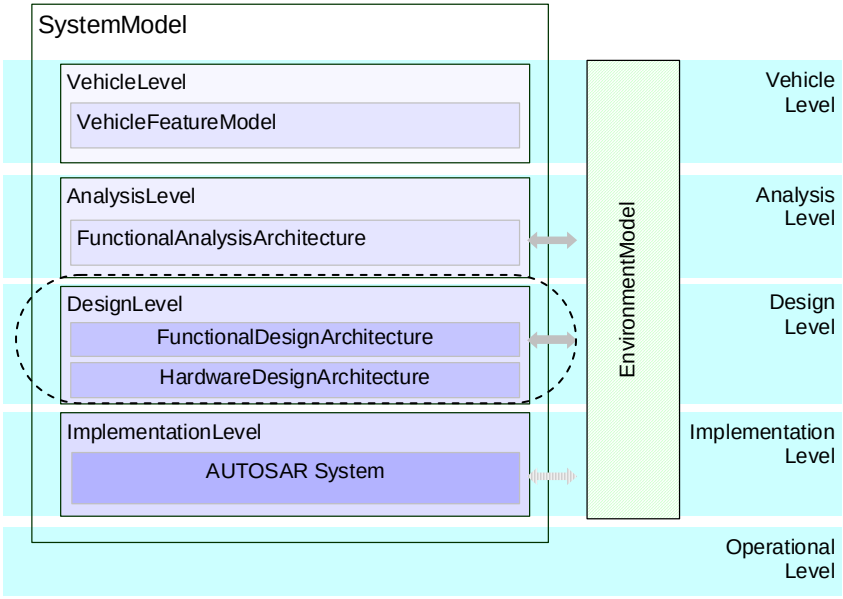
- Realizes functionality based on the features and requirements
- Captures abstract functional definition while avoiding implementation details
- Defines the system boundary
- Environment model and stakeholders define context
- Basis for safety analysis



# Design Level


Design Level captures the concrete functional definition with a close correspondance with the final implementation

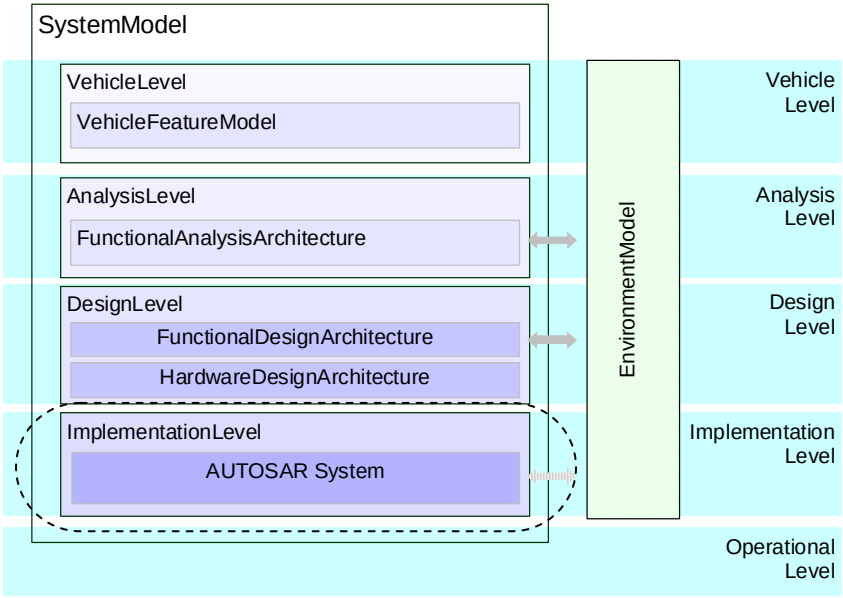
- Captures functional definition of application software
- Captures functional abstraction of hardware and middleware
- Captures abstract hardware architecture
- Defines Function-to-hardware allocation



# Implementation Level


The Implementation Level represents the software-based implementation of the system

- Software components represent application functionality
- AUTOSAR Basic software represents platform
- ECU specifications and topology represent hardware
- Model is captured in AUTOSAR
  - Software component template
  - ECU resource template
  - System Template



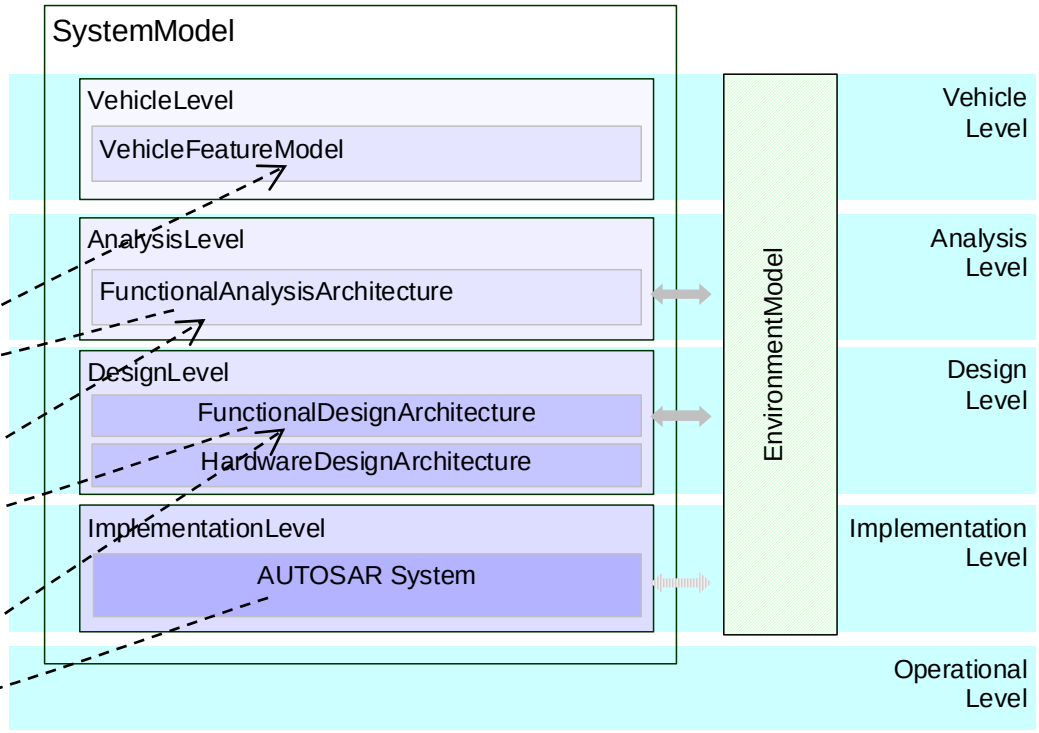
# Traceability between abstraction levels

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

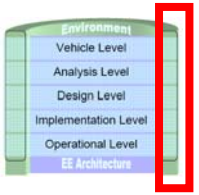
*Functions on analysis level realizes features on vehicle level*

*Functions on design level realizes functions on analysis level*

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

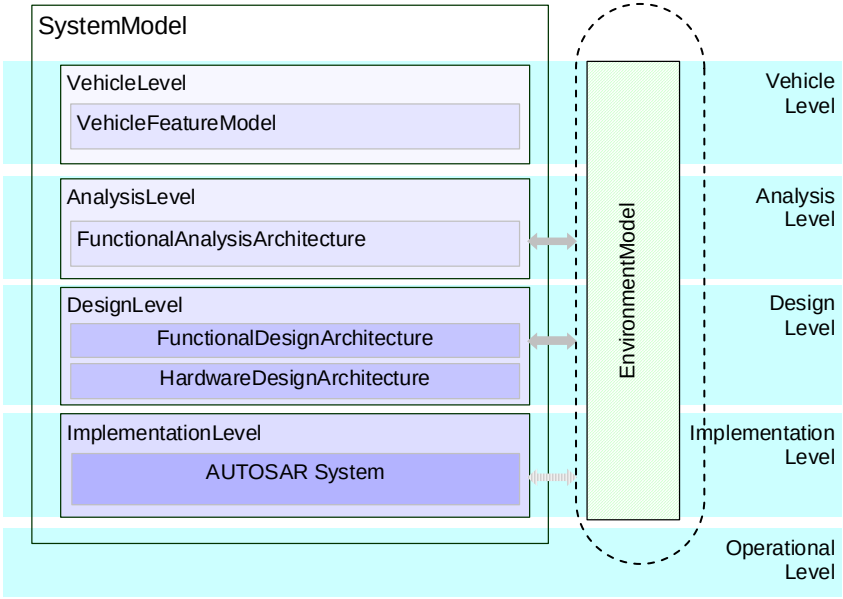


# Environment Model




The Environment model captures the plant that the EE system control and interact with

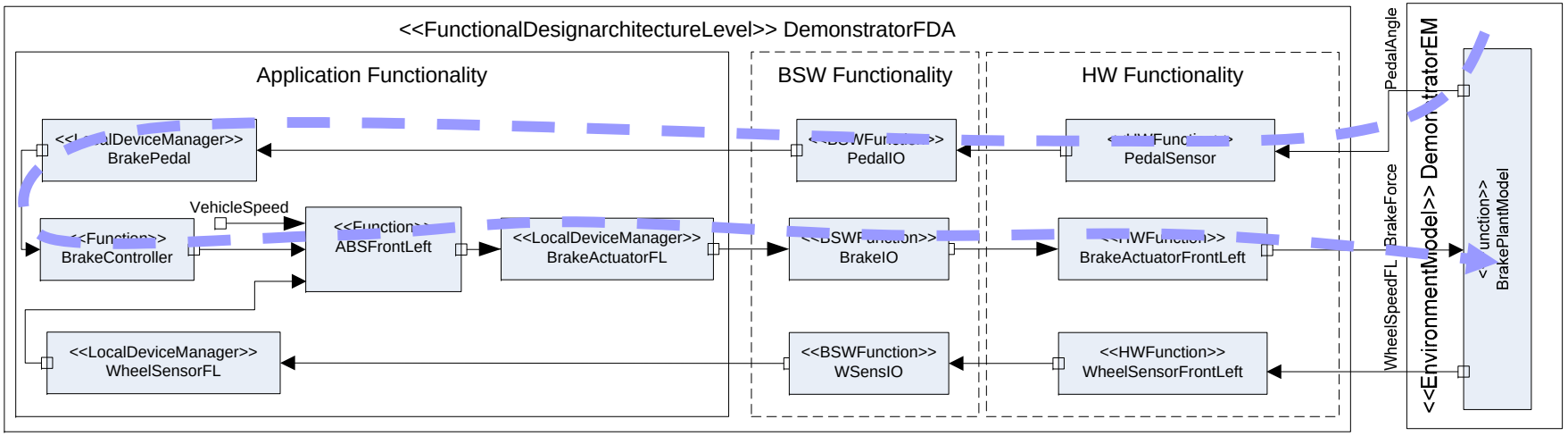
- In-vehicle, near and far environment is covered
- Same Environment Model may be used on all abstraction levels
- Different Environment models may be used depending on validation scenario



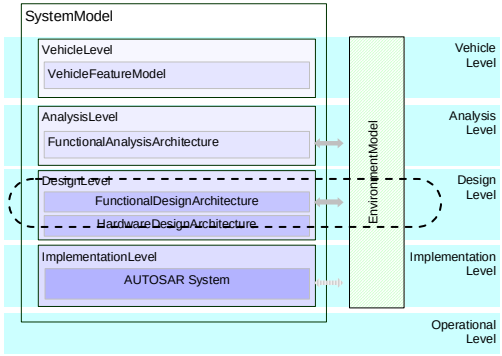
## Outline

- Example usage of EAST-ADL2
- Model Structure
- **Example Model**
- AUTOSAR Relation
- Areas covered by EAST-ADL2
- Conclusion

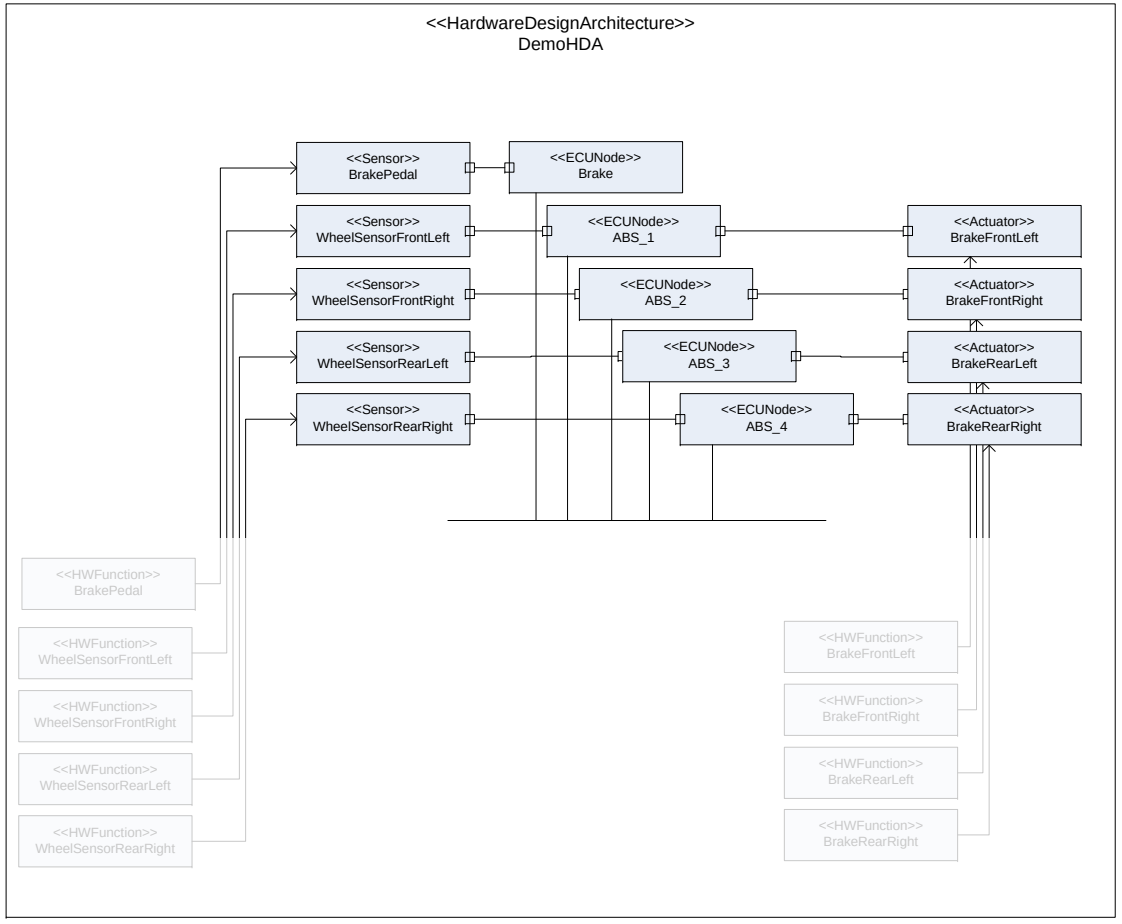
# Function interaction – end-to-end




Model structure supports interaction with the environment and end-to-end functional definitions

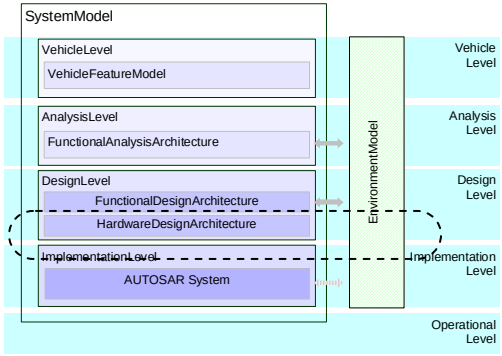


# Hardware Design Architecture




Hardware architecture support hardware design and functional allocation

Behavior of HW entites is defined for use in Functional DesignArchitecture



## Outline

- Example usage of EAST-ADL2
- Model Structure
- Example Model
- **AUTOSAR Relation**
- Areas covered by EAST-ADL2
- Conclusion

# EAST-ADL2 Complements AUTOSAR

EAST-ADL2 is an information structure including aspects beyond the Software Architecture

- Requirements, traceability, feature content, variability, safety, etc.

Provides means to define what the software does

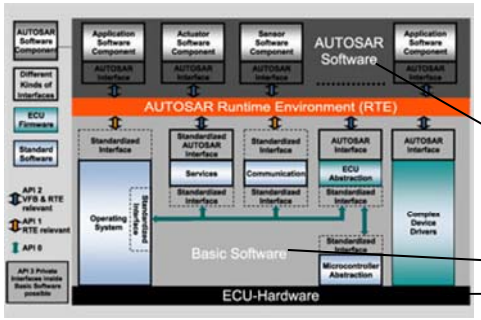
- An AUTOSAR specification defines the software architecture and information required for SW integration - but is neutral to its functionality

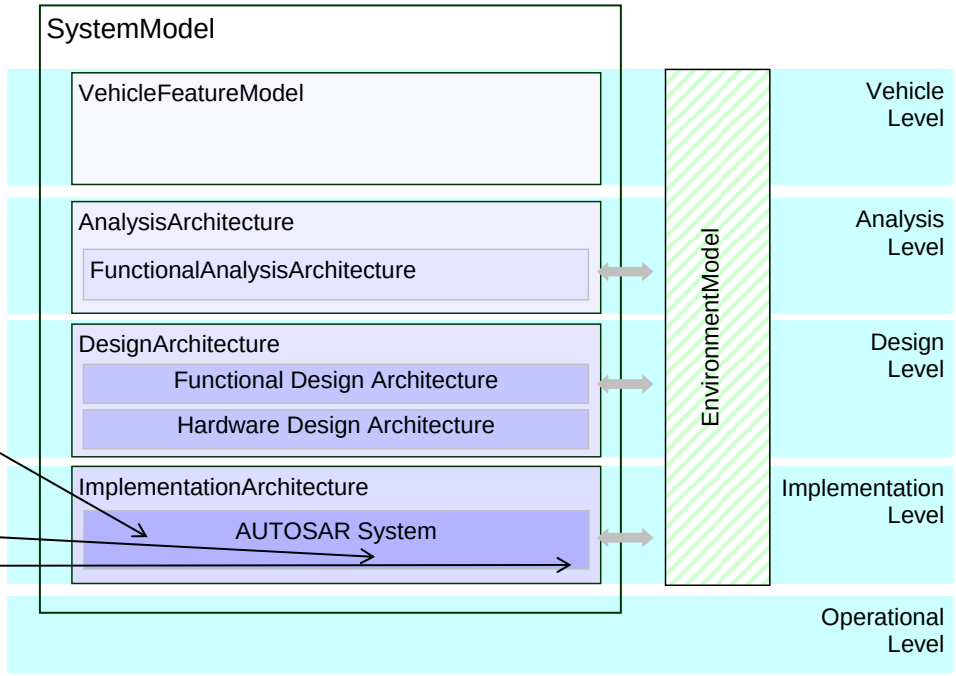
Provides means to model strategic properties

- Key vehicle aspects is captured independently of the software architecture

Supports modelling of error behavior and the representation of safety-related information and requirements

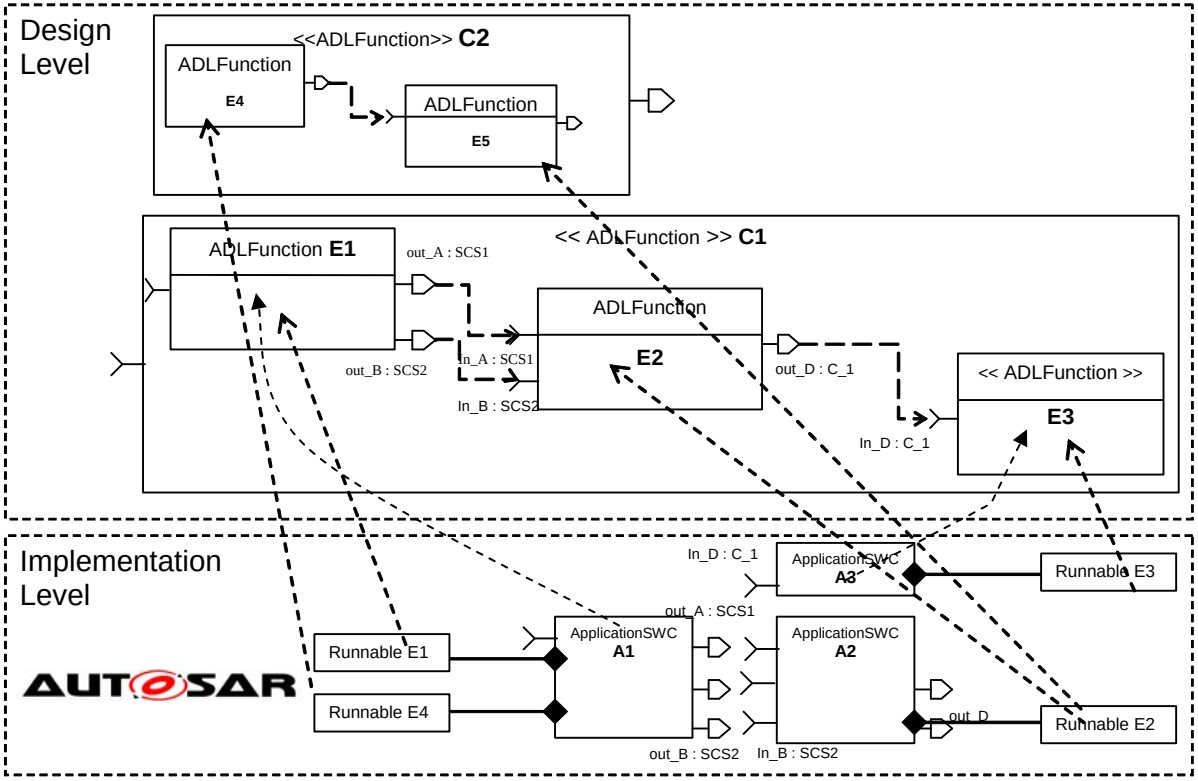
# AUTOSAR vs. EAST-ADL2 System Model

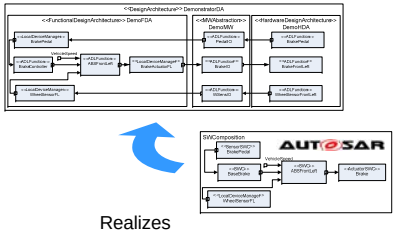


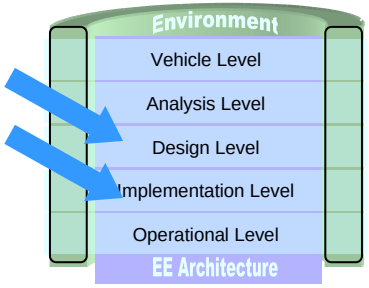



# Relation between Model entities

## Example of Mapping



## Outline

- Example usage of EAST-ADL2
- Model Structure
- Example Model
- AUTOSAR Relation
- **Areas covered by EAST-ADL2**
- Conclusion

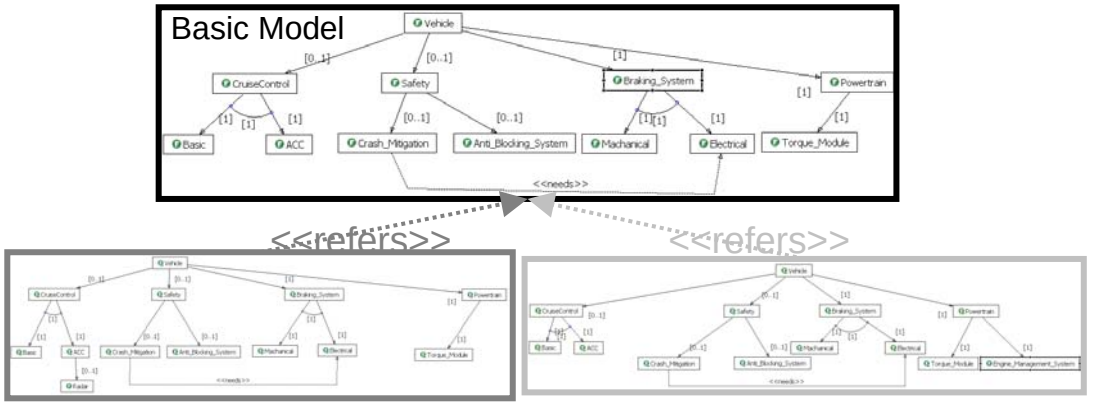
# Variability

## Definition of Feature Content of Vehicle using Feature Trees

- Definition of Product Line in terms of mandatory and optional features for each vehicle category

## Definition of Variability rules for realization

- Optional/mandatory functions and components
- Definition on how to resolve variability based on feature content



# Requirements and V&V

Definition of Requirement modelling framework based on SysML

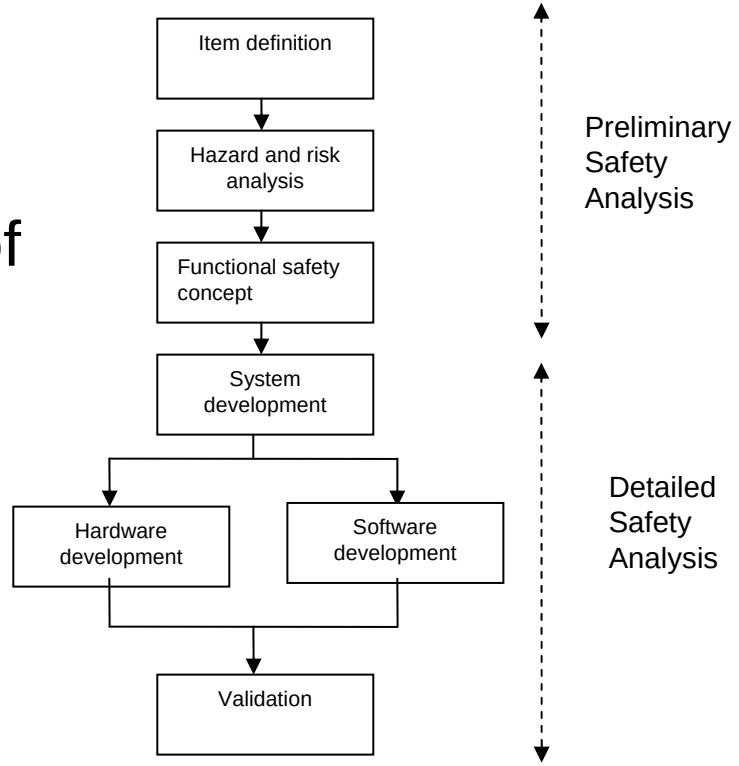
- Concepts for capturing  requirements and components in same model
- Traceability between requirements, components and V&V

V&V constructs to capture test case, test outcome, etc.

Integration of RIF concepts (Requirement Interchange Format)

# Safety Aspects & ISO 26262

- ASIL Categorization through requirements
- Support for Safety Case – Use of model entities to argue safety
- Organization of information in line with ISO 26262
- Support for methods required by ISO26262

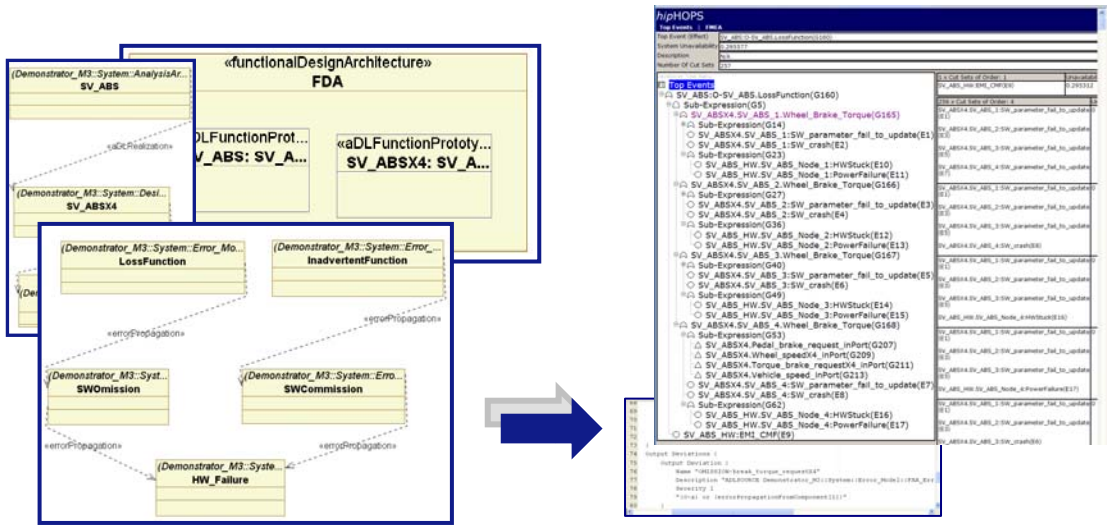


# Error modelling & failure analysis

Modelling Concepts for Hazards and Error Propagation

Basis for Hazard Analysis and Fault Tree and Failure Modes and Effects Analysis

Tool Interface for Automatic FTA/FMEA



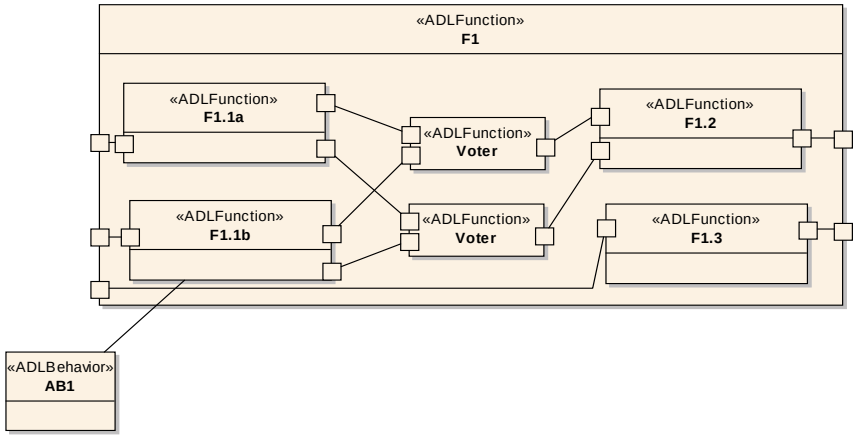
# Behavior

Definition of Behavioral semantics to allow legacy tool integration

- Ascet, Simulink, legacy code, etc.

Definition of relation to AUTOSAR behavior

Behavioral Semantics for Environment model (Plant)

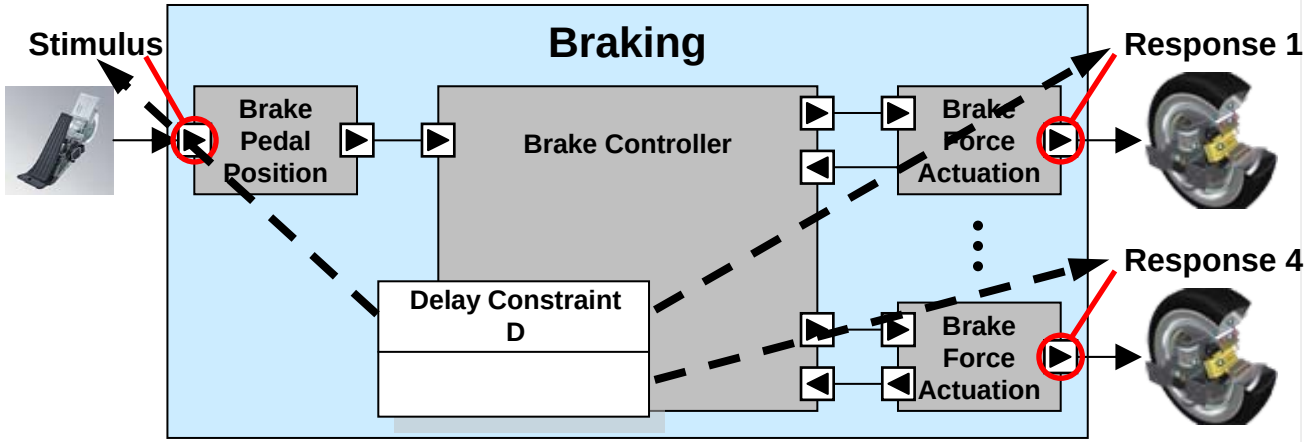


# Timing

Formalization of timing requirements and properties in relation to structural model elements

Approach is symmetric w.r.t AUTOSAR V4 timing constructs

Reaction, age, synchronization and repetitions can be defined



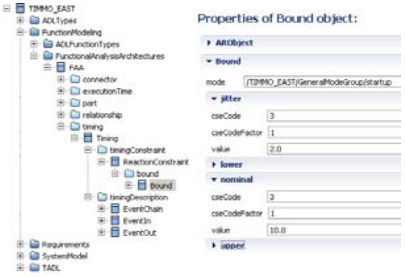
# EAST-ADL2 Tooling

## UML-based Tooling

- Based on CEA Papyrus
- Integrated Eclipse application with 5 ATESST plugins

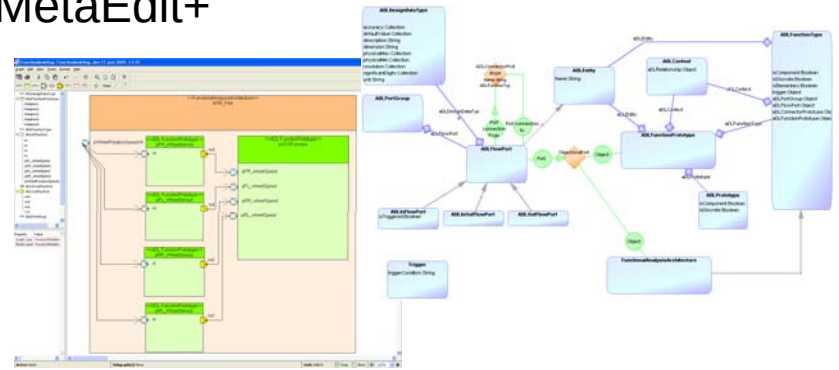

## AUTOSAR-based Tooling

- MentorGraphics VSA




## DSL Tooling

- MetaEdit+



## Outline

- Example usage of EAST-ADL2
- Model Structure
- Example Model
- AUTOSAR Relation
- Areas covered by EAST-ADL2
- **Conclusion**

# Conclusion

EAST-ADL2


EAST-ADL2 provides an information structure for design of automotive embedded systems

- Architecture Description Language and Architectural Framework

Use of abstraction levels is a fundamental concept

- entities on lower levels *realize* entities on higher levels

EAST-ADL2 is a fully aligned complement to AUTOSAR

- AUTOSAR is the SW architecture definition enabling SW component integration on ECU
- EAST-ADL2 supports the successful integration of AUTOSAR components
- EAST-ADL2 Supports additional engineering steps including

***feature definition, requirements engineering, V&V , safety analysis, functional modeling/integration, product line engineering***